SSME*
Service Systems, Modeling, Execution, Education, Evaluation

Manuscript

Zdenko Staníček

2009

Contents

Foreword

PLEASE EXCUSE ME FOR THE ENGLISH! IT IS NOT MY MOTHER LANGUAGE. ALL THIS WILL BE CORRECTED. NOW THE MANUSCRIPT CONTAINS RAW IDEAS EXPRESSED IN A POOR STYLE. FOR ALL THAT I HOPE IT WILL BE USEFUL FOR MY STUDENTS.

Zdenko Stanicek (August 2009)

Chapter 1

Introduction

⋇ [STA]$_1$:PRELIMINARY VERSION. WILL BE REVIEWED AS THE LAST OF ALL CHAPTERS.

The resolution to write this work emerged when the following co-incidence of events in running processes came up: (1) the research on FI MUNI in the field of knowledge and information robots came to new very exciting results (2) specification of fundamental building blocks for knowledge and information robots technology was finished in its pre-alpha version in my research team, (3) SW engineering proof of concept for this technology was successfully done in Mycroft Mind company, (4) course 'IT services management' containing the first insight into SSME was successfully finished on FI MUNI, (5) the process of preparation of a SSME curriculum at FI MUNI, which I was responsible for, brought first clear visualizations and a preliminary curriculum.

Thus the work is a combination of academy research, business practice and academy education for this research and practice. It seems to be an advantage of such a work. But I feel it as a disadvantage according to my experience. There is too much academy theory for practitioners, too much business aspects and issues for academics, and a non balanced portion of educational aspects for both.

But this is exactly this what I wanted to say to this subject in the time I have been written this work.

Why this Theme

The last 2000 years we have lived in goods economy. The focus was and still is on goods exchange, in a product trade. This had brought prioritizing of owner-owned relationship - one of the special connections (cf. below).

It also had created an environment supporting ROS (Red Ocean Strategy - cf. ⋇ [STA]$_2$:citace citace W. Chan Kim and R. Mauborgne: Blue Ocean Strategy. How to Create Uncontested Market Space and Make the Competition Irrelevant, Harvard Business School Press, Boston, Massachusetts, 2005).

And last but not least this focus and this prioritizing of the owner-owned relationship leads to a zero-sum game in the so called GTM (Give-Take Matrix) (see picture 1.1). This is my opinion based on longlasting experience.

Special connections which help us to be oriented in our world are:

- Whole-Part
- Predecessor-Successor
- Container-Item

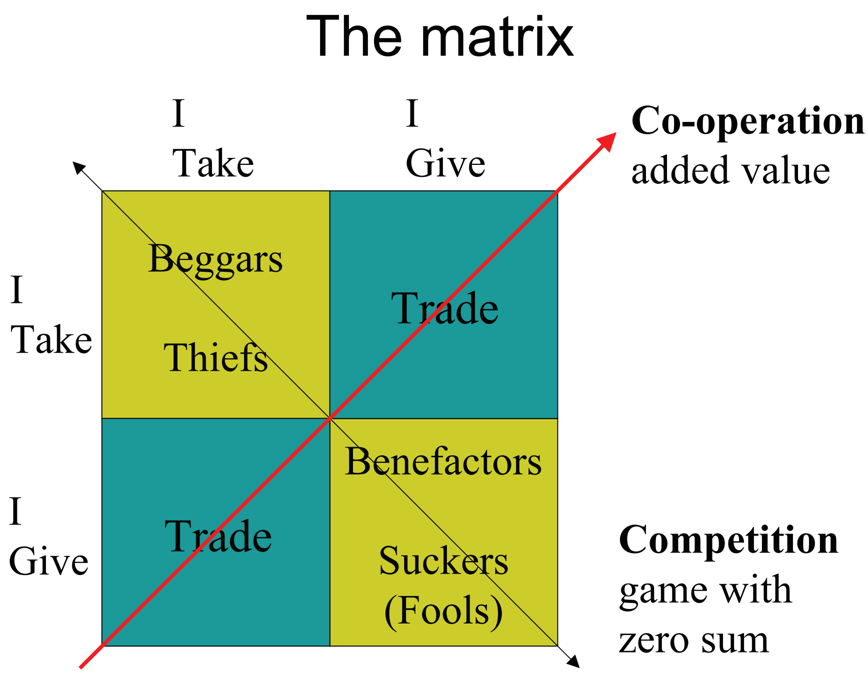

Figure 1.1: GTM

- Pattern-Imprint
- Abstraction-Specialization
- Owner-Owned
- Superior-Subordinate
- Provider-Client
- Serviceability
- Faith
- Fight
- Identity

This is a pragmatical selection. A question wether it is complete or wether it is a minimal set of special connection is out of context. Usually models of the world we have in our brains often count on those connections.

However, the relationship Provider-Client, together with relationships Serviceability, and Faith bring us to the service-focused world. Serviceability connects this what is or can be useful to a player for whom it is or can be useful. Faith connects believing player to the subject of his/her/its belief. Both these relationships are unavoidable in Provider-Client connection establishing.

Let's mention now, as preliminary, that from the above mentioned connections we can construct more complex relationships. Co-operation, e.g., can be viewed as a sequence of Provider-Client connections in which the roles of a provider and of a client are mutually exchanged.

More details to special connections can be found in [PRO2006 –⋇ [STA]$_3$:TODO]. Here, in next chapters, we will use them to explain service system nature, service system structure, its behavior, and its development and evaluation.

Let us note, that especially the identity connection, which is something strange but the most important one, is worth our attention. Who am I? Are those two focused service systems really two or is it only one and the same? What is this what differs between individuals and what is this what gives us a sense of identity of an individual? Such questions sound like philosophy far from business. But to formulate what is a benefit for particular agent and to associate real benefits to their winners it is necessary to answer those questions. We start with this in chapter 3 especially in its subchapter Boundaries, Contexts, Co-operation and we will continue in chapter 6 on service systems modeling.

The Red Ocean Strategy is a never ending battle resembling pack of dogs brawling for a bone. Isn't it better to try to find another bone? Intuitively we know: yes it is. And this is the Blue Ocean Strategy. Practically we don't do it very often. We forget our intuition and we throw into the battle. It means we stay in the Red Ocean Strategy. Chapters **??** and 9 on an usability framework of service system and its evaluation deal with this topic in more detail.

The Give-Take Matrix (GTM) is my old picture by which I provoke people when they forgot that „to englut is allowed but to smack is forbidden" ⁕ [STA]$_4$:–TODO preklad: „žrát se může, ale nesmí se mlaskat". The fundamental message of this GTM is a preference for the principle „cooperation and creating value" to the principle „competition and zero-sum games." The principle „cooperation and creating value" means that when we first GIVE (investing - energy, time, money, passion, influence, expertise) we then expect that we will TAKE (rewards of adequate capability). And conversely: when first TAKING (receiving money, case, energy, fruits of labor, or expertise from someone in our proximity), we know that we will then GIVE (rewards for prior gains by giving investment of our energy, time, money, passions, influence, expertise). And we will see that this GTM principle is a base for service system functioning so as for appropriate value proposition formulation.

Now the complexity of business and technology relationships overflows the goods exchange paradigm. Co-operating relationship is in focus rather than ownership. The success of service business goes along the red arrow in GTM only. This is my honest opinion.

Moreover, I am sure the future of the world economy will be the service-centered economy. We are going back to ancient times where everything was about services. The goods economy period was and is a useful bridge from the very past to the near future, only, which made and make a significant step in wealth of nations and which helps us to understand that the life is about THE USE not about THE MENTION. Goods is useful when serves us, only. Mentioning it without obtaining a service from it the goods is only for swagger.

But the right use of the above mentioned special connections to the benefit of service systems providing and utilization is not an easy task. We are not well trained for it from our history. Thus I would like to make one of the small first steps towards this challenge. This means:

- To describe one possible view of service systems
- To make a comprehensive and consistent explanation of their nature, of their functioning, of their organization, of their behavior
- To bring ideas of their applications and dissemination
- To start a discussion on the education within the service system framework
- To show possibilities of service system effectiveness and efficiency evaluation
- To sketch possibilities of their evolution

This is the answer on the title question–why this theme.

Inspirations

My twenty years practice in information strategy and global strategy preparing with my clients and in leading people to strategic thinking was for me the main inspirative issue. Based on classical joint development approach ⋇ [STA]$_5$:najit od HAJ citaci knihy, kde se o tom psalo and on practice of workshops I developed my own method of leading a group of people to an agreement. This method–called a method of forming seminars–helped me and my consultant teams to create and handover tenth of information or global strategies in a range of Czech companies and institutions.[1]

The most important issue from this experience is: without a clear and common shared strategy any action in business so as in life tends to be similar to the well known Brown movement. Thus strategy is the first citizen of any action preparation. Strategy tells us the direction. Strategy help us in ongoing decisions. Strategy is a criterion of what could be meant by success. We will speak on it more in chapter 5 while dealing with agents team organization and its predictive behavior. A utilization of the issues of strategy will be done in service system modeling process.

[1] Information Strategy of Expandia Group holding which had led to the first Czech internet bank–eBank (1996), Global Strategy of Brno Heating Plant (1997), etc.

Endeavor of Business Rules Community was the second inspirative issue. When we started to think on universal information robots[2] in 2002 year with my student and later co-worker Filip Prochazka[3] we had found the Manifesto of Business Rules Community ⋇ [STA]$_6$:citace BR Manifesto. The main idea of this Manifesto is separation of business rules from program code of information system or other SW tool supporting the business. We can have a business rules engine on one hand and a set of actual business rules on the other hand. The engine works as an interpreter of those rules. When anything change in business we rewrite the appropriate rule, teach it the engine and the business rule system starts to work according to the new rule immediately–without the classical long way of (analysis of the change)–(designing)–(implementation)–(testing)–(running) process. According to my point of view it was a service approach: the business rule engine was for me something like a service agent or an assistant. When using services from an assistant first we have to teach the assistant how to react on chosen events, how to perform desired processes, how to recognize which way is better, etc. And such a way trained assistant can provide us with relevant services. This is the main point of our past idea of universal information assistant or robot. ⋇ [STA]$_7$:citace clanku o UIR The way how the robot performs services we ask on it/him(?)/her(?)have to be learned by the system so as we humans learn what to do and how to do it (or how to behave) in particular situations. My opinion is that it is the only way how to bring SW systems closer to human like adaptivity to situations and to progress. Business rules community assured my belief that the „assistant providing services“ point of view to SW systems is possible and, moreover, is useful.

Multi-agent systems and their construction was the third inspiration. Such a system is composed of agents. Each agent is autonomous and co-operative. The autonomy means it can perform some actions individually. Cooperativeness means it can communicate with other agents, it can together with other agents create a plan, and they all together can perform actions according this plan. When we started research of multi-agents systems just in first days of January 2007 in my team[4] we found out very soon that better

[2] first they were universal information recorders; we turn to robots at the beginning of 2004 when elements of artificial intelligence occurred in our SW product

[3] I had trained him together with other members of the team and now he is the leader; I am something like a coach.

[4] really it wasn't my team only in this time already; in this time it was our team–excellently trained more than twenty members and two leaders: Filip Prochazka and me

than one universal robot-assistant is a team of specialized robots-assistants, in other words agents, which could be trained to do a work of the universal robot and much more over the former universal robot's capabilities. In such a team of agents (or robots-assistants – choose what wording is better from your point of view) each agent switches between a role of a provider of some service for other agent (or agents) and a role of a client of some service from other agent. Usually, the target of these services is information and/or knowledge.

Our research in Knowledge and Information Robots (KIR) domain in the first half of year 2007 was like an eruption. We drew mind maps, argued on workshops, and wrote research reports and specifications repeating this cycle many times. In a half year of this very intensive work and team cooperation we obtained consistent and enough complete specification of an multi-agent system or we can say of a team of robots-assistants with some special properties: First, our agents were[5] either primitive or composite, i.e. an agent could possibly be a composition of other agents. Second, there was[6] a special agent/component called semantics memory which enables in one universal structure to store any object model of a part of reality so as we, human beings, do it in our brains.[7] Third, the memory component and the agents know to work with uncertain and context dependent information. Fourth, the team of robots-assistants is self-referential. This means: so as a football team, e.g., can work on itself to be better, the agents team can do something similar. These features lead to amazing adaptability of such systems. Services provision and consumption requires a high level of adaptability. This was the fourth inspiration.

At this point I would like to say „Good by" to those readers who cannot continue reading as they are sure all this is a fiction. I am sorry. It looks like, but it is not.

Along with the KIR domain research I started to read papers from the domain of SSME–my fifth inspiration. The most inspirative were papers form Jim Spohrer and other IBM Almaden fellows. Their description of the

[5] and they still are and will be ...

[6] and there still is and will be ...

[7] I do not want to argue that this semantics memory is a copy of our brain internal structure; I only argue that so as we have in our heads only one structure to which we store any kind of knowledge and/or information, the proposed (and now more than a year implemented and working) semantics memory is able to store any kind of knowledge and/or information

state-of-the-art in the services domain shifted me from the world of classic applications and classic IT approach–not very pleasant for me–to the world completely friendly and for me understandable.[8] Something like de-ja-vu ⋇ [STA]8:overit psani for me.

The first thing is Spohrer's definition of service and service system as an composite of technology and people (cf. SPOH ... ⋇ [STA]9:TODO citaci). My point-of-view is that our problems with understanding the world arise from „boundaries". Trying understand to a natural phenomenon we put artificial, i.e. non-natural, boundaries to those natural issues believing the situation becomes more clear, but it deceives us and precludes to understand very often. Every boundary crosses some natural connections which start to be invisible due to this boundary. May be it is too hot coffee for the beginning but try to live with it for some time; in the following chapters I will explain the idea more deeply.

A typical such boundary in IT is: we - creators of IT based solutions - are on one side, and the system - the means of our IT based solution - is on the other side. Thus our rules and our behavior is something completely different from the system rules and the system behavior. More than six year I had felt it as a fallacy. Four years ago I started to speak about information and knowledge robots to express that system (the robot) lives and behaves under the same rules as we do. Spohrer's definition brings me liberation from my lonely opinion. My robots are simply service systems. And we, I and you, are service systems too. Moreover when we together (with some other fellows) create a team and start to produce some solution for a client, we are service system again. When we adopt a software solution to help us with the service provision, we are together with the piece of SW a service system. In any case: a service system is something which can be used and which brings a new value for its customer. It doesn't matter whether it is completely artificial or completely natural, namely human, or a mixture of both.

If it is so, the behavior of such service system have to be the same even if its artificial part is changed, e.g. is extended against its natural (human) part.

It looks like this is all. Five inspirations is enough. But at the point when we finished a pre-alpha version of our technology components and the whole platform for „real life" of the robots-assistants team (at the beginning

[8] I hope my point-of-view is comprehensible taking into account the previous paragraph.

of 2008) I was overwhelmed by a Jeff Hawkins book „On Intelligence" ⋇ [STA]$_{10}$:citace . The quotations (written in Italics) used here till the end of this section comes from this book. ⋇ [STA]$_{11}$:TODO: make a bridge from service science and systems to intelligence and brain functioning.

„Behavior is manifestation of intelligence, but not the central characteristic or primary definition of being intelligent".

Jeff Hawkins had formed three criterions to understanding the brain. The first one is the inclusion of time in brain function. This means: we cannot understand the brain when investigate it statically. The time dimension is unavoidable. Every process in the brain must be studied in its time progress. *„Real brains process rapidly changing streams of information. There is nothing static about the flow of information into and out of the brain."* Something very similar we will see on service systems in chapter 3.

The second Hawkins criterion is the importance of feedback. *„... for every fiber feeding information forward into the neocortex, there are ten fibers feeding information back toward the senses. Feedback dominates most connections throughout the neocortex as well. ... It must be important."* We will emphasize in chapter **??** the role of visualization. We can say the service system usability depends much more on its ability to produce an effective feedback to the user than on its ability to absorb and maintain data from the environment. The feedback is what matters. The effective and intelligent feedback is the most important feature of the system.

The third Hawkins criterion is that *„... any theory or model of the brain should account for the physical architecture of the brain. The neocortex is not a simple structure ... it is organized as a repeating hierarchy."* The physical architecture of the brain tell us some important pieces of knowledge:

- We do not remember whole objects of real world in our neocortex, we remember connections only; the cognition of objects is a result of brain's process–building desired object from appropriate connections.
- We have only one uniform structure to remember the whole variety of things from the real world; that is why we can remember for us something very new without a necessity to attend a hospital where surgeons would operate our brain to restructured memory cells.
- Repeating hierarchy will be a very important and very fruitful pattern.

The Jeff Hawkins hypothesis is: functioning of our brains could be explicated by the so called memory-prediction model. Basics of this model are:

- there is one „container structure“ for all possible real world structures being stored into this container (memory),
- there is only one operation by which the content of memory is used, maintained, actualized, etc.; this is the so called „prediction“
- *„The brain treats abstract and concrete objects in the same way“.*

„The human brain has an incredible capacity to learn and adapt to thousands of environments that didn't exist until very recently. This argues for an extremely flexible system, not one with a thousand solutions for a thousand problems.“

Concerning my inspiration it is enough. We will bring some more details from the Hawkins' book in chapter 4 as a preparation for Diamond path framework understanding.

But it was not only the content of this book what appealed me. I had found the way it was written as a very proper to the situation when author writes on very innovative subject where there is no long common share of ideas, approaches and opinions between the public. Thus I decided to follow this style, and nevertheless there exists a criticism to writing serious work in this „I-form“, I will do it. When something is my point-of-view, my formulation, my opinion, my hypothesis, or my mistake or misunderstanding, I do not want to hide myself behind an evasive „we“.

SSME and SSME*

Service Science, Management and Engineering (SSME) is a new multi-disciplinary research and academic effort that integrates aspects of established fields such as computer science, operations research, engineering, management sciences, business strategy, social and cognitive sciences, and legal sciences. A great movement is around, lot of initiatives and dozens of articles, presentations and conferences is devoted to it. SSME = Service Science, Management and Engineering–this is world wide used meaning of this abbreviation.

But there can be another „working“ meaning: SSME = Service Systems, Modeling, and Execution. And not only Execution, but Education, Evaluation, Evolution, etc., too. So I will use the abbreviation/symbol SSME* for this (little bit extended) meaning. In fact there is no extension in the meaning comparing it with the common understanding of SSME. There is only explicit focusing to a pragmatically (and according to my experience) chosen aspects of the phenomenon of Service.

The following pages up to end are about this meaning, i.e.,about SSME*. And I hope the title of this book starts to be better understandable. A structure of this work is encoded into this abbreviation, i.e. SSME*, too, see below in the Work Structure Explanation paragraph.

Understanding the Services Games

A service is a provider-client interaction that creates and captures value. It accounts for technical, people, social, organizational, business, management, and others aspects of offering a service. Thus human and software assets are the starting point of modeling a service. Simultaneity of production and consumption is a key issue of this game.

Frequent lack of a central artifact in executing of a service raises an important and interesting corollary: Each party in the exchange needs the other's party knowledge in negotiating the exchange.

Service is an application of competences for the benefit of another; this means: service is a kind of action, performance, or promise that is exchanged for value between provider and client.

„A computing-driven revolution is under way in global economy guided by the principle that every business must become a service business in order to survive." [CACM07 ⋇ [STA]$_{12}$:TODO citace]

Multidisciplinary nature of services theory and practice is crucial. „It is not only the insights from more disciplines, but more importantly it is a collaboration of experts from several disciplines examining the same challenge." [CACM07 ⋇ [STA]$_{13}$:TODO citace] The reason for this is not only experience-based. It has a deep theoretical background: Let us collect all information that could be obtained from a set of propositions. Let's do it for two or more sets of propositions. Making union of those information isles gives us less information than if we first make an union of all primary sets of propositions and collect all information that could be obtained from this big set of propositions. This property of information contained in propositions is called that a „Reverse Triangle Inequality" holds for information capability. Precise understanding to this topic will be offered in chapter 4.

However, what is this what differs services innovation from goods innovation?

First, it is that services are increasingly knowledge-based, not in the service development phase, only, but in the service execution phase, too; this means the knowledge mentioned here is a knowledge shared between a client of the service and a provider of the service. When I buy a bread I needn't to share a knowledge how to use it with my bread provider. But when a bank top management buys a service consisting in a help in the information strategy preparing process, then provider's consultants have to share banking knowledge with the client so as the bank top management has to share knowledge of strategic formulation, strategic choice, strategic implementation, and strategic changes and strategic continuum management with the provider's consultants.

Second, it is that services are increasingly drivers of growth, which is not true for the goods, nowadays. You cannot move from Brno to Prague using three cars simultaneously. Using of goods has physical limits. But you can outsource practically anything what has a nature of service utilization. A peep-show, for example, is outsourcing of sexual requirements of the viewer. Looking on civilization achievements we can see thousands of sometimes interesting and sometimes stupid services which are payed for. Services utilization has no physical limits at all.

Third, services innovation comes less from classical research and development and it comes more from acquiring knowledge from outside sources, i.e. from collaboration, most often between client and provider of the service. The driver of innovative movement is combining of at least two different approaches and two different concerns: the first approach and concern belongs to a provider, the second approach and concern belongs to a client. It is the same principle as that to the birth of a child there have to be a heterosexual couple.

Fourth, services innovation depends on highly skilled and educated employees and entrepreneurship is a key driver of services innovation. Services provision is a very dynamic activity. It is very individual to the situation, very individual to the players. Simply said, this is a project encompassed by all its properties, specialties, and issues.

Summarizing what was said, the service oriented approach needs tools facilitating enterprise transformation from goods economy to services economy. The economy paradigms, rules of law, and the whole society is not prepared enough for this transition. To make appropriate abstractions and to formulate usable models that will help us to understand a new arising complexity of the service world needs some new approaches. Agent based modeling could be a useful support as it captures complex adaptive systems where organizational behavior is emergent. It can represent and facilitate work process changes. Interactions among autonomous agents (human and/or non-human) generate patterns that cannot be reduced to the proprieties of the agents themselves.

Cognition, Knowledge, Semantics, Behavior

The importance of cognition, knowledge, semantics, behavior, and of motivation for improving of our ability to cope with any of those topics descends from the tops of philosophy and theoretical science into a deep and a really life practice.

No provider can provide a service without cognition of state-of-affairs which comprise not only the current situation but the appropriate history and the appropriate prediction of future developments.

No client can lean on services of a provider without cognition of provider's capabilities, provider's history and provider's intentions. Thus the mutual cognition in service provision is unavoidable.

A cognition is based on knowledge gathering, knowledge storing and knowledge utilization. Cognition process can be modeled in many ways. In any case it is heavy based on knowledge management. This is why chapter 4 is devoted to a deep explanation of knowledge management and cognition processes.

Cognition and knowledge have to be transferred and/or shared when used for service execution. This is the only known way how to co-produce a value within a co-creation process held by both–the provider and the client of the service. Knowledge transfer is technically performed by using some kind of language. An artificial or natural language is employed. Nevertheless which of these two kind is used the semantics of transferred messages is crucial. The provider and the client have to understand each other.

A wide accepted opinion is that dealing with semantics on computers must be simulated in a way by performing some magic tricks with syntax. The semantics hide something mysterious which is connected with questions like: What does cause I understand what I am seeing? Do I understand one situation so as you do? What is this what gives sense to things? We can put any description of a situation to a computer but the computer doesn't start to understand the situation. Why it is so? Is there any possibility to change this state-of-affairs? Will be possible to construct such a computer or SW program that would be able to deal with sense? Simply said, will a system (HW and/or SW) be able to „understand“?

Remember my last inspiration–Jeff Hawkins Book On Intelligence. My honest opinion is: yes, it will be possible. The semantics is maintainable on computers. And I have to say it is maintained. The technology components developed in my team[9] on the base of research conducted on Masaryk University (in my team again), namely in year 2007, are able to maintain the semantics in a way. A remaining question is: well, but what is semantics itself? Is this semantics maintained on computer the same one maintained in our brains? My hypothesis is that they are very closed. To explain this more precise the chapters 4–dealing with knowledge, concepts, domain understanding and modeling, and cognition and 5 on Diamonds like conceptual models of the understandable part of the world are included.

Why am I speaking on such a theoretical and „far from business“ topics? Will this gun be used somewhere in this book?[10] Yes, it will. The main reasons can be summarized as follows:

- „the business“ needs to understand behavior of service games players,
- understanding of behavior of all systems involved in service games is necessary for the ability to predict their future behavior,
- as Jeff Hawkins argues in his book, it is not enough to try to predict future behavior of an intelligent system based on the cognition of its previous behavior, only,
- the previous sentence (previous point) follows from the assumption that to predict a behavior of an intelligent system needs to understand its principles, not only a set of behavioral descriptions,
- I am absolutely sure the assumption is valid.

So, my point-of-view is that to describe what is the nature of service systems, how to model them, how to model execution of services within

[9] exactly Filip and my team

[10] A rule of a good drama is: when somewhere in description of environment there is written „... a gun hangs on the wall ...“ the gun must be used in the story.

such systems, how to evaluate value propositions connecting customers and providers in the frame of service systems, and what have to be learned by all types of agents employed in such service systems–a human ones or an artificial ones, we need to understand principles of semantics, maintenance the knowledge, and maintenance the information.

Knowledge and Information Robots History

The fourth inspiration I had mention before was knowledge and information robots research. It was not only an academical research conducted at university laboratory. It was encouraged by a serial of attempts to develop a business utilized product or solution. Research results of my student's team on the university were used in business practice in firms established for this purpose. And vice versa, business requirements discovered along the way of business solution creation influenced the research directions.

A brief history of this process follows and short conclusions are added explaining the crucial role of thinking on those two flows–the academic one and the business one–as on mutual service provision.

In the following each paragraph contains an action or product which had been done, together with a short explanation of its influence to our topic.

Years 2001-2002. System e-Dialog–CRM system for not computer experts that was demanded to be so flexible to be viewed by an Internet visitor according to his/her knowing of the firm owning the system.

We tried to build as much as possible general system to be adaptable for almost any (maybe not predicted) situation. This was done under the firm Shine Consulting which was in that time my private firm. This was the beginning.

Year 2002. Diamond of focus and the universal information recorder. Three technology component architecture.

Together with my master student at that time–Filip Prochazka–we had discovered an universal model enabling to store any kind of information in one and the same structure. Two o more years later we had found a paper from Dan Linstedt (see ⋇ [STA]$_{14}$:citace Linstedt)where this idea was enforced. The model we named „Diamond of Focus" and Filip programmed its first working implementation in Prolog. As a Prolog solution was unusable in business, Filip came with idea of three technology component architecture. The technology components were: Prolog for the inference machine, SQL database for the system's memory, and PHP for the communication interface.

The three technology approach gave arise the acronym „eTrium“. You will find Diamond of Focus in section 5.1.

Years 2002 - 2004. Universal Information Robots generations 0, 1-3, 4.

First we had call our product Universal Information Recorder (UIR) as it was based on the universal data model–the Diamond. We had created in a quick succession generations 0, 1, and 2. UIR generation 0 was this mentioned special CRM system eDialog for one local consultancy firm from Prague. This system is running till now (year 2008). No support is provided.

Generation 1 and 2 were about improving robustness of the system's database and about a switch from PHP to Java. We started to speak on our solution on conferences and we tried to catch some attention on market. We explained that we have an universal solution for everything. People looked on us as on green men.

Generation 3 was under a sign of changing the word „recorder“ into the word „robot“. A TV wanted to shoot the robot, but when I said to them it is a piece of software only, they told that it is not for them but for radio. First international presentation on IADIS conference ⋇ [STA]$_{15}$:cite IADIS ct. s Hrebickem z r. 2004was made and in few month a second one on ISES symposium at Harrissonburg, Virginia ⋇ [STA]$_{16}$:cite Harrissonburg. At this conference I spoke on universal information robots utilization in crisis management. The reactions were: „Very interesting, but near the fiction; we will wait“. We spoke about our product as about UIR = Universal Information Robot.

Till this point all activities were financed partly from two customers of the solution and partly from private resources of a small group (3 or 4) of enthusiasts. Then the eTrium Corporation was established, we found another enthusiast believing he is paying for his future and the core development team was formed (approx. 15 members).

From summer of 2004 to the end of that year the generation 4 was built as a first product having attributes of a technical proof of concept. I had found a first serious expert who believed the direction matters. It was[11] Vaclav Racansky, director of the Institute of Computer Science of Masaryk University. We together decided to make a submission into Academy of Science

[11] and still is

call for projects financed from the program „Information Society“. Vaclav put together consortium consisting from his Institute of Computer Science and from Institute of Biostatistics and Analysis of Masaryk University and from Masaryk Memorial Cancer Institute, the goal of which was to prove and validate a possibility to use tools of knowledge and information robots type in research and clinical practice in oncology. The submission was successful and from the January 2005 we started a 2-years research and development project, the so called Project UIRON (meaning UIR in Oncology).

Years 2005 - 2006. Project UIRON. UIR (Universal Information Robot), generation 5. We discovered that relational tables and objects as are used in relational and object-relational databases are not suitable for our intentions. We developed the so called Connection Oriented Approach (we focused not on objects but on connections), and we started to model domains of interest by using the so called „synapses“, i.e. a kind of binary connections. In these terms an object is simply a cluster of synapses. As this process of „creating“ objects is recursive one, we obtained a powerful hierarchy, where a more complex object is build on less complex objects. As we stored our synapses into tables of two or four columns (two columns for intended possibilities, four columns for real existing connections)the description of an usual domain costs thousands or millions rows. Those were very thin and very long tables formed of segments according to types of connected objects. Thus we called them bamboo tables and the whole database we called bamboo forest. The whole implementation was based on a fixed type system, which brought a necessity of a top ontology and a so called universal fixing point enabling to fix the universal system. One application–the project UIRON–has no problem with this, but we found out that it would be very difficult and worse almost impossible to integrate more such solutions into one consistent.

The UIRON project held in academic area was about research in possible applications of universal information robots and about a research what has to be developed. The parallel UIR project held in a private firm eTrium Corporation was about development of a business usable product fulfilling requirements discovered in the academic project.

The date 12/2006. End of project UIR and UIRON. Start of KIRLab on FI MUNI. Nevertheless the UIRON project was successfully finished and the possibility to use universal information robots in a nontrivial domain was proofed. Not the same happened to project UIR (development of universal information robot). The former enthusiastic business angel lost his enthusiasm and the project was stopped. On the base of successful UIRON project defense the Knowledge and Information Robots Laboratory (KIR Lab) was established at Faculty of informatics Masaryk University, and I was ordained its head.

The date 1/2007. Start of project Mycroft. The team was reduced to my recent master students and to my Ph.D. students. The others were spread over some temporary employments or stayed unemployed. Not be disturbed with business problems we started from zero point being richer in experiences. Our decision was: better than adapt our way of universal information robots development to the new discovered results and issues is to start a brave new way on a new basics. We reconciled our goals, issues and requirements together with our experience and we found a new project–the project Mycroft. Aim of this project was not (and is not) to develop a product. Its goal is to develop technologies on the principle of multi-agent systems enabling shorten time-to-market in solutions development according to customers demands. Such solution is a composite of knowledge of the development and application teams, technology platform, set of autonomous, cooperative and hierarchically composing-able agents, and of customer domain understanding and modeling. A parallel enabler of such solution is a competence of project, and program of projects, management. About this is the Mycroft project.

The interval 1-6/2007. Resarch in KIRLAb (Knowledge and Information Robots laboratory). First big shift in our thinking was a transition from one unique type system to the world of context dependent elementary facts with measure of belief and amount of attention paid to this fact in this selected context. Nowadays I prefer to say 'belief' instead of 'fact' in alignment with Jeff Hawkins (exactly saying there is no fact we can focus on, there are only beliefs, as the world, we live in–either the space-time world or the cyberspace–is much more probabilistic one than an exact zero-one world). This problem will be dealt in chapter 4, again. Together with Filip we developed the so called Diamond of cognitive elements (see the chapter 5). This helped us to revise our strategy of memory implementation, and we propose the so called T-space. The name is inspired by the shape of elementary construct (pictured as a diagram) assigning any item/element to a category in a chosen context (see 5.2).

At the same time we recognized that the success depends not only on a well managed projects and well managed program of projects, but it depends to the same measure on long-time attention paid to the projects crossing themes or topics. The following questions help to understand what I want to say: how much attention we pay to research, to specification, to implementation, and to production? At the whole and in each project separately. Is it enough or is it less then necessary or is it more then needed? This led to a scheme looking again as a Diamond (brought by Filip) which I fixed and defined as a well defined conceptual model (what it means you will find in 4.4). Thus we enriched our collection of diamonds by the so called Diamond of agent's team organization (the third Diamond, see section 5.3).

The last big achievement of this period was identifying of five basic building blocks–five agents or service supplying systems (service systems), that form „a special unit“, i.e. a team being able to solve most of typical business problems. Exactly saying there were only four agents and the so called architectural platform. A year later we had recognized that architectural platform is not something passive, but it is an active mediator of all communications in the system. From this time we speak on this former architectural platform as on agent-mediator, and we have really 5 basic agents in our service system.

Those agents or service systems (explication of this interchangeability will be done in chapter 3 where Service System is defined) are:

- Semantics Memory Engine
- Pattern Engine
- Visualization Engine
- Synthesizing Engine, and
- Mediator Engine (former architectural platform)

All these are agents or service systems or specialized robots co-operating on performing services for clients.

Besides that at the end of this period we found another business angels and we found a company Mycroft Mind, Inc. to have a body to develop technologies up to the in business utilized level.

As I have said before, this was a very productive time period.

The interval 7-12/2007. Pre-alpha version of Mycroft technologies and Service Science, Management, and Engineering on the stage. The whole team from previous periods was hired again and we started to develop a SW engineering proof of concept of the research results reached in KIRLab: it was the so called pre-alpha version of Mycroft technology and together with it we develop its first application–its utilization in a service system being a (tele)scope into computer networks–MyNetScope. This decision was partially formed by successful results in project CAMNEP (⋇ [STA]$_{17}$:cite CAMNEP), solved as a grant for USA Army, in which members of our team gathered experience (we were in this project as workers of ICT [12] MU, and MU was a subcontractor in this project under Czech Technical university from Prague) and understood the importance of this domain. At this time I added (until now) the last diamond-shaped conceptual model called Diamond of predictive behavior (you will find it in section 5.4). Those times I call it „Diamond of pro-active marketing and acting“. The new name is obviously influenced by my favorite book On Intelligence (⋇ [STA]$_{18}$:cite Hawkins).

During this year grew up faculty activities in SSME under moderate pressure of IBM ambassadors from Czech, Zurich, and Almaden service and research centers. I was asked to be a guarantor of the new, maybe somewhere in future being settled branch of study. Reading papers and presentations on SSME, namely Jim Spohrer articles, I had found this, what we did, we do, and we want to do with our knowledge and information robots or holonic-like agents, is just the subject of this large SSME movement. Moreover, we felt more and more that the problem of our new technologies development and utilization lies not in the technology itself, but in prepared and well trained people. Where to find well educated people to hire them into development and applications teams? Where to find well educated people by customers understanding a completely new approaches, a new paradigms of information service provision?

My answer is: if you need this, do it yourself. Thus, I upped the speed in SSME activities on our faculty, started international co-operations [13] and prepared a master degree study branch at FI MU completely in accordance with SSME on other world universities, and I can say, completely in accordance with our business needs.

[12] Institute of Computer Science leaded by previously mentioned Prof. Vaclav Racansky

[13] one of the most important was, and is, with Prof. Joao Falcao e Cunha from Porto University

The interval 1-6/2008. Business proof of concept: alpha versin of Mycroft technology. Jeff Hawkins's book „On Intelligence". The SW engineering proof of concept and the pre-alpha version of the MyNetScope were successfully finished in time, budget and expectations, and we build on this base a business proof of concept. The end of this period was with two pilots of our solution–one at Masaryk university and one in a big business company working in telecommunications.

The beginning of this period was marked by the book On Intelligence. Till this point I felt I am something like a green man striving to create a „circled square". Now I know, we are at least two. And, I must say, it is very pleasant to read about issues you have worked on more than 8 years, almost in the „same words", but from another side (not only of the ocean). That, what Jeff Hawkins discovered in neuro-science, I found in a way in logic–in the transparent intensional logic (more about it you will find in chapter 4). This „that" is the „Hierarchy". The hierarchical nature of the world. Hierarchy in space and hierarchy in time. Causes which are hierarchically build one upon the others, potentially with no limits.

The interval 7-12/2008. An interval of the start of writing of this book. Business solution–beta version of Mycroft technology–created. Master degree study branch SSME at our faculty accredited by the Czech Ministry of education. This way Masaryk university joined the few universities in the world, that has this full program included in their curricula (this is the state-of-affairs in 2008). All the ideas from the past epoch are step-by-step getting to be materialized. The endeavor of our team brings results. We know that we have just open „the door". The door into amazing world of thoughts, ideas and concepts. We really start to see what the cyberspace is. But this is no more than a beginning. I hope this book could be helpful on this future road. I tried to do it so.

Concluding lesson from the history. I would like to stress that the inspiration in KIRs research and development for a business utilization was not only the technical one. The technical point-of-view was a half of obtained experience, only. The second half was and still is the organizational, management, and engineering point-of-view. The little bit conflicted and complicated entrepreneurship experience focused my attention to questions like:

What is a benefit of business from an academic research?

What is a benefit of academic research from a business utilization of its results?

How to distinguish between this what is owned by academic part and what is owned by business part?

Is it possible to separate what belongs to university and what belongs to a firm when the solution depends on both sides and is created in mutual co-operation?

How to write good agreements between co-operating parties when information, knowledge and working methods have to be shared to co-produce an usable solution for a customer?

What is the base on which co-creating of value can be performed?

How to cope with existing rule of law[14] according which to provide free of charge a beta-version of a software product to a partner for testing and validating purposes is almost impossible?[15]

This experience and this attention focusing helped me very much in seeking of the consistent and as much as possible complete solution of my personal more than 30 years endeavor in the field of information, knowledge, management, engineering, and project/program/portfolio management. A rhetorical question on the end: Is there any other discipline that could help me instead of SSME=Service Science, Management, and Engineering?

Goals of this Work

The goals I would like to reach are:

- To put a comprehensive explication of a conceptual framework enabling design of services, executing a service for the benefit of a client and evaluate the added value of the performed service.

- To specify a support for this conceptual framework which can be materialized as a usable platform enabling to model a service and to execute a service both as an action of a service system.

[14] at least in Czech republic

[15] Financial managers and other financial wizards classify such free of charge provision as unauthorized enrichment on one side, but who will want to play a role of guinea pig and moreover to pay for it on the other side?

- To extend this framework to a consistent education program enabling to teach and to train individuals, teams, a composed teams of people and artificial agents to be well prepared and trained to a co-production of a value within a service execution.

I would like to transfer your attention from goods paradigm and a world of competition and of ownership to the service paradigm and a world of co-operation and of usability. I would like to convince you of possibility to organize agent teams, whatever the agents are (people, technological artifacts, etc.), to play interesting, useful, comprehensible, and effective and efficient games.

Most of all i would like to bring a comprehensive and consistent picture of value co-creation when the roles of a client and of a provider are dynamically exchanged between the academic world and the business world. In other words I like to bring a small contribution to a building an co-operative environment in which „to have one leg in theory and the other leg in practice“ is not viewed as to be a strange green man seeking for circled squares, but where it it is undisputable advantage.

What is not a goal of this work, is to write an easy readable book. The principles are quite simple, but the way to understand them is often difficult. Especially, when lot of common paradigms is inverted and new paradigms are introduced. So, when you don't like „to climb a heavy mountain“, please stop the reading. I don't need disturbed readers.

Work Structure Explanation

We start with Service Science, Management, and Engineering state-of-art from the academical and from the business point of view. Resources for this are ⋇ [STA]$_{19}$:cite CACM06? and others see DATAKON'08, presentations of colleagues from IBM, presentations and works of other frontiers in SSME. This is the chapter 2.

Third chapter brings a service system definition together with its extension to time dimension and mention-use operations. The explanation starts with understanding of a service and than explains what service system is. A concept „context“ is used here on an intuitive basis, only. Take it, please as a motivation for the next two chapters which are the most theoretical in this book. I know that people don't like theories at all. My opinion is, it is a mistake of our former teachers in all kinds of schools. Very often we have had to learn something strange from our point of view and nobody told us why, for what it is useful. However „... there is nothing better for practice then a good theory!“[16]

As models and modeling will be the crucial tool for our approach and our proposed framework and platform their role in cognition and co-operation is described and explicated. Starting with Jeff Hawkins' Memory-Prediction Model and his hypothesis what the intelligence is, we introduce Universal Modeling and the Mention-Use Complementarity based on my dissertation and fix all this in services and service systems. The role of prediction will be emphasized, namely. But as the thinking will be dealt with so often and a basis for thinking are concepts, conceptual systems and the language we need to spend a nontrivial time and space in the theory of that. Without understanding a domain and proper modeling this domain no service can be provided and no service system can be constructed; that is why a section on a practically proofed semantics modeling (but seemed to be almost theoretical one) is included.

Then a very new conceptual top-down framework of Diamond path is introduced and explicated. The first diamond, Diamond of focus, I had introduced in my dissertation in 2003. The second diamond, Diamond of cognitive elements, we introduced with Filip Prochazka near the beginning of year 2007. Third diamond, Diamond of agent team organization, had appeared somewhere in the middle of 2007 and I had specified it exactly according the project, program, and portfolio management pattern. The fourth diamond, Diamond of predictive behavior, I had introduced in September 2007 as a result of my thoughts on marketing strategy of solutions based on our technology of knowledge and information robots. The fifth diamond, a Diamond of Diamonds, I had developed in October 2007 as a covering wrapper for a complete theory, practice, and learning/training field of something which a month later I started to call Services Systems and for which I found a very natural embedding into SSME.

This is the content of chapters 4 and 5.

Chapter 6 turns us back to more practical issues. Its aim is to explain one possible and proven way how to model a service system in its structure, its elementary functioning, how to model its organization and its predictive

[16]This is Dines Bjorner sentence I had heard more than quarter of century before on the former well-known software seminar SOFSEM in earstwile Czechoslovakia. This sentence was for me so important that I can say this drew and drive me through my professional career.

behavior. The whole service system life-cycle modeling is proposed.

Having defined service systems and knowing what models and cognition have in common in the Diamond Path conceptual framework we can propose a platform for Service Systems modeling and execution. Features and aspects of this framework is discussed together with basic use-cases and a description of existing SW engineering proof of this concept is briefly outlined. This is a content of chapter 7.

Service Science, service models (so as cognition, co-operation, prediction, and the Diamond path framework), service system definition, service system platform specification, all these are interesting topics in academical world. The business world could be a little bit impatient: How to use this service systems? What they bring to customers? What kind of added value will be created? Why clients will pay for such a things? A framework of service system usability based on third and fourth diamonds is explained in chapter **??**.

The usual approach is to stop with explanations and talks on the point reached in chapter **??**. My thirty years practice as a consultant helping with corporate strategies, information strategies, system integration, and effective and efficient business processes tuning in various kinds of companies, together with my five year experience in building and leading very innovative project team developing business solutions upon theory and scientific results in Knowledge and Information Robots Laboratory tell me: no, it is not enough. There is one other thing, the absence of which leads to unsuccessful attempts. This thing is education, education in very wide sense. No successful solution can be systematically obtained in service systems utilization without well and proper educated staff. This was recognized by IBM researchers and leaders and this is why IBM pays so much effort to SSME = Service Science, Management, and Engineering as to an academical discipline. In chapter 8 I start with strategical and philosophical contemplations, continue through people education to be a usable part of service systems, and conclude with education of a service system, as it is a „recursive composite“ of people, technology, knowledge, and information.

The last chapter of the body of my work, chapter 9, is about the last big theme, which, I am sure, have to be addressed when speaking on service systems. However, in opposite to previous chapters this chapter is much more about ideas than about proved concepts and proved techniques and methods. The chapter on service system evaluation collects my ideas and visions, the proof of which is now under the running.

At the Conclusion I try to justify the „*“ in the work title. Except the „E“ standing for Execution, Education, Evaluation, the „E“ can stay for Evolution, Environment, etc. Only a case of Evolution is briefly outlined.

My aim was to put down a comprehensive and in a sense complete monograph on service systems. Most of the opinions written here was proved in a co-operative work with my thirty members team on knowledge and information robots technologies, part of which working in KIRLab and most of them working in a spin-off company „Mycroft Mind“ of Masaryk University. I hope the monograph will become a useful guide in some domain of service science, management, and engineering.

Chapter 2

Service Science

⋇ [STA]$_{20}$:PRELIMINARY VERSION. WILL BE COMPLETELY RE-WORKED JUST BEFORE THE „INTRODUCTION“ REVISION.

Inspiration and resource for this chapter were papers and presentations of Jim Spohrer and other IBM Almaden Research Center and IBM generally colleagues together with a series of papers, presentations and studies on SSME from all over the world.

A second inspiration was the realization of similarity between the notion of knowledge and information robots and a concept of service systems.

2.1 Towards Service Science

TODO based on

Jim Spohrer,

University of Cambridge - Succeeding through service innovation

maps from mapPool Lab svn ... -Projects-SSME

namely a STA map Towards the Services Science

CACM IBM special issue

SME Thesis – Done ! Doplnit, Upravit

DUS thesis

book: SSME - education for the 21st century

⋇ [STA]$_{21}$:cited from SME Thesis

It took a very long time when services become dominant in world developed economies. At the beginning, people had to gather or hunt to stay alive. As it was very demanding, people realized that they would obtain more if they try to make food and tools by themselves. The wealth came from the earth that time, especially from agriculture, mining or breeding. The sources were limited and therefore there was a strong competition of them. Later, the richest countries, which were able to process and transform the sources, became richer using factories and manufactures. Although the value of a man raised, people were still replaceable by mechanical machines. It is not long time ago when people started to earn most of the money from services. The focus is on intellectual work now and it is a question what will come later.

Another question follows – how should we ensure that the world's economy will grow in the future? A factory production is higher and higher to keep their profits growing. The supply is wider and wider and, as a result of that, we can choose from more and more types of products. And what will happen then? We will not be able to eat 100 rolls per day as well as we will not be able to drive four cars simultaneously.

The answer resides in services. In services area, the limitations are not so strict so that services provide us a large space for economic growth.

The described development goes from the material and tangible things to intellectual, intangible and abstract areas. The focus moves from an ownership of a product to gain value from services, no matter if the service is provided by the product which I own or it is provided by something or someone else. The traditional idea of product exchange meaning "I have something, I trade it, I receive money and the process of interaction is finished" is shifting to the service idea meaning a long-term interaction between service providers and service clients.

Governments are aware of the increasing importance of services in many countries. And it is not only the importance of services itself, but it is also crucial to study services and establish service science as a real science taught and researched at universities. In the United States, US House and Senate voted to approve "The U.S. National Innovation Investment Act" on August 2nd 2007 and the president has signed it [12]. The text can be seen in Figure 2.1.

2.1.1 History of Economic Paradigms

The shifts in economic thinking and practises have been outlined in the first introductory chapter. One possible approach to the paradigms development is illustrated in Table 2.1.

SEC. 1106. STUDY OF SERVICE SCIENCE.

(a) Sense of Congress: It is the sense of Congress that, in order to strengthen the competitiveness of United States enterprises and institutions and to prepare the people of the United States for high-wage, high-skill employment, the Federal Government should better understand and respond strategically to the emerging management and learning discipline known as service science.

(b) Study: Not later than 270 days after the date of enactment of this Act, the Director of the Office of Science and Technology Policy, through the National Academy of Sciences, shall conduct a study and report to Congress regarding how the Federal Government should support, through research, education, and training, the emerging management and learning discipline known as service science.

(c) Outside Resources: In conducting the study under subsection (b), the National Academy of Sciences shall consult with leaders from 2-and 4-year institutions of higher education, as defined in section 101(a) of the Higher Education Act of 1965 (20 U.S.C. 1001(a)), leaders from corporations, and other relevant parties.

(d) Service Science Defined: In this section, the term "service science" means curricula, training, and research programs that are designed to teach individuals to apply scientific, engineering, and management disciplines that integrate elements of computer science, operations research, industrial engineering, business strategy, management sciences, and social and legal sciences, in order to encourage innovation in how organizations create value for customers and shareholders that could not be achieved through such disciplines working in isolation.

Figure 2.1: The U.S. National Innovation Investment Act [?]

Economic Offering	*Commodity Goods*	*Packaged Goods*	*Commodity Services*	*Consumer Services++*	*Business Services++*
Economy	Agrarian	Industrial	Service	Experience	Transformation
Economic Function	Extract	Make	Deliver	Stage	Co-create value growth
Nature of Offering	Fungible	Tangible	Intangible	Memorable	Effectual
Key Attribute	Natural	Standard	Custom	Personal	Value growth relationship
Method of Supply	Stored in Bulk	Inventory of product	Delivered on demand	Reveal over duration	Sustained over time
Settler	Trader	Manufacturer	Provider	Stager	Collaborator
Buyer	Market	Customer	Client	Guest	Collaborator
Factors of Demand	Characteristics	Features	Benefits	Sensations	Capabilities (cultural values)

Table 2.1: History of Economic Paradigms [?]

There cannot be any doubt about the first two product economy phases – agrarian and industrial. Although it seems that we live in service economy now, some authors, especially Prahalad and Ramaswamy [?] show that many firms base their strategies on co-creating value with customers. The definition of service used in SSME initiative also deals with value co-creation.

It seems that the third phase (services) is a base for the other two phases meaning the nature of service determines the result of its impact. As the agrarian, industrial and service phases were not strictly isolated, we could expect that the situation will be similar in the next phases.

2.1.2 Product Economy Paradigms

The main principle of the product economy (i.e. product-oriented or product-based economy) resides in a tangible product exchange between a product supplier and a product consumer. The paradigm comes from the era, when it was considered that wealth is created only by a production of tangible commodities. The simplified product life-cycle in the product economy can be summarized as follows:

1. A manufacturer develops a product.
2. The manufacturer makes the product.
3. The product is given to the market.
4. A consumer buys the product.
5. The consumer uses the product.
6. The supplier eventually provides additional support of the product.
7. The consumer gets rid of the product.

As we can see, the process is easily conceivable. The role of the producer and the consumer is exactly defined in every step of the process as well as the ownership of the product. The process could be considered as an ownership transfer – from the producer to the consumer. The producer and buyer are not closely connected during the whole life-cycle. Basically, they have to be in touch only at the moment of the ownership transfer.

Products are tangible, therefore it is not usually so hard to convert a product value into money. At least we are able to set the price on the market according to the production costs. The major task in production is an optimization of production quantity according to fixed and variable costs to achieve maximum profit.

2.1.3 Service Economy Paradigms

On the other side, the service economy paradigm is different. The emphasis is not laid on the tangible products, it is laid on services which a customer can get. No matter if the service is realized through a product or someone else performs the service. The result is important as well as the experience which a customer undergoes. To deal with ownership is not needed primary, service customer obtain benefits by renting the right to use a physical object, to hire the labour and expertise of personnel, or to pay for access to facilities and networks [**?**].

The second crucial issue of the service paradigm is the act of customer cooperation during the process of service. The service provider creates the value together with the service consumer and the both sides learn and influence each other.

As Gummesson wrote even in 1994 [?], customers do not buy goods or services: they buy offerings which render services which create value. The traditional division between goods and services is long outdated. It is not a matter of redefining services and seeing them from a customer perspective; activities render services, things render services. The shift in focus to services is a shift from the means and the producer perspective to the utilization and the customer perspective.

2.1.4 Several Definitions of Service

Various sources understand and define a service concept differently. This section introduces some of them in order to compare their scope and choose the most proper one.

The word service was originally associated with the work that servants did for their masters [?]. In time, a broader association emerged, captured in the dictionary definition of "the action of serving, helping, or benefiting; conduct tending to the welfare or advantage of another [?]".

The more complex, general definition of the service term reads as follows [?]: A service is a set of benefits delivered from the accountable service provider, mostly in close coaction with his service suppliers, generated by the functions of technical systems and/or by distinct activities of individuals, respectively, commissioned according to the needs of his service consumers by the service customer from the accountable service provider, rendered individually to the authorized service consumers on their dedicated request, and, finally, utilized by the requesting service consumers for executing and/or supporting their day-to-day business tasks or private activities.

Lovelock [?] defines service as economic activities offered by one party to another, most commonly employing time-based performances to bring about desired results in recipients themselves or in objects or other assets for which purchasers have responsibility. In exchange for their money, time, and effort, service customers expect to obtain value from access to goods, labor, professional skills, facilities, networks, and systems; but they do not normally take ownership of any of the physical elements involved.

Vargo and Lusch [18] define services as the application of specialized competencies (knowledge and skills) through deeds, processes, and performances for benefit of another entity or the entity itself.

Spohrer [12] defines service as the application of competencies for the benefit of another, meaning that service is a kind of action, performance, or promise that's exchanged for value between provider and client. He uses the definition based on Gadrey [?], which is illustrated in Figure 2.2.

Services are anything of economic value that cannot be dropped on your foot – the key to service value is in actions, performed now or promised for the future. Services transform/protect or promise to transform/protect a state of the target of the service. The client may not have the skill, time, desire, or authority to perform self-service, do it themselves. Services often create mutual interdependencies. Service is performed in close contact with a client; the more knowledge-intensive and customized the service, the more the service process depends critically on client participation and input, whether by providing labor, property, or information [12].

Services are value coproduction performances and promises between clients and providers, with alternative work sharing, risk sharing, information sharing, asset sharing, and decision sharing arrangements and relationships (promises to perform now or in the future, once or repeatedly, when needed or demanded, standard or customized, satisfaction guaranteed or best effort, service levels fixed or variable).

The true meaning and implications of services will be completely expanded only in a context of service system.

A service system comprises people and technologies that adaptively compute and adjust to a system's changing value of knowledge [13]. It comprises service providers and service clients working together to coproduce value in complex value chains or networks. Providers and clients might be individuals, firms, government agencies, or any organization of people and technologies.

The key is that providers and clients work together to create value. The client owns or controls some state that the provider is responsible for transforming according to some agreement between provider and client.

Individuals, families, firms, nations, and economies all represent instances of service systems [13]. The illustration of such service system can be seen in Figure 2.3.

Service system complexity is a function of the number and variety of people, technologies, and organizations linked in the value creation networks, ranging in scale from professional reputation systems of a single kind of knowledge worker or profession, to work systems composed of multiple types of knowledge workers, to enterprise systems, to industrial systems, to national systems, and ultimately to the global service system [?].

Most of the time, real-world competencies of great value are not simple. Some competencies even might have side effects and associated risks to other

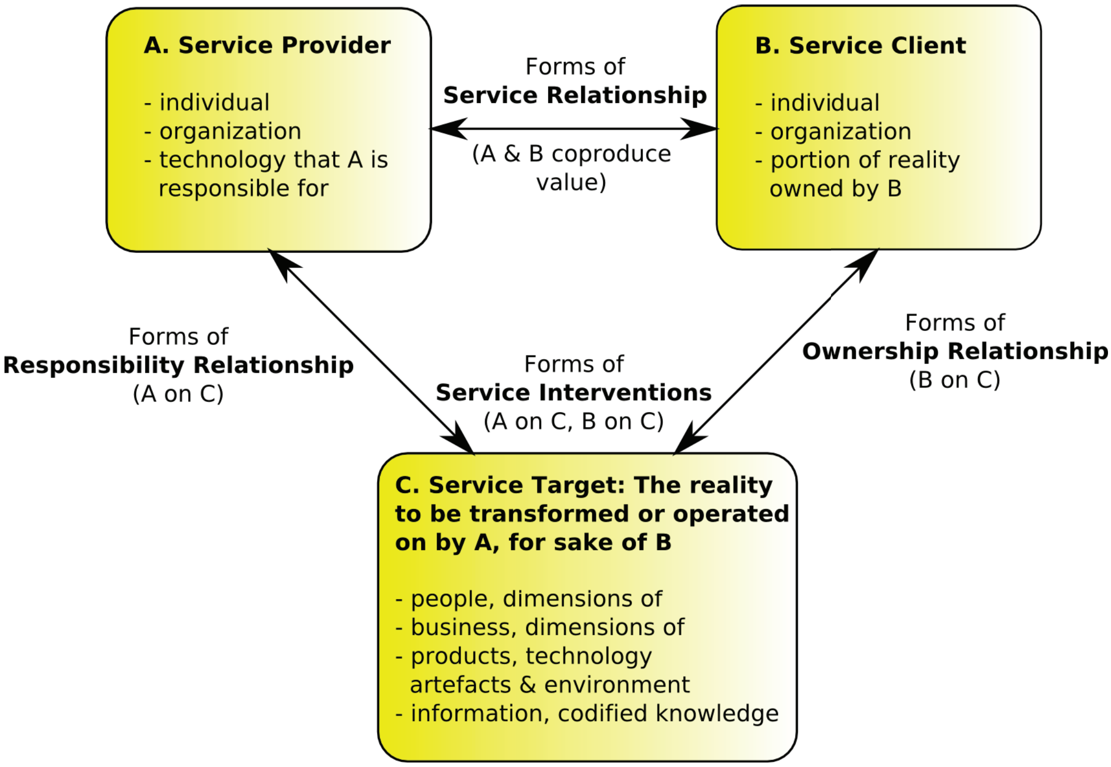

Figure 2.2: Service Illustration [?]

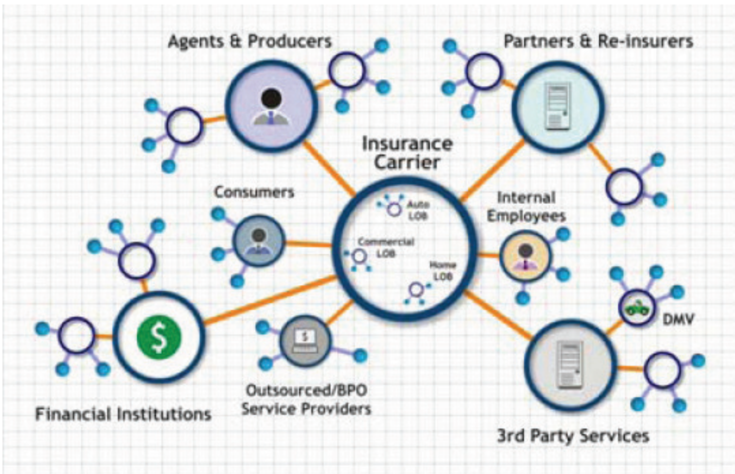

Figure 2.3: Service System Illustration [?]

service systems if not executed properly. Regardless of how competence leads to action and value, coordination and governance require shared information. Three key types of shared information are language, laws, and measures [13]:

Language Without some form of language, signaling, or standard encoding of information, systems would find coordination difficult, leading to missed opportunities for innovation or efficiency gains.

Laws Provisioning sophisticated service and maintaining complex service systems requires laws and contracts. Typically, every service system has a governing authority that seeks to ensure that all those in the service system can communicate in shared languages and abide by shared laws. In firms, it is the CEO and board of directors; and in nations, it is government leaders and agencies, as well as shared legal documents and enforcement agencies.

Measures Prices are one type of measure of the value of services exchanged within or between service systems. Often, standardizing the sets of measures used within and between service systems improves the productive capacity of the system by eliminating unneeded transaction costs and improving coordination.

Language, laws, measures, and other types of shared information evolve over time as service systems invest to improve productivity, quality, compliance, and innovation.

2.1.5 Shift to Service Economy

Vargo and Lusch [18] illustrate the shift to service-centered view in Figure 2.4. The distinction between mentioned two views is obvious – the emphasis shifts from tangible, statics, discrete transactions, and operand resources to intangibles, competencies, dynamics, exchange processes and relationships, and operant resources.

Operand and Operant Resources

Over the past 50 years, resources have come to be viewed not only as stuff but also as intangible and dynamic functions of human ingenuity and appraisal,

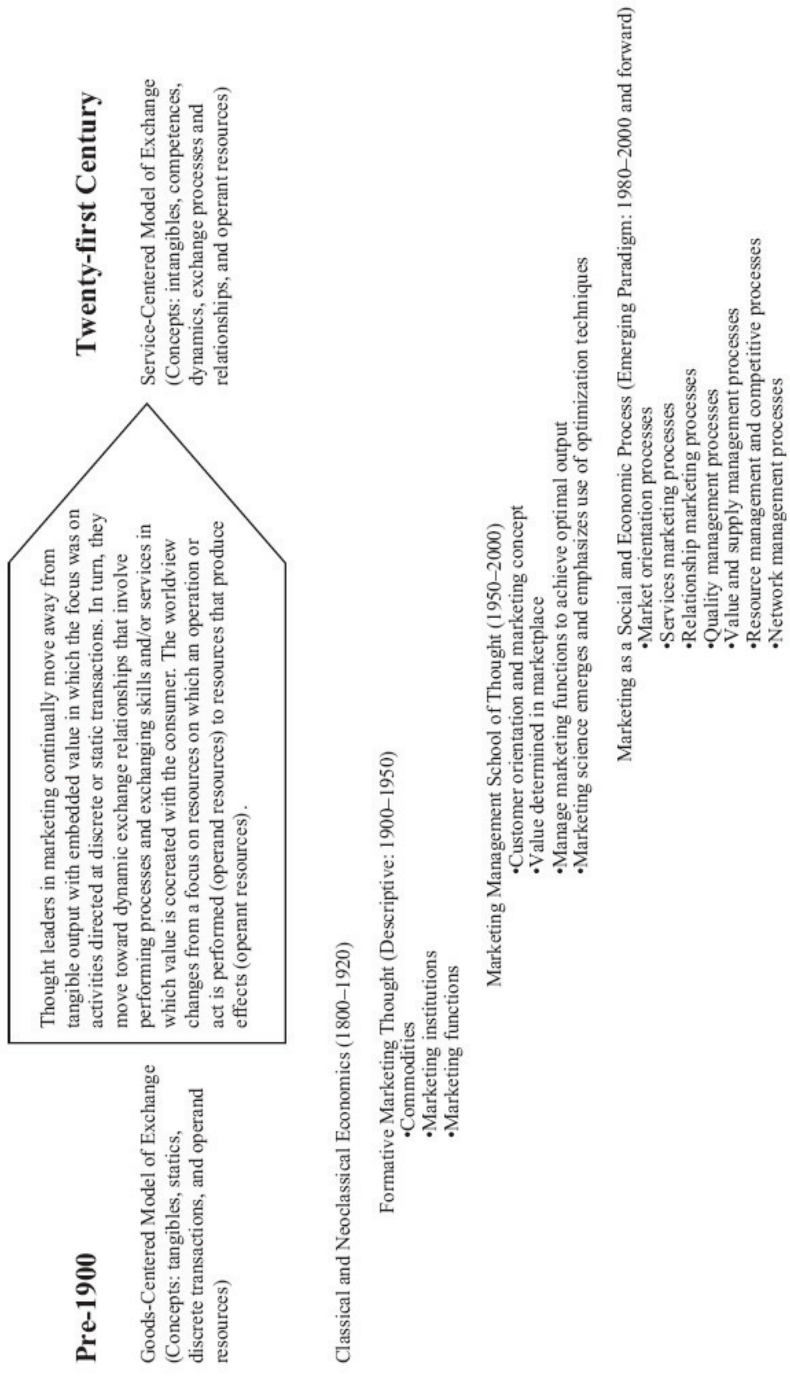

Figure 2.4: From Goods-Centered to Service-Centered View [18]

and thus they are not static or fixed. Everything is neutral until humankind learns what to do with it. Essentially, resources *are* not; they *become* [18].

Vargo and Lush illustrate the change to service-centered view with the role of operand and operant resources [18]:

Operand resources are resources on which an operation or act is performed to produce an effect (eg. goods, minerals, animal life, plant life, and other natural resources)

Operant resources are resources which produce effects, they are employed to act on operand resources and other operant resources (eg. knowledge and skills, technology).

The shift to the service paradigm explained by a role of operant and operand resources is illustrated in Figure 2.5.

In the product-oriented economy the operand resources were considered primary. A firm (or nation) had factors of production (largely operand resources) and a technology (an operant resource), which had value to the extent that the firm could convert its operand resources into outputs at a low cost. Customers, like resources, became something to be captured or acted on [18].

The relative role of operant resources began to shift in the late twentieth century as humans began to realize that skills and knowledge were the most important types of resources. It is never resources themselves that are the "inputs" to the production process, but only the services that the resources can render. Operant resources are often invisible and intangible; often they are core competencies. They are likely to be dynamic and infinite and not static and finite, as is usually the case with operand resources. Because operant resources produce effects, they enable humans both to multiply the value of natural resources and to create additional operant resources [18].

The service-oriented economy perceives operant resources as primary, because they are the producers of effects. This shift in the primacy of resources has implications for how exchange processes, markets, and customers are perceived and approached [18].

Eight Foundational Premises of Service-Centered View

Vargo and Lush introduced eight premises of service-centered view (adapted from [18]):

Operand and Operant Resources Help Distinguish the Logic of the Goods- and Service-Centered Views

	Traditional Goods-Centered Dominant Logic	Emerging Service-Centered Dominant Logic
Primary unit of exchange	People exchange for goods. These goods serve primarily as *operand resources*.	People exchange to acquire the benefits of specialized competences (knowledge and skills), or services. Knowledge and skills are *operant resources*.
Role of goods	Goods are *operand resources* and end products. Marketers take matter and change its form, place, time, and possession.	Goods are transmitters of *operant resources* (embedded knowledge); they are intermediate "products" that are used by other operant resources (customers) as appliances in value-creation processes.
Role of customer	The customer is the recipient of goods. Marketers do things to customers; they segment them, penetrate them, distribute to them, and promote to them. The customer is an *operand resource*.	The customer is a coproducer of service. Marketing is a process of doing things in interaction with the customer. The customer is primarily an *operant resource*, only functioning occasionally as an operand resource.
Determination and meaning of value	Value is determined by the producer. It is embedded in the *operand resource (goods) and is defined in terms of* "exchange-value."	Value is perceived and determined by the consumer on the basis of "value in use." Value results from the beneficial application of *operant resources sometimes* transmitted through *operand resources*. Firms can only make value propositions.
Firm–customer interaction	The customer is an *operand resource*. Customers are acted on to create transactions with resources.	The customer is primarily an *operant resource*. Customers are active participants in relational exchanges and coproduction.
Source of economic growth	Wealth is obtained from surplus tangible resources and goods. Wealth consists of owning, controlling, and producing *operand resources*.	Wealth is obtained through the application and exchange of specialized knowledge and skills. It represents the right to the future use of *operant resources*.

Figure 2.5: Operant and Operand Resources in Goods- and Service-Centered Views [18]

1. The Application of Specialized Skills and Knowledge Is the Fundamental Unit of Exchange

   People have two basic operant resources: physical and mental skills. Both types of skills are distributed unequally in a population. Each person's skills are not necessarily optimal for his or her survival and well-being; therefore, specialization is more efficient for society and for individual members of society. Largely because they specialize in particular skills, people (or other entities) achieve scale effects. This specialization requires exchange.

2. Indirect Exchange Masks the Fundamental Unit of Exchange

   Over time, exchange moved from the one-to-one trading of specialized skills to the indirect exchange of skills in vertical marketing systems and increasingly large, bureaucratic, hierarchical organizations. Consequently, the inherent focus on the customer as a direct trading partner largely disappeared and most marketing personnel (and employees in general) stopped interacting with customers. In addition, because of the confluence of these forces, the skills-for-skills (services-for-services) nature of exchange became masked.

3. Goods Are Distribution Mechanisms for Service Provision

   Goods are not the common denominator of exchange; the common denominator is the application of specialized knowledge, mental skills, and, to a lesser extent, physical labor (physical skills). Knowledge and skills can be transferred (1) *directly*, (2) *through education or training*, or (3) *indirectly by embedding them in objects*. Thus, tangible products can be viewed as embodied knowledge or activities.

4. Knowledge Is the Fundamental Source of Competitive Advantage

   Knowledge is an operant resource. It is the foundation of competitive advantage and economic growth and the key source of wealth. Knowledge is composed of propositional knowledge, which is often referred to as abstract and generalized, and prescriptive knowledge, which is often referred to as techniques. The techniques are the skills and competencies that entities use to gain competitive advantage. This view is consistent with current economic thought that the change in a firm's productivity is primarily dependent on knowledge or technology.

5. All Economies Are Services Economies

   Formal economic thought developed during the industrial economy, and it has tended to describe economies in terms of the types of output, or operand resources (game, agricultural products, and manufactured products), associated with markets that were expanding rapidly at the time. However, the "economies" might be better viewed as macrospecializations, each characterized by the expansion and refinement of some particular type of competence (operant resource) that could be exchanged:

   - The hunter-gatherer macrospecialization was characterized by the refinement and application of foraging and hunting knowledge and skills.
   - The agricultural macrospecialization by the cultivation of knowledge and skills.
   - The industrial economy by the refinement of knowledge and skills for large-scale mass production and organizational management.
   - The services and information economies by the refinement and use of knowledge and skills about information and the exchange of pure, unembedded knowledge.

   Services and the operant resources they represent have always characterized the essence of economic activity.

6. The Customer Is Always a Coproducer

   From the traditional, goods-based, manufacturing perspective, the producer and consumer are usually viewed as ideally separated in order to enable maximum manufacturing efficiency.

   From a service-centered view of marketing with a heavy focus on continuous processes, the consumer is always involved in the production of value. Even with tangible goods, production does not end with the manufacturing process; production is an intermediary process. As we have noted, goods are appliances that provide services for and in conjunction with the consumer. However, for these services to be delivered, the customer still must learn to use, maintain, repair, and adapt the appliance to his or her unique needs, usage situation, and behaviors. The customer is continuing the marketing, consumption, and value-creation and service delivery processes in using a product.

In summary, the customer becomes primarily an operant resource (co-producer) rather than an operand resource ("target") and can be involved in the entire value and service chain in acting on operand resources.

7. The Enterprise Can Only Make Value Propositions

   The enterprise can only offer value propositions; the consumer must determine value and participate in creating it through the process of coproduction. If a tangible good is part of the offering, it is embedded with knowledge that has value potential for the intended consumer, but it is not embedded with value (utility). The consumer must understand that the value potential is translatable to specific needs through coproduction. The enterprise can only make value propositions that strive to be better or more compelling than those of competitors.

8. A Service-Centered View Is Customer Oriented and Relational

   The customer-interaction process begins with the interactive definition of the individual customers' problem, the development of a customized solution, and delivery of that customized solution to the customer. The solution may consist of a tangible product, an intangible service, or some combination of both. It is not the mix of the solution (be it product or service) that is important, but that the organization interacts with each customer to define the specific need and then develops a solution to meet the need.

   It is in this sense of doing things, not just for the customer but also in concert with the customer, that the service-centered view emerges. It is a model of inseparability of the one who offers (and the offer) and the consumer. We also suggest that the interactive and integrative view of exchange is more compatible with the other normative elements of the marketing concept, the idea that all activities of the firm be integrated in their market responsiveness and the idea that profits come from customer satisfaction (rather than units of goods sold).

A service-centered view of exchange points in an opposing normative direction. It implies that the goal is to customize offerings, to recognize that the consumer is always a coproducer, and to strive to maximize consumer involvement in the customization to better fit his or her needs. A service-centered view identifies operant resources, especially higher-order, core competencies, as the key to obtaining competitive advantage. It also implies that the resources must be developed and coordinated to provide (to serve) desired benefits for customers, either directly or indirectly [18].

2.1.6 Business As Service System

Considering eight foundational premises of service-centered view introduced in the previous section, especially the fifth premise meaning "all economies are services economies", we could assume that business in the service economy will exactly satisfy the definition of service system (see section **??**). Assuming that, many interesting implications appears (adapted from [12] and [18]):

The importance of strategies First, and most fundamentally, organizations and business strategy can be as important as technology, cost, and demand in determining a firm's success.

The source of competitive advantage Only true source of competitive advantage is the ability to conceive the entire value creating system and make it work.

The importance of marketing strategy The new paradigm is service-oriented, customer-oriented, relationship-focused, and knowledge-based, and places marketing, once viewed as a support function, central to overall business strategy.

Organizational structures in service systems The study of organization is not about how berries are arranged on a tree of authority, but about how people are coordinated and motivated to get things done.

Coordination What needs to be coordinated, how coordination is achieved in markets and inside firms, what the alternatives are to close coordination between units, and how the pieces of the system fit together.

Incentives and motivation What needs to be motivated, why incentives are needed, and how they are provided by markets and firms, what alternative kinds of incentive systems are possible, and what needs to be done to make incentive systems effective.

2.1.7 Service Systems As Computational Systems

Because IT is such an important part of service systems today, we might ask how service systems are similar to and different from computational systems [13]:

The main difference is people. The largest service system, the global economy, includes more than six billion people. Some large firms have hundreds of thousands of employees. People do a lot of the work – physical, mental, and social. Furthermore, unlike computational system components, we cannot easily model and simulate the behavior of people doing work in service systems. For example, laws and policies only partially govern people. Even when citizens and employees know government laws and corporate policies, compliance is not complete, which creates risk as well as opportunities.

Spohrer also introduces a very interesting problem [13]: Many innovations break a rule or violate a policy. How can we tell the difference between cheating and innovation in a service system, where people informally and formally change rules and policies?

The other emerging problems in modeling service systems formally [12]:

- Accelerating economic, technological, social, and environmental change challenge managers and policy makers to learn at increasing rates, while at the same time the complexity of the systems in which we live are growing. Many of the problems we now face arise from unanticipated side effects of our own past actions.

- Dynamic complexity arises because systems are: Dynamic, tightly coupled, governed by feedback, nonlinear, history dependent, self organizing, adaptive, counterintuitive, policy resistant, and characterized by trade-offs

- How rapid is the change and are there any patterns in how humans deal with complexity, how do people invest their time?

Spohrer also tries to compare service science and grid computing for the first time in the paper [**?**]. Although he introduced several distinctive similarities there, he concludes with many issues which have to be solved in the future.

2.1.8 Services Science, Management and Engineering Initiative

Many innovative companies (such as IBM, Accenture, HP, EDS, CSC, Cisco, P&G, American Express, John Deere, Avaya, Oracle, and many others) are aware of emerging importance of services. Therefore, the IBM company has established the Services Science, Management, and Engineering (SSME) initiative[1]. It is one of the materializations of service approach and IBM calls it as "an urgent call to action" [12]. The major goals of the initiative are:

- to become more systematic about innovation in services
- to complement product and process innovation methods
- to develop "a science of services"

Although several theories have identified the building blocks of service systems, researchers have not yet developed a theory of service systems [13]. Therefore, the proposed academic discipline draws on many disciplines and aims to integrate them into a new speciality. The proposed research areas should deal with various disciplines such as [12]:

- computer systems (how service systems are designed)
- linguistic and social systems (how service systems evolve)
- economic systems (how service systems have scale-emergent properties)

The proposal of a general theory of service systems should consist of three parts [13]:

- science: what service systems are and how to understand their evolution; it is a way to create knowledge
- management: how to invest to improve service systems; it also improves the process of creating and capturing value
- engineering: how to invent new technologies that improve the scaling of service systems; it is a way to apply knowledge and create new value

[1] http://www.research.ibm.com/ssme/

The application of scientific, management, and engineering disciplines to services should help to [12]:

- understand the evolution and design of service systems
- make productivity, quality, compliance, attainability, and innovation rates more predictable, especially complex organization to organization services – business to business, nation to nation, organization to population
- invest in service systems to make them into double-loop learning systems

The focus on SSME is important for many parts of economy service systems. Governments need to make service innovation a priority, because GDP growth of nations increasingly depends on it. In businesses, revenue and profit growth also increasingly depend on service innovation. And academics also need to make service innovation a priority because improved education productivity and quality depends on the SSME disciplines as well as a new frontier of research with business and societal impact. The SSME also emphasizes the cooperation of governments, business and academics through win-win strategy.

The SSME initiative also points out the need for people educated in more disciplines. They call them as T-shaped professionals (Figure 2.6), which are supposed to be both "deep and broad". Deep means that they should have deep knowledge in a selected discipline (the core field of study) and broad aims to multidisciplinary overlaps (communication skills across other fields). The wider scope of T-shaped people is necessary due to an ability to deal with multidisciplinary nature of service affairs.

As mentioned in this section many times, the SSME initiative lays stress on the multidisciplinary approach. Hence, the service science touches many areas and disciplines. This approach implies that if we want to deal with the emerging service paradigm in a context of a particular discipline, we cannot avoid investigating its relationships to many adjacent areas. The relationship matters, therefore it is important to introduce connections between different views of services, strategies and derive new consequences.

⋇ [STA]$_{22}$:end of SME Thesis citation

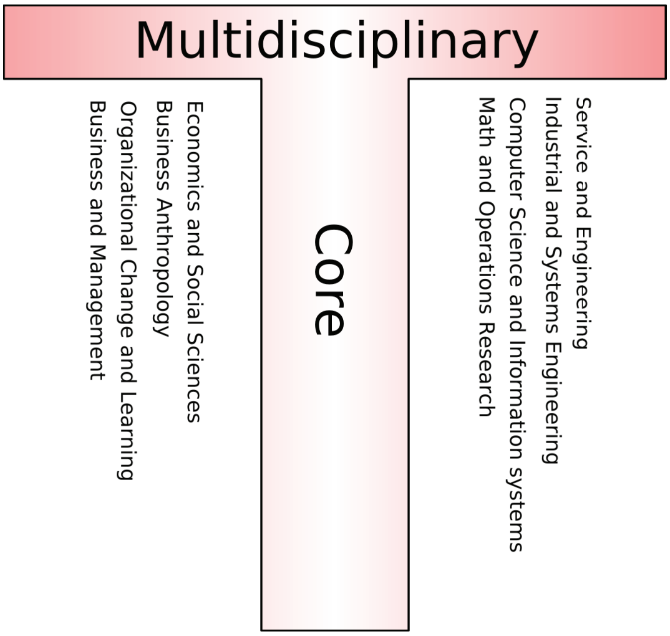

Figure 2.6: T-shaped Professional [?]

2.2 Academy and Business Co-operation

TODO based on
survey of SSME related universities and their approach
maps from mapPool Lab svn ... -Projects-SSME
book: SSME - education for the 21st century

2.2.1 SSME Curricula

TODO

2.2.1.1 University of Porto, Portugal

A master study program devoted to SSME in Porto is called MESG.

The Master in Service Engineering and Management (MESG) aims at developing competences to conceive, design, implement and operate technology enabled service systems. MESG is directed to graduates in Engineering, Science, Economics or Management, who want to develop their career in service development, design and management, either in service or in manufacturing industries [⁎ [STA]$_{23}$:citace stranka FEUP http://gnomo.fe.up.pt/ mesg/].

The study plan of MESG comprises three semesters with courses, including individual and group projects, and one semester for a master dissertation, either a research project at FEUP or an internship project in a company. Each of the three semesters involves elective courses, in a total of five disciplines per semester. The plan of studies can be adapted to each student's background and desired specialization profile. The program offers the following courses:

1st Year

1st Semester

Management, Business Process Modeling, Information Systems, Organizational Behavior, Decision Support Systems, Human-Computer Interaction, Data Base Systems and Information Management, Creativity

2nd Semester

Service Operations Management and Logistics, Services Marketing, Accounting and Financial Management , Requirements Engineering for Services, Multimedia and New Services, Human Resources Management , Cognitive Psychology

2nd Year

1st Semester

New Service Development and Design, Information Systems Architecture, Capital Budgeting, Enterprise Management Architecture, Corporate Strategy, Customer Relationship Management, Project Management Laboratory, E-Business Technologies, Introduction to Research Project

2nd Semester

Dissertation – Research or Internship Project

[citace stranka http://gnomo.fe.up.pt/ mesg/studyplan.html]

2.2.1.2 University of California, Berkeley

The following text is a quotation from the university web pages.

The Information and Service Design (ISD) program was established at UC Berkeley's School of Information in 2007 to provide a focus for teaching and research on the skills and concepts required by a service-led and information-powered economy.

The ISD program runs a Clinic where students gain hands on experience in information service design and consultant practice. The program has research initiatives in mobile and location-based services, and systems that facilitate collaboration and knowledge sharing among professional communities. The program works to advance service research by hosting a lecture series, organizing an annual symposium, and by hosting visiting scholars. The ISD program also encourages collaboration with other academic, non-profit, and commercial partners.

The ISD Clinic is a University-based information technology consultancy and instructional program providing project-based services to on-campus and off-campus organizations. It has the mission to give students real-world experience in the design, implementation, deployment, and evaluation of information systems. The Clinic is hosted at the School of Information but open to students from engineering, computer science, business and other disciplines. The Clinic's primary client base are organizations on the UC Berkeley campus, but the Clinic also works with industry partners, and nonprofit public interest organizations.

Work with the Clinic is typically sustained through partnership relationships, gifts, or through collaborative project development and grant writing to acquire third-party funding. More information on http://isd.ischool.berkeley.edu/about/clinic.

2.2.1.3 North Carolina State University

A quotation from http://www.ssme.ncsu.edu/: Services science applies insights from scientific, management, and engineering perspectives to analyze how to align people and technology effectively to generate value for both services providers and clients.

A new field called Services Science is beginning to emerge in academia. Services dominate our economy (three-fourths of all jobs), and a rising share of service jobs are highly skilled and technology-intensive, including such activities such as outsourcing, consulting, and process re-engineering. The purpose of the new discipline is to bring analytical rigor to key issues such as services efficiency and services innovation. As a new academic discipline, services science will simultaneously create new knowledge as well as train the first cohorts of services experts with a strong understanding of both business and technology. Ideally they would have the technical expertise of a computer science or engineering major and the business savvy of an MBA.

Why a new discipline is needed:

- Although services dominate the world economy, academic programs and research activity in engineering and business schools do not meet the needs of this sector.

- Services pose unique business and technical challenges. Services are co-produced by the customer and the provider on a customized basis. The provider must understand the customer's business and the customer must understand the provider's capabilities for the exchange to be successful. New tools need to be developed to measure and model efficiency and innovation in this unique context.
- The marketplace is changing. Businesses need to be „open" - able to interact with other businesses, to be adaptive - to respond to market pressures, and to be reconfigurable - to accommodate geopolitical changes and strategic realignments. Even manufacturing and IT companies are moving towards providing services of immediate and explicit value to their customers, and away from selling components that would likely be commoditized . Services experts with a blend of managerial and technical skills are needed to meet these challenges.

※ [STA]$_{24}$:TODO podle nasledujicich poznamek
Univ Cambridge
Arizona State Univ
Stanford
Univ of Cambridge
Masaryk Univ Brno - other approach: harmonizing of customer and provider focussing

2.2.2 Summary

TODO based on
goals from Project DELLIISS and principles of their achievement
book: SSME - education for the 21st century
principles of curriculum design, the focused orientation, comments
focus on business and management
focus on a customer not on the process of service delivery

2.2.3 Knowledge Intensive Services

TODO

Sweden school, prof Margharia Maula
presentations from Web

2.2.4 Why We Need Intelligent Service Systems

Service system is a very complex phenomenon. Prediction of a behavior of particular players in it and prediction of the behavior of the whole service system cannot be done for a long sequence of steps. It is possible to predict one or a very few steps of some agents in a service system and than we need ceaselessly to actualize this predictions. Thus massive adaptation is crucial. A goal oriented adaptiveness is about well done prediction and about well organized memory. In other words it is on intelligence.

Intelligent service system is autonomous, co-operative, composeable, based on massive self-referential and recursive approach, and has ability to evaluate its behavior against taken goals.

TODO:

... instead of tools we need partners to help us ...

Jeff Hawkins: what are the dimensions in which intelligent machines will overcome our human abilities. Four attributes: speed, capacity, replicability, and sensory systems.

Ultimate needs from the security domain

A problem of tuning of complex systems namely service systems

A question of verification, validation, and tuning ...

A classical approach to system or tool or device testing is not adequate! „School examples“ do not work and real life test could be mortally dangerous.

Chapter 3

Service System Definition

⋇ [STA]$_{25}$:ALMOST DONE. PREPARED TO LANGUAGE CORRECTIONS.

Service system definition based on fundamental papers of Jim Spohrer is little bit amended by time dimension here. A shift from focus on ownership relation to focus on co-operation relation is emphasized and its consequences are explained by Give-Take Matrix. If you feel that I am against the ownership, be quiet. This is not so. I only try to give the causes the right balance. I will explain my position deeply in chapter 9 in the end of this book.

Service systems are typically interactive systems. A service which is not only one-shot action cannot follow the classical „paradigm of pipe“, i.e. the sequence (requirements collection)–(analysis)–(solution design)– (solution implementation)–(verification and validation)– (hand over)–(operation). Such a service have to run according to „dialogue paradigm“: (problem formulation)–(consultation, specification)– (operation)–(consultation, specification)–(operation)–...– (close-up). The (consultation, specification)-step is about an answer to a sequence of questions: What - How - Who - Where - When - Why. Each next run through the cycle of the service „dialogue life-cycle“ has to be an improvement of the previous run. The service performing capability must increase cycle by cycle as in other case a customer will be disappointed and consequently lost. Thus a serious service providing needs a good knowledge management.

The requirements of knowledge management from the point of view of a service system needs will be discussed in this chapter as an assignment for research which will be accomplished in the next chapter.

3.1 Understanding of Service

Let us start with several questions:

What is a service?

How it differs from product?

How it differs from other intangible actions?

What are the boundaries between service and non-service?

Aren't such questions only an expression of a wrong point of view?

Thousand years we grow-up our mastership in specializing. In parallel we create and build and maintain boundaries between topics on behalf of ⁎ [STA]$_{26}$:ve jmenu ceho effectiveness and efficiency in our specializations. Boundaries help us to focus our attention. Without this the current advancement in scientific, business, and cultural areas will be not possible. But, maybe we are on the top of this approach capabilities. The continuing boundaries maintenance and development could be contra-productive and may leads us into degradation of a lot of things and topics we were, and we are, successful till nowadays.

How is it possible? Are there such changes in our environment and in our working and business culture comparing these both to the past situation? My point of view is: yes, there are quite big changes.

Looking into very past history, to the times when humankind didn't know money but they worked together in groups to survive we can see that there were nothing other than service in their lives. The most important categories in their thinking were the fight and the co-operation. Both of them were important for survival. The co-operation is a mutual exchange of services between individuals, individuals and groups, and between groups. The faith and its special kind–the hunt is about mutual service exchange again. Comrades-in-arms serves each other through all battle and the same do hunters during a hunt.

Well, the only interesting and important issue an individual could solve for the other or for a group was a service in those ancient times. The result of a service was something usable. And what couldn't be used was for nothing.

But through the time the situation was step-by-step changed. After some thousands of years people recognized that somebody was better in hunting, somebody in fighting, and somebody in manufacturing of knifes or spears. The last one started to sell knifes and spears to hunters and fighters. Vice-versa the hunters sold meat from caught animal to manufacturers. Everybody knew what to do with a knife or a spear or a meat. It was again a service what was usable. But the service was not perceived to be provided directly by those individuals. People started to think about such situations that the service is provided by the product–the knife or the spear or the meat. As the use of things was obvious lot of consultations was not necessary. On the other hand, to have a knife or a spear or a meat, this was very important. The ownership relationship started to grow in its importance.

We can say that at the same time started the development of business with tangible products. You can object there were medicine men and their products were intangible. Yes, it is true. But the business with intangible products was in a way plastered to the business with tangible products. The leading edge was the business with tangible products. This one directed the economy evolution, the law evolution, and the culture evolution.

A criterion of wealthiness started to be how much an individual or a group or a nation owns, not how big and/or how many services he/she/it can provide.

Honestly, I have no objections against this. Without the attention concentration by boundaries between topics, without ownership relationship, money and product driven economy, we would be still running in forests with a piece of leather around waist and a spear in hand. However, the understanding that almost all such boundaries are artificial, i.e. we ourselves somewhere in the past made those boundaries, can help us to comprehend the current very complex situation.

At least more than three thousands years the tangible product driven economy grows-up and anchor itself as a dominant and unavoidable phenomenon. The useful point of everything was service so as in the ancient times, but the service was mostly perceived to be provided by particular tangible products. The products, namely the tangible products, were in focus. Service was think of as something plastered to products. The last sixty years we can feel a breeze of change. The last ten years it is a wind if not a beginning storm. The problems were arisen when intangible products spread on the market.

A little and simply overcame problem was selling a piece of SW and still having it for a new sell. The instrument of licence looks like solving this problem. We sell the rights as the product not the piece of SW. But again, we sell particular rights and we still have those rights for a new sell. But we are accustomed with this situation. The institution of a rental of a land or a realty is known maybe thousands years. A little difference is that in case of piece of SW also the primary product is intangible.

More serious problem is complexity of utilization. There weren't necessary consultations and knowledge sharing when a hunter needed a knife or manufacturer wanted a meat. Moreover, there were not, and are not, necessary consultations and deep knowledge sharing when buying a modern car. It is a pure negotiation. Requirements collected by a buyer serves not in the process of this car manufacturing but for the selection of the desired car. Know-how concerned with manufacturing a car is separated from know-how concerned with driving the car in our particular trip.

But this start to be not true in more and more areas where knowledge occurs to be the main asset. An information system implementation to an organization is an example. The customer has to share its knowledge of particular business domain and its knowledge of particular way it wants to create its success with the information system provider. If not, the information system will not produce expected benefits.

On the other hand, information system provider cumulates knowledge of particular domain and knowledge of ways to success performing its activities. This knowledge it shares with its new customers. If not, it looses its competitive advantage. Exactly, it uses knowledge which it didn't bought. In some rules of law (maybe in lot of them) this situation is classified as non-chartered or non-licensed enrichment. ⋇ [STA]$_{27}$:zjistit presny terminus technicus pravnicky This means the product based economy, law and culture is not enough prepared for service providing. According to my opinion that's one of the reasons why IBM advocates SSME as a new academic discipline.

Both of those knowledge sharing events cannot be seen as services provided by products. These are services provided by a provider to a customer. The first situation was: the customer of IS is the provider of service consisting in knowledge transition to the IS provider. The second situation was: the provider of IS is the provider of service consisting in cumulated knowledge transition to a new IS customer.

This was an example, only. The same situation is, or is arising, in each event in which knowledge intensive processes are involved. Recently, the leading edge of business is this one going through events with knowledge intensive processes.

And this is the change we speak on in the beginning of this section. We, as humankind, have made a great cycle from service focusing, through step-by-step strengthen of tangible product focusing, through ownership relationship domination, through step-by-step strengthen of knowledge intensive processes and knowledge sharing, to service focusing again, but on higher level of knowledge intensiveness.

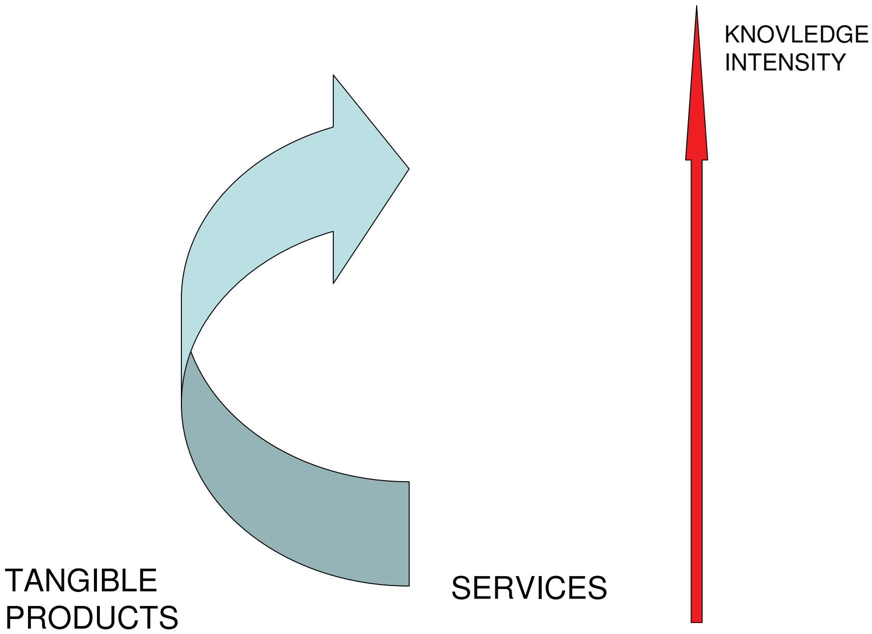

Figure 3.1: Human History–The Progress

Often we can hear, namely from banks or other financial institutions but from SW houses, too, they speak on actions they sell on the market and which are typical services, as they was products. Typical examples are: we offer a product 'term deposit'; we can grant a product 'credit'; we amended our product portfolio by a special 'consultancy product'. This are examples of metonymy according George Lakoff (see [?] ⋇ [STA]$_{28}$:Lakoff: Women, fire, and dangerous things). This is often used metonymy, but not exact identification of the focused topic or object. My opinion is, that this way wording is ambiguous and could be avoided.

Let us return to the introductory questions. The answers could be summarized in following items:

- A service is an action bringing something usable. The mode of „use“ is unavoidable.
- A bearer of service is either an individual, or a group of individuals possibly in a way organized, or a tangible product the use of which provide the service.
- In the modern age of information intangible products could be bearers of service.
- During a life-cycle of an organization or organized group or an individual there are two important things, only: (1) services, as something usable and useful for the receiving subject, and (2) bearers of those services–which could be individuals, in a way organized groups, and/or products.
- The question–„What are the boundaries between service and non-service?“– is not a well done question: service and non-service is a point-of-view not essential categories of things in our world.

The last point is an important one. I will guide you through a lot of similar examples in next pages. It will be a part of my boundaries breaking approach.

Let’s end with a clear and enough general definition of the manner in which we will understand a service here in the following text.

Definition 1 (Service) *By a term* service *we will mean any action bringing some usefulness to a receiver of this action. The ever-present purpose and/or goal of this action have to be* ***the use*** *of the action’s results or outcomes.*

Service posses the following properties:

1. *Each service has its provider, i.e. somebody or something which perform the action and by this provide the service.*
2. *Each service has its client, i.e. somebody or something which receives results of the action.*
3. *The words ’somebody or something’ mean it could be*
   - *individuals,*
   - *group of individuals organized in a way,*

    - *technology assembled and organized into value adding application,*
    - *a combination of previous items.*

4. *A service can be one-shot or repeatable.*

5. *Each service is connected with shared information. Information is shared between a provider of the service and a client of the service. Information concern of what/how/who/where/when/why operates to a benefit of a client.*

6. *Each service is connected with shared knowledge. Knowledge is shared between a provider of the service and a client of the service. Knowledge concern of principles, methods, categories, rules, connections, and operations enabling to create value to a benefit of a client.*

The more information and knowledge is necessary to perform a service the more the service is information and knowledge intensive. Following considerations in this book will cover all kinds of services. But a benefit from systematical approach to modeling and execution of services will be remarkable in case of knowledge and information intensive services.

3.2 Understanding of Service System

Knowing what service is and understanding that services are omnipresent in our private, social and business lives we need to formulate a systematic and general enough approach to

- services creating,
- services performing,
- services maintaining,
- evaluating benefits from services,
- evaluating contributions of partners co-operating in service provision,
- etc.

A helpful concept will be 'service system' for this purpose. Service system will be defined here in compliance with Jim Spohrer definition in [7] [8, 13, 14], but a little bit amended. Let's follow a step-by-step understanding of this concept.

Figure 3.2 give us a first insight. A service system must cover not only the service itself but all necessary surroundings. Thus in the first approximation a service system is a triple *(C, P, T)*, where C is a **client** of the service (the service yet not expressed in this approximation), P is a **provider** of the service, and T is a **target** of the service.

The provider could be an individual, or a group of individuals in a way organized, i.e. an organization, or a technology assembled and organized into value adding application, and/or a combination of those possibilities (cf. definition 1). The provider must be an agent in a sense. It has to be able to perform actions. It has to be able to perform those actions autonomously and it has to be able to perform those actions in co-operation with other agents.

The client is usually seen as an individual or an organization, as fig. 3.2 demonstrates. But in fact, now, it could be the same entity as a provider can be. In the past, before the computer era it was not possible. Now, an entity consisting of an organized group of people and a piece of information technology, both working together as a unit in a business environment and for a purpose of this business environment, can be seen as a client of a service.

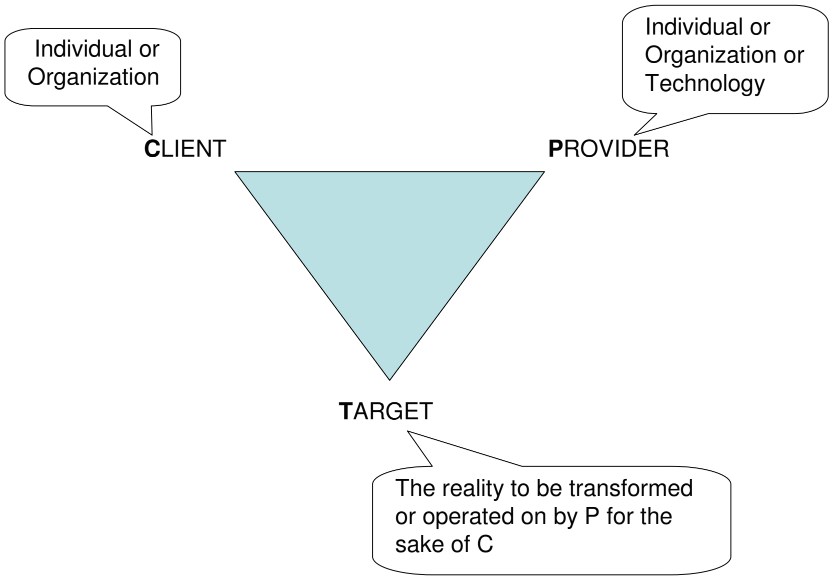

Figure 3.2: Service System Definition - phase 1

The target is a part of reality to be transformed or operated on for the sake of a client. Such a part of reality could be anything. It could be a land, an electric infrastructure, a building, a computer network, but a social network, too, an individual, an organization, etc.

For example a service consisting in education has individuals as target. If these individuals are very young–at basic school, e.g.–they are not clients of the service of education. They are just pure target. But, if these individuals are university students, they are target of the service of education and at the same time they are clients of this service.

Another example could be my one of former client for whom and with whom I had performed strategic change project. The client's organization was a target of this service I was a provider of, and again this organization was a client of this service.

It is not difficult to imagine that the role of a target could be played by any entity which can play a role of a provider and/or of a client, in a special case. On the other hand there are targets that could play a role of a client, or of a provider, with difficulty only, if not impossible. An example of such a target can be a piece of land with meadow. A good target, but bad client. Maybe it could be seen as a provider–it provides a service for herd of cows, e.g., but this is a service with very low intensity of knowledge and a very low level of autonomy.

—

The first insight, cf. fig. 3.2 gives us a picture of a playing field. In the second insight we can focus on important relationships. Important relationships are those one, which connect the above established triple (Client, Provider, Target). Figure 3.3 shows the relationships.

The game starts by initial negotiations between a client and a provider. A presupposition is that the client wants to do something with a target or on a target he/she/it is owner of or responsible of, and that the provider has a competency to perform desired transformation on the target.

The client and the provider have to negotiate what/how/who/where/when/why will be done during the service performance. They have to negotiate this initially in the initial phase of the service execution. But this negotiation has to be maintained and revised continuously through the service provision lifecycle. Moreover, they must negotiate what portion of information will be shared between them and what are the conditions of this information sharing. These conditions have to be mirrored in a way in a price negotiations. The price of the service is a subject of initial negotiations, too [1]. The result of such negotiations can be materialized in a contract between the client and the provider upon which the service will be provided.

Till now it looks as known process where no hidden problems can appear. Unfortunately it is not so easy. There is knowledge which has to be shared between a provider of the service and a client of the service. Knowledge concerns of principles, methods, categories, rules, connections, and operations enabling to create value to a benefit of the client as was said in the previous section. Without the shared knowledge no service can be executed successfully. But knowledge of principles, methods, rules, ..., etc., form a

[1]But paying the price could be seen as a service, again! The deep discussion to this issue will be in section 3.3.

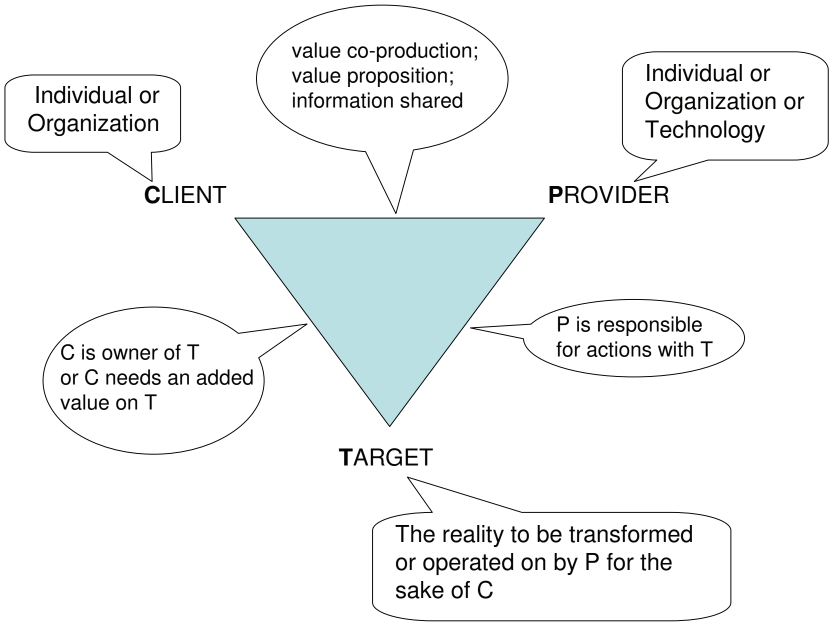

Figure 3.3: Service System Definition - phase 2

very important part of know-how of both, the client and the provider. Evidently there are two sets of principles, methods, categories, rules, structures, and operations in this game: the first one is a property of the client, and the second one is a property of the provider. This knowledge is very often a subject of business secrets of an organization. The problem arise how to share something which is a subject of business secret on both sides, the side of a client and the side of a provider. We can say: This is a problem of lawyers and of rule of law. Yes it is, but it increases difficulties in service provision. I, personally, count this problem to those one that led to SSME formulation, namely it approves for me the concept of 'service science'.

But the issue is not only on the side of law. The problem is, moreover, a technical one–if I can say that a way we externalize our knowledge is a technique. Value co-creation performed by both, the client and the provider, means mutual adaptation of both to the conceptual system of its counterpart. The client's knowledge is expressed or encoded in a client's conceptual system, so as the provider's knowledge is encoded or expressed in the provider's conceptual system. The aim of service providing and consuming is to utilize something out of one's professional abilities. Thus profession is what separates the customer's conceptual world from the provider's conceptual world, often. But the knowledge has to be shared between them. How can we do it? To translate the knowledge of the customer to the conceptual system of the provider? No, the customer will be not able to utilize the service in provider's conceptual system. And to do it vice versa? Again not a very comfortable solution. The provider could be overflowed by customer's conceptual systems. But the situation is better than in the first case. If technology will be able to support embedding of a customer conceptual system into the one of the provider than it could be helpful.

Nevertheless where the leading edge is going in a particular case, there will appear a mutual adaptation of conceptual systems of both, the provider and the customer, in some measure. A really helpful topic in this situation could be a widely recognized and accepted pattern to which the mutual adaptation could head towards.

Value co-creation means a mutual co-operation between the provider and the customer. We have mention the necessity of knowledge sharing and particular conceptual systems adaptation to each other to be able to co-create value. As well, we have mentioned information sharing from which the provider recognizes what to do for the sake of the customer and which helps the customer to understand its problem deeply and to formulate its requirements in more proper way. All those knowledge and information sharing is materialized during consultations and negotiations which are frequent at the beginning of the process, but which continue through all the process. However, through all the process of service provision there must be at any time point valid **statement of work** declaring what have to be done, how it will be done, who will do what, where it will be done, when what will be done, and why it will be done. This statement is called a *value proposition.*

The term 'statement of work' comes from project management. It is not accidental. I will argue in next pages that project and program management is a topic very closed to service provision.

Value proposition is the most important connection between the customer and the provider. This must be understood by both parties in the same way, this must be valid all the time through the service provision life-cycle. As the reality continuously changes the value proposition has to be repeatedly aligned with the changing reality. Everything which is said in project management body of knowledge, e.g. in the IPMA[2] Competences Baseline (one of the world-wide standards of project, program and portfolio management) ⁂ [STA]29:cite ICB standard about interested parties and project management success competences can be applied to value proposition declaring the scope, the content, the process, the conditions, the agents and the target of the service provision.

Let's turn our attention to the Client-Target relationship. Typically it is a kind of ownership relation. A target is something the client is responsible for and/or the client needs for its benefit. This can be based on client's ownership of the target or client's ownership of rights to decision making on the target. To find a position for a service there must be a gap in competences of the client to handle with the target as in other case the client helps itself. This gap could be formed by a missing technical competence or by an economy of scale problem with manipulating with the target by the client itself, e.g..

Thus the Client-Target relationship expresses that in general a target belongs to a scope of a client. When a provider starts to manipulate with the target for the sake of client, he/she/it is „on visit“ in this scope in a way. And this is what brings difficulties and complexity to the service provision, again.

The last relationship on fig. 3.3 is Provider-Target. This relationship is much of all about provider's capabilities to perform operations on or transformations of a target. The provider is responsible for the operations on or the transformations of the target for the sake of the client. To do this he/she/it must have necessary competences. These could be competences to problems, competences to processes, competences to being on visit in client's scope. Those competences prove the usefulness of the provider's actions. And usefulness is a top measure of service success.

—

The third insight of service system is on fig. 3.4.

If anything is to be done first of all we must think on it and often speak on it with others or with ourselves to negotiate what/how/who/where/when/why[3] to do. In other words we must plan for the future actions. From the philo-

[2]International Project Management Association

[3]these six questions come from the well known Zachman Framework [19]

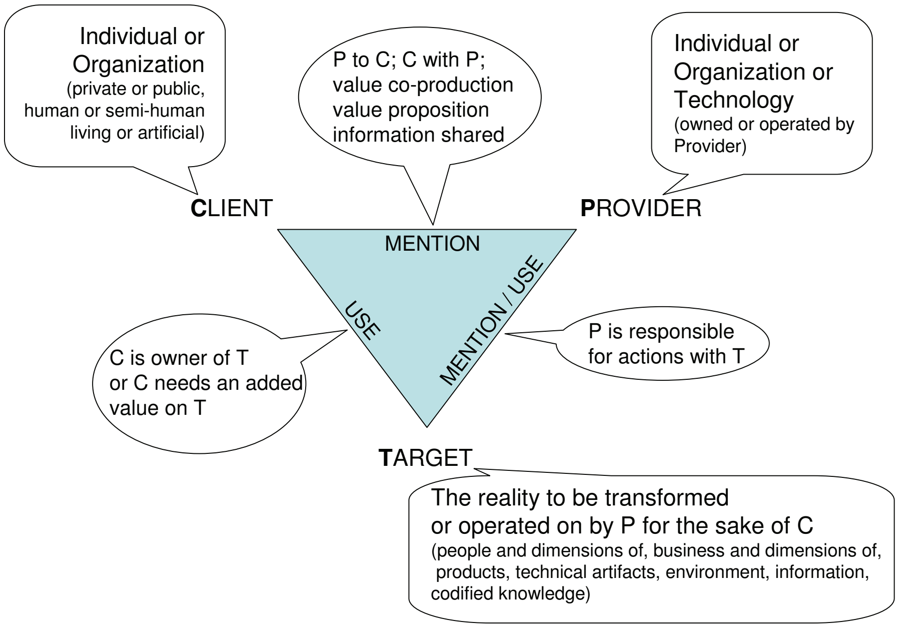

Figure 3.4: Service System Definition - phase 3

sophical point of view it means mentioning (MENTION) of what could be or probably will be performed and by what means. The second step is acting according those plans. This means using (USE) of our capabilities, appropriate tools, or components of service system, simply given means, to act to bring a value.

All our life, as we are members of humankind, is a continuing duality of MENTION and USE. When acting concerns a physical world and tangible products, there is no problem. Everybody can distinguish what is MENTION and what is USE. But acting in cyberspace (a space of our ideas, information, and knowledge) brings difficulties. An attention have to be paid what is planning–i.e. MENTION, and what is acting according those plans–i.e. USE. Confusing this leads to great misunderstandings. An example could be paradoxes of ancient Greek philosophers (paradox of liar, e.g.). Another example is the well known Russell's paradox of the set of all sets[4].

The Client-Provider edge (see fig. 3.4) has a label MENTION. It denotes that there are correct relationships between the client and the provider. If there would be an USE label, it could marked a kind of harassment between those two. The client and the provider, both together plan, revise plans, evaluate situation against the plans. Those plans are about how to provide a service on the target for the sake of the client. The value proposition is an expression of this MENTION activity.

I had mentioned in previous sections that service is nonseparable connected with usefulness. This feature is manifested by the label USE on Client-Target edge. And this feature is the reason for a service provision and hence for a service system existence. The service is always a means of an utilization of a target for the sake of a client.

The Provider-Target edge is little bit more complicated. To perform an operation on or a transformation of the target to the benefit of the client needs at least two steps: first to plan what to do, and second to do it. Usually it consists of more than two steps: there is repetitive sequence of those two steps. This repetitive plan-do cycle is expressed by the label MENTION/USE on the Provider-Target edge in fig. 3.4.

The importance of this MENTION-USE point of view will be clear when we start to analyze the knowledge intensive services deeply. Each such service is about intensive knowledge and information sharing. To share information we need to mention it. To share knowledge we need to mention it and to use it, too. Not used knowledge is not well accepted. This means a co-operation, and namely, a value co-creation requires using of both modes–the mention mode and the use mode by both actors–the client and the provider. The picture in fig. 3.4 give us an idea how it works.

—

[4]in Appendix 11.1 you can find comments to a classical Russell's solution of the paradox and a solution of Russell's paradox without type theory and self-reference forbidding using strict distinguishing between MENTION and USE

The fourth insight to service systems nature shows us the deep of the possible service system complexity. Small C-P-T triangles in vertices of the basic service system triangle on fig. 3.5 represent service systems again. Each of the three–Client, Provider, and Target–can be a service system again. Let us awake to the fact that this means that in any vertex of service system triangle there can be recursively embedded a finite sequence of service systems.

The lengths of these sequences in different vertices are not the same, of course.

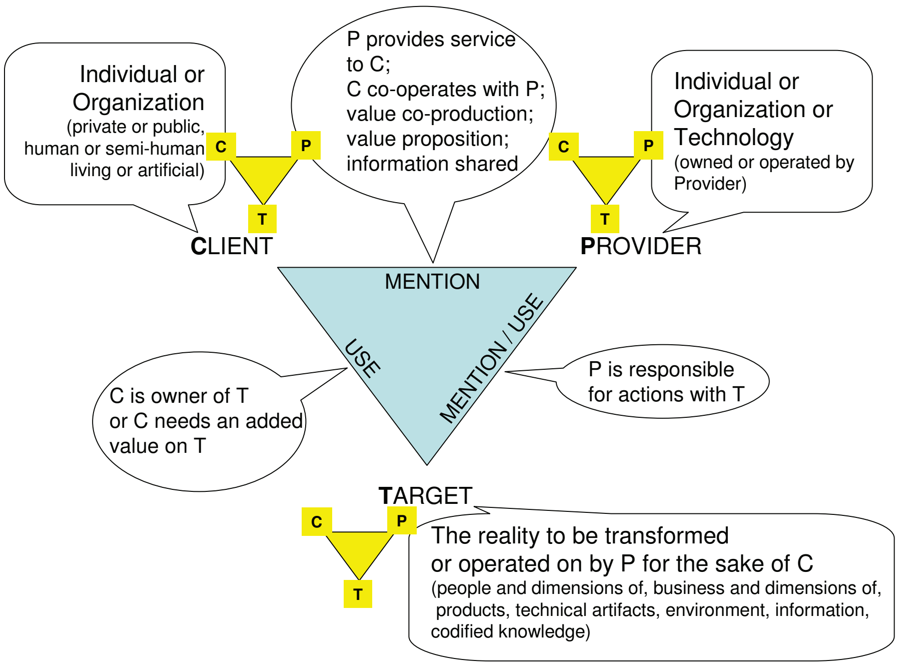

Figure 3.5: Service System Definition - phase 4

The length zero means there is no embedded service system in the focused vertex. It is usual in case of the Target vertex. Often, there is no reason to consider the target to be a service system again. But it can be. For example when the target is an organization, an enterpriser or a plant, it is natural to consider it to be a service system again. On the other hand, e.g. a pure infrastructure as is an electric power network (only the cables and technical tools, nothing more) is naturally seen as a target being not a service system itself.

Thinking on Provider and/or Client vertex the situation is opposite. Providers or clients are agents in the game, i.e., they are autonomous in some sense and they are co-operative. If not, they would be not competent to play their roles in service system game. Moreover, providers and clients are compound very often. Typically an organization is such. It is not difficult to distinguish a client part or component, a provider component, and a target component in their inner structure with respect to an appropriately chosen context. So, it could be natural to consider them to be service systems.

But again, there can be a situation in which to consider a provider or a client to be a service system brings nothing. As I said in previous section a pragmatical point of view helps us to decide whether to speak on service or service system or not in a particular situation.

Very often the provider or the client is an organization or a part of it, in which providers and clients and targets can be recognized in a form of embedded organizational units, in which, again, providers and clients and targets can be recognized in a form of teams, team members, information technology applications, other technology builds, etc. The important message is: thinking on service systems it is better to avoid a boundary between people and technology. This boundary complicates situation only. The boundary means proper distinguishing between people in service systems and technology in this systems. The nature, the purpose, the functioning, the processes of, the behavior, the goals, the objectives, and the deliverables of a service system can be understood without this boundary. We will see this boundary avoiding advantage in the following chapters.

—

The figure 3.6 contains the whole service system definition. Recursive nature of the definition was mentioned within the explanation of fig. 3.5, the time dimension of the whole picture of the service system is the last amendment of the definition.

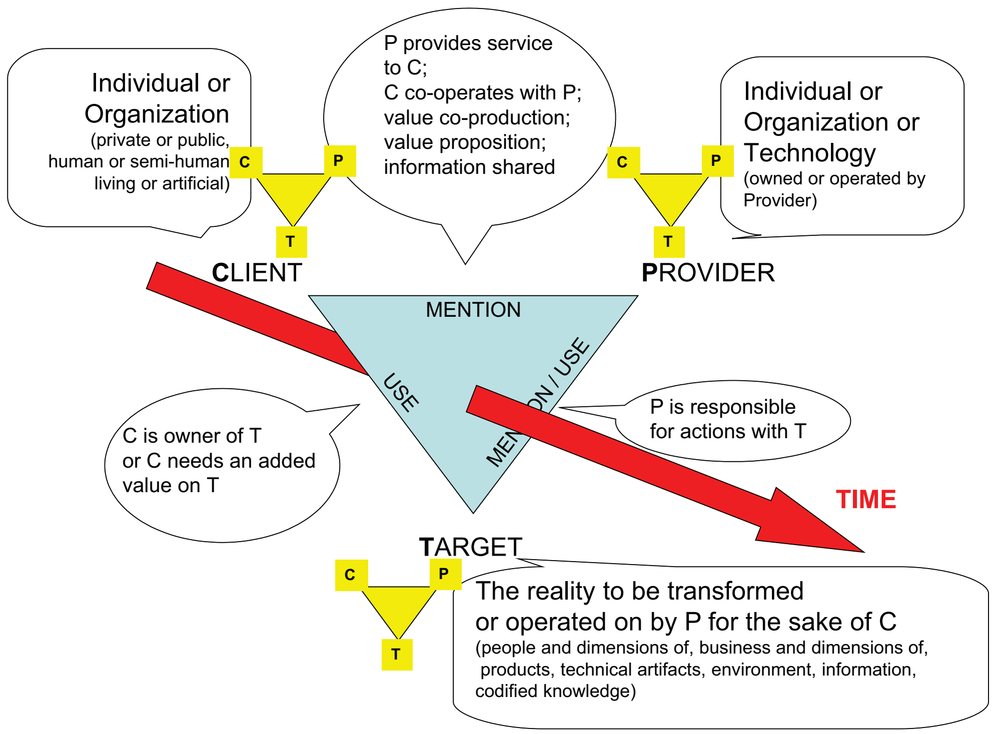

Figure 3.6: Service System Definition

To sell anything, except trivial goods of daily consumption, means lot of preliminary work. This is the same for a product as well as for the service. But reaching the point „done“, i.e. the sale is executed, differences occur. When selling a product it is almost a successful finish of our endeavor. When selling a service, it is practically the beginning of a second phase of our endeavor. At this point start processes of service execution. The preliminary work for the sale before this point was much more about value proposition formulating and desired service modeling. This was the first phase of our endeavor.

This, what makes difference between selling a product and selling a service is a situation development through the time. When selling a product there is no situation development after the decision that the sale will be done. But when selling a service, there is. Remember that at least two of three vertices of service triangle are autonomous agents; these are the client and the provider. (And the third vertex can be an agent, too.) Each of them has its own history and the content of memory of the client is usually very different from the content of memory of the provider and, if target is an agent, from the content of its memory, too. From the point „the sale is done" these two, maybe three, agents start their common period of history. They start to influence each other, they start to share their knowledge, till this point a private one. And when the service provision ends, this common history remains. This process is not reversible. And both two, maybe three, parties leave the common period of their history changed–influenced by the other parties. Sometimes it is called a customer experience dimension in service economy ⁎ [STA]$_{30}$:citace od SME kdosi tomu rika customer experience dimension in service economy - najit a zaclenit — TODO. Hence, it is not so easy to sell a service to your competitor, while to sell a product to a competitor often doesn't make problems.

Looking on management and business books, we can see how many books was written on competitive strategies and on the other hand how few books we can find about co-operative strategies. We are much more trained in competition than in co-operation. But service provision needs co-operation. My opinion is that this is again an argument for service science discipline.

Watching a service system from the time flow point of view there is in each vertex of the service triangle its own stream of events mapping the vertex history. Event stream in each vertex is naturally different from those in the remaining vertices. So the history, the experience, the expectations, etc., of those two or three agents, client, provider, and maybe target, are different. Those streams begin to influence each other at the beginning of service provision. At the end of service provision the intensity of mutual influence of those streams decrease, but often we cannot say it completely ends.

Simply saying, a service provision can be seen as a temporal marriage of a client and a provider. And maybe instead of looking for advice in management and business books it could be better to seek advice in marriage advisory centers. This looks like a joke, but the marriage of a man and a woman is an archetype of service system. In this service system both partners play a role of client and of provider, and sometimes of target simultaneously. And this could be seen as the last important feature of service systems, generally. During a time of service lifecycle the roles of agents, i.e. the roles client, provider and target, can change.

At each time point and a given point of view there is one spread of those three roles on playing agents. From another point of view the spread of roles could be different. As well in other time point it could be different. Thus, the roles of agents in service system depends on context in which we evaluate the service system state.

Moreover, there is no reason to limit our observation to only three agents and/or pieces of reality. Usually we observe a set of agents and/or pieces of reality and in particular contexts we spread the roles of client, provider and target on those actors. The service system triangles Client-Provider-Target are in a way interconnected. Our pragmatic point of view drives our attention to focus on a particular service system triangle. And this focus changes in time and changes according to a given point of view.

—

Summarizing the whole explanation through the previous four figures to the last fig. 3.6, we can formulate a following definition.

Definition 2 (Service System) Service system *is a composite of agents, technology, environment, and/or organization units of agents and/or technology, functioning in space-time and cyberspace for a given period of time. There is always lot of contexts from which the service system could be evaluated, explicated and comprehended. There exists at least one context from which the roles of Client, of Provider, and of Target could be recognized on agents or environment.*

The following statements must be valid for such composite entity to be a service system:

1. *The Client-Provider connection covers*
    - *information sharing between them,*
    - *knowledge sharing between them,*
    - *negotiations between them,*
    - *value proposition establishing,*

- *repetitive reviewing the previous items.*

2. *The Client-Provider connection is about consultation, negotiation, planning, ideas and plan development, simply its nature is MENTION mode.*

3. *The Client-Target connection expresses a kind of ownership which could be*

   - *the Client owns the Target,*
   - *the Client owns rights to use and/or manipulate the Target,*
   - *the Client owns problem the solution of which could be performed by operating on or transforming of the Target.*

4. *The Client-Target connection expresses that the Client uses in a way the Target.*

5. *The Client-Target connection is about life, about putting into practice. Its nature is USE mode.*

6. *The Provider-Target connection expresses a kind of competence. It could be*

   - *the Provider knows and is able to operate on the Target,*
   - *the Provider knows and is able to transform the Target,*
   - *the Provider understands the Target and is able to plan for operation on or transformation of it.*

7. *The Provider improves in a way the Target for its better utilization by the Client which means a benefit for the Client.*

8. *The Provider-Target connection is about project or programme of projects the goal of which is a change improving Client's using of the Target. Thus its nature is both, the MENTION mode and the USE mode.*

9. *Any agent playing a role in the service system triangle Client-Provider-Target could be a service system again.*

10. *Any agent playing a role in the service system triangle Client-Provider-Target could be*

- *human being,*
- *group of human beings in a way organized,*
- *any of the previous items coupled with a technology and possibly with a piece of environment,*
- *simply a kind of organization,*
- *virtual, i.e. artificial, agent made from SW and/or HW components,*
- *any of previous items in a combination.*

11. *The Target could be an agent or a piece of environment including a technology.*
12. *The time period of the service system „life" is not a trivial one compared to actions performed within a service provision process.*

Often I hear that service systems could be so complex that it is almost impossible to define them. My point of view is: if we want to mention something and to use it, it is necessary to understand what this „something" is, and to be sure whether a focused instance/element belongs to this „something" or not. That is why I prefer to put a complicated definition as above instead of to say 'situation is too complicated to formulate a definition'.

Summarizing our thoughts and discussions we can conclude:

- Service systems are composed of agents and sometimes of pieces of environment. When environment is involved, than it plays the role of Target. Agents can play all three basic roles–the role of Client or Provider or Target.
- There exist lot of contexts from which we can watch and evaluate a service system life cycle. A context is valuable if we can recognize a Client, a Provider, and a Target roles on playing agents and possibly environment.
- Studying service systems is, so as our perception, an enactive process [5]:
  - first, we pragmatically choose a context,

[5] enactive perception is well described, e.g., in Alva Noe book 'Enactive Perception'

  - second, we pragmatically spread the roles Client, Provider, Target,
  - third, we watch this way established service system behavior and its structural preconditions and connections.

- Service systems are very useful constructions helping us to understand the reality deeply if the appropriate contexts and roles spread was chosen.

Following figures contain example of a part of reality that could be possibly seen as service system. The first figure 3.7 pictures team members (ellipses) of a program and their relations to projects (rectangles). The team members could play roles of Clients and Providers, while projects could play roles of Targets. The second figure 3.8 pictures a spread of those roles on the agents.

A project could be seen as an agent, too, as it is a composite of people, environment, technology, connected with shared knowledge and shared information. Not surprisingly we can see a project as an embedded service system, again.

Surprisingly lot of things in our world and in our life could be seen as service systems.

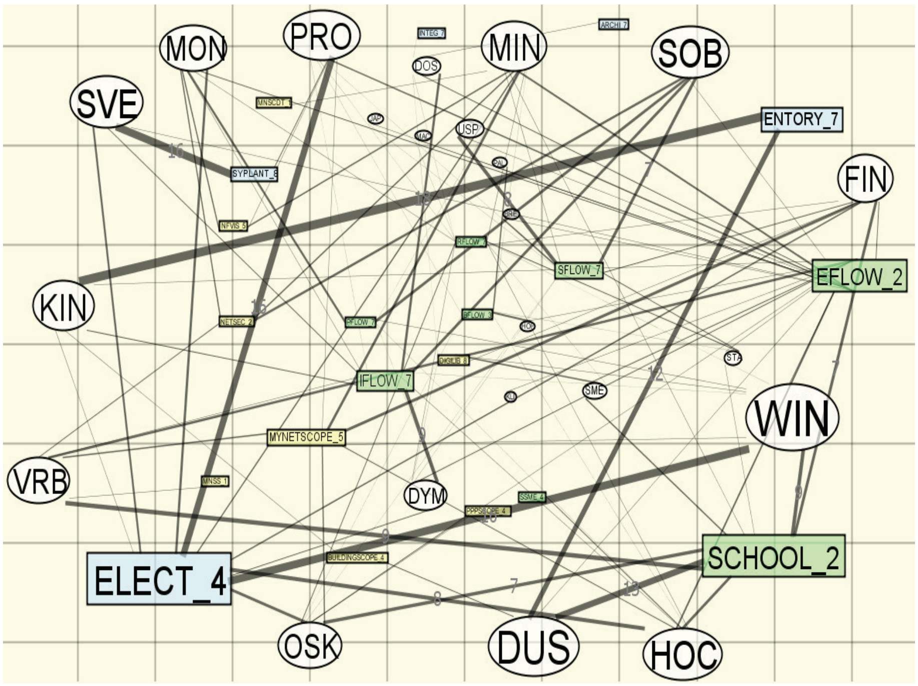

Figure 3.7: Service System: Team members on Projects.

3.3 Boundaries, Contexts, Co-operation

Service system is a very fruitful concept. Lot of business situations could be better understood using this concept. Typical situation in which service system concept brings better understanding is project management and programme management, where programme is a portfolio of projects connected by one set of shared goals. Another such situation is a complex information support of a company or of a group of co-operating companies.

More details to these topics we leave to next chapters while focussing here to one challenging question: what is better to describe as one service system and what is better to describe as two or more service systems in some way connected or co-operated. In other words: are there criteria according to

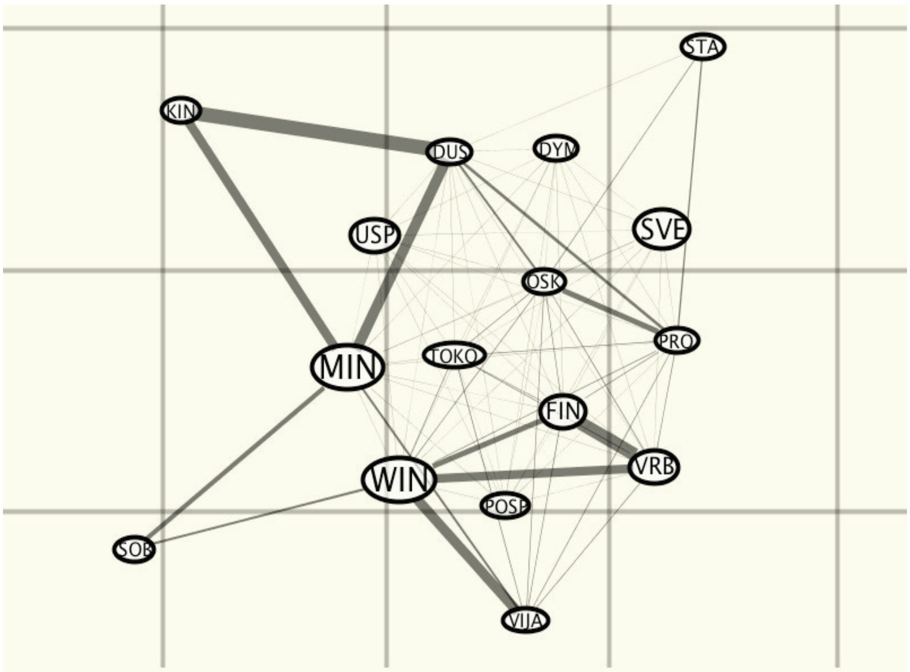

Figure 3.8: Service System: Co-operations among team members; Targets hidden.

which we can decide what way is better or is it a pragmatic decision only?

Let's start with an example: There are two firms, a software house Heavysoft and a telecommunication company Telecoco. Telecoco wants to outsource its information system. They made negotiations, share necessary information and share necessary knowledge, formulate a value proposition and establish a service system in which Telecoco will be a client and Heavysoft will be a provider. The target of this service system will be run of the information system of Telecoco together with all directives and regulations connected with Telecoco operations. Heavysoft uses its competences to act for the sake of Telecoco on this target. It is an usual situation and it is a common way to understand the service system operating.

Of course Telecoco pay a fee for this service. Remembering what we said in sections 3.1 and 3.2 we see there was nothing about paying. Where to put this process? Did we forget to speak on this very important topic? No, we didn't. All this is included in the service system definition. Let us turn our attention to contexts in which we try to explicate the system. In case of Telecoco and Heavysoft we described the outsourcing system from the point of view of the client. This is an usual approach to take this context when explaining such a service. This is a context of Telecoco benefits. But there is another important context–the context of Heavysoft benefits.

From this point of view Heavysoft is the client and the target is Heavysoft's bank account. And of course Telecoco is the provider of the fee payment service. Again this service was negotiated, information and knowledge was shared, and the result of negotiations is a part of the value proposition mentioned in the beginning of the example. This is the second context from the lot of contexts in definition 2 from which the roles client, provider and target could be recognized.

In both contexts, the context of Telecoco benefit and the context of Heavysoft benefit, the roles spread is stable through all the lifecycle of the service system. This means the triangle Client-Provider-Target doesn't change in time in no one of the two contexts. When trying to catch and to understand a situation it is more comfortable if the roles client, provider and target are spread on service system's agents in stable manner during the system's life time.

The situation in our example could be described in other way. We can „decompose" the whole service system with those two important contexts, the context of Telecoco benefit and the context of Heavysoft benefit, into two service systems. The first one will contain only the main context of Telecoco benefit, and the second one will contain only the context of Heavysoft benefit. Let us leave the understanding of the expression „service system contains a context" on an intuitive level, now. A deep explanation will be done in chapter 6 where a way how to model particular service system's elements will be introduced.

A service system containing only one context such that client, provider and target roles spread on the systems agents is stable through all the life time of the system is much more useful for describing of time-space and cyberspace situations and event sequences. Service system of this type is more comprehensible and more predictable.

The above example is a generic one. Almost all business situations could be seen in this manner. This leads to the following definition.

Definition 3 (Prime Service System) *A service system is called* prime service system *if the following conditions hold:*

1. *there is only one context involved in which the roles of client, provider, and target are spread on agents and/or on a piece of environment,*

2. *all the time the service system is alive the role assignment does not change.*

Prime service systems are usually focused on when speaking on service systems. This is a natural point of view and it is widely used. However, the general definition–def. 2–is important not to miss marginal cases. In the following pages when speaking on service systems we will have in mind the prime service systems, except a situation we explicitly express we speak on general service system or other kind of service system.

Maybe you can feel now little bit confused (we are „skating on thin ice“) as the concept 'context' is on one hand intuitively clear but on the other hand it is vague. This is not a very good reference for using it in a definition. Yes, you are right. It is so. I would like to ask for your patience here. An exact explication of what is context will come in chapter 5 and elaboration of service system definitions you will find in chapter 6. The reason why I do such a loops and why I speak on context before its precise definition is simply didactic and marketing one. I do not want to lose your attention. Thus, you must be motivated to read a non easy abstract chapter 4 on cyberspace and modeling and similarly difficult chapter 5 on Diamonds as thinking patterns enabling consistent comprehension of a very various reality of service systems life cycle.

The two service systems based on Telecoco benefit context and Heavysoft benefit context, respectively, are tight connected. Doing a decomposition to prime service systems we have to describe them together and we have to describe their mutual co-operation. This will be one of the first steps towards a theory of co-operation.

Let's become aware of more than three thousand years of dominance of product economy, i.e. product ownership and product selling. And remember that most of this time this were tangible products which were subject of ownership and of sale. Our mind was developed, and still is developed, under the **tangible product ownership paradigm**. It could be expressed in a very simple and primitive pattern:

(I have)–(you haven't)–(you have)–(I haven't)

According to my opinion this is not surprising that this pattern leads often to the zero-sum game in the GTM (Give- Take Matrix)–see fig. 3.9

Again I want to stress that those arguments are not directed against the ownership relationship. This is one of the most important relationships in our lives. I would like only to advocate another one relationship which according to my opinion is of the same importance as the ownership relationship. And the problem I feel is that this relationship is not widely recognized as the second one of the most important in our lives. Arguments for this claim are covered by a simple ratio: number of books on competitive strategies to the number of books on co-operative strategies.

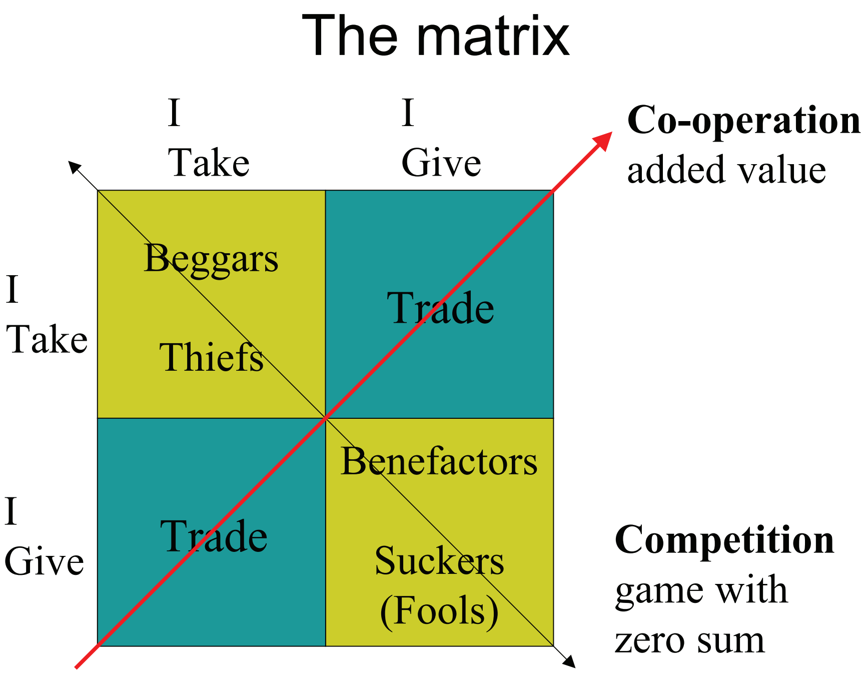

Figure 3.9: Give-Take Matrix: a basic pattern of behavior

This above mentioned relationship between two service systems, namely the one Telecoco benefit context based and the one Heavysoft benefit context based, expresses a **co-operative paradigm** or a **value co-creation paradigm** and can be described by a following simple pattern:

(I am useful for you)–(you are useful for me)–(we both together create a value)

This pattern cannot be applied to zero-sum games. The only possibility is to apply it to the value added game in the GTM on fig. 3.9.

To be sure what to have in mind when speaking on co-operation in this book, we put a definition as follows.

Definition 4 (Service Systems Co-operation) *We will say that a prime service system co-operates with other prime service system if the following conditions hold:*

1. *the Client of the first system is a Provider of the second one,*
2. *the Provider of the first system is a Client of the second one,*
3. *a benefit of each Client of those two service systems depends in a way on the benefit of the other Client.*

Note, that no constraint is done on targets of those systems. Describing the outsourcing of Telecoco IS and operating regulations by Heavysoft in terms of two prime service systems we, in alignment with definition 4, can say that those two service systems are co-operating. The target of the first system–described from the Telecoco benefit context–is Telecoco IS together with all directives and regulations connected with Telecoco operations. The target of the second system–described from the Heavysoft benefit context–is Heavysoft's bank account. The service provided by Heavysoft in the first system is outsourcing of IS operations, the service provided by Telecoco in the second one is payment for this outsourcing. Success of one of those processes cannot be reached without success of the other.

Another typical example of co-operating service systems could be co-operation between a business entity and an academic institution. A similar example could be co-operation between two business entities from different domains. The business-academic example is more illustrative, so I will use this one.

Let's take Faculty of informatics of Masaryk university and IBM Integrated Delivery Center in Brno a business partner of Masaryk university. The faculty teaches students to be usable in their future employments and does research in new technology principles. The business partner develops new services using new technologies and utilize competences of faculty graduates. Let's imagine they want to co-operate. [6] What is the reason for such a co-operation? Benefits for the business partner could be for example:

- graduates prepared to direct enter into partner's business processes,
- possibility to influence the education process on the faculty,

[6] They does. But the example is fictional.

- possibility to select graduates with proper skills and capabilities,
- possibility to extend own research capacity by faculty laboratories,
- possibility to do the first selection of business valued research results.
- possibility to use research results,

Benefits for the faculty could be, e.g.:

- students pass a practical projects in real business environment during a master degree curriculum,
- faculty graduates have a good position on the work market; they are well prepared for practice and lot of them catch a place by business partner,
- faculty can obtain a good evaluation of transfer of research results into practice,
- the market position in the competition for new students is empowered by a good reputation of a rate of employment of faculty graduates,
- faculty can obtain business grants from the partner.

Let us describe the situation in terms of service systems. The first co-operating service system is IBM benefit system: the client is the business partner (IBM), the provider is faculty (FI), and the target is business area of strategic outsourcing. To understand a service the appropriate value proposition must be read. Let the value proposition contains (except others) the following statements:

- A pipeline for talented students and graduates prepared directly to enter our business processes is established.
- Experts from our management and strategic research and development teams are associated members of scientific boards influencing education process and curricula at the faculty.
- Research problems arising from our business practice are investigated and studied at faculty laboratories.

- A list of annotated research projects solved on faculty together with short actual situation messages is at our disposal

The second co-operating service system is FI benefit system: the client of this system is FI, the provider of this system is IBM. The target of this second system is the process of study and the research process, both on FI. Let the value proposition of those second service system contains (again except others) statements:

- There is a contract between FI and IBM on a number of students which can pass 5 month full time project each semester in IBM department the scope of which is principally connected with service systems (their creation or operation)
- A list of diploma thesis topics originated in business practice is established together with guarantors from IBM and is continually amended.
- Grants are regularly obtained from IBM supporting faculty research.
- Faculty is on the list of well co-operated schools published in official web pages of IBM.

Can the second service system work without the first one? Of course it can. But there must be a reason for the provider in this system, i.e. for IBM to do it: the faculty or the university have to perform something for IBM which brings a value comparable in a way with the effort spent on working of the second service system. In other words a kind of service system in which faculty or university plays the role of a provider must exist. Thus, in essence, this is the same situation as the one described above as the first service system.

This what connects service systems into a co-operating ones is mutual usability. The same pattern we can recognize in the first example about Telecoco and Heavysoft.

The usability for Client is expressed in the value proposition. Hence, when trying to evaluate whether a service systems co-operation will be successful, i.e. both parties will be satisfied in their client roles, we need to compare, analyze, and evaluate respective value propositions. To do this we need a deep understanding of what matters a **value co-creation** in each unique case, what level of **cognition** of both worlds, the world of client/provider and of provider/client, is necessary to achieve, what particular **knowledge** needs to be expressed and shared, and finally, what **information** needs to be collected and shared. This is why the next chapter 4 is devoted to those topics.

Note a very important issue: when speaking on service systems co-operation there is no interaction between service systems as business entities. There is interaction between agents playing roles in service systems. If in turn those agents are service systems, then there could be interaction of service systems, but not of those we speak on primarily. A service system is something like relationship envelope on existing business entities. Business entities are objects, while service systems have not a nature of objects. They have a nature of connections. Remember the definition of service system co-operation: the definition do not speak on service systems as interacting entities but it describes a way the business entities–agents playing roles of client or provider or possibly a target–interact mutually. Those connections bring the property of co-operating service systems.

But let's go back to the situation with Telecoco and Heavysoft and the payment for the outsourcing. To formulate value propositions of both co-operating systems first we needed to make clear what is a target of the first service system beneficial for Telecoco and what is a target of the second service system beneficial for Heavysoft. At the beginning of this example we have said the first target is information system of Telecoco together with all directives and regulations connected with Telecoco operations, and the second target is Heavysoft's bank account. Only after this is stated we can try to formulate appropriate value propositions. Thus we can argue: the target is the point to which we relate statements of value proposition in a given prime service system. This is a kind of measure that help us to understand the service system, to evaluate its usefulness, to comprehend wether it is successful in achieving its goals or not.

Again this example is a generic one. Generally, a formulation of value proposition is based on chosen target of the service.

The same situation we have met in the second example of FI and IBM co-operating service systems. Value proposition formulation in both cases was based on chosen targets. Remembering the GTM (see fig. 3.9) and the both value proposition formulations a question arises: how to recognize if the value proposition of the IBM benefit system is in a good balance with the value proposition of the FI benefit system? The same question is legitimate in the first example with Telecoco and Heavysoft. But in this case we naturally suppose there is a balance. To solve this question for IBM and FI we need to compare the value added in the IBM benefit system to the value added in the FI benefit system.

Businessmen would be seeing no problem. This is simply a matter of marketplace. But analyzing service systems we need a more deep insight into the problem. To compare value propositions of two service systems a kind of similar structure of those propositions will be useful. A theoretical insight to this will be introduced in next chapter–chapter 4. More practical view on this problem will be taken in chapter 6 and amendments from an evaluation point of view will be added in chapter 9.

However, one point is clear just now. The situation will be a simpler one if the targets of both co-operating service systems will be the same. In this case both parties (IBM and FI, so as Telecoco and Heavysoft) will understand the target from both sides: from the point of view of a provider, so as from the point of view of a client. Their conceptual systems will be close together in such case and the value propositions will be easily formulated. Moreover, it looks like this two propositions could be substituted by one two-directional value proposition.

Surely, there is not only one way how to identify a common target for both, the IBM benefit system and the FI benefit system. One possible solution could be: let the common target of both service systems–IBM benefit system and FI benefit system–is „usable and in IBM used graduates and research results of FI together with usable and at FI used graduates specifications and research direction specifications from IBM“. Over such a target a two-directional value proposition can be formulated. It could contain, e.g., the following statements:

- A pipeline for talented students and graduates prepared directly to enter IBM business processes is established.
- There is a contract between FI and IBM on a number of students which can pass 5 month full time project each semester in IBM department the scope of which is principally connected with service systems (their creation or operation)
- Experts from IBM management and strategic research and development teams are associated members of scientific boards influencing education process and curricula at FI.

- A list of diploma thesis topics originated in business practice of IBM is established together with guarantors from IBM and is continually amended.
- Research problems arising from IBM business practice are investigated and studied at FI laboratories.
- Grants are regularly obtained from IBM supporting FI research.
- A list of annotated research projects solved on FI together with short actual situation messages is at disposal for IBM
- FI is on the list of well co-operated schools published in official web pages of IBM.

I think you can imagine a similar procedure with the other example. In next chapters we will have opportunity and better means to express this approach to two-directional value proposition formulation over a common target of two co-operating prime service systems. To simplify the phrasing I propose a following definition which help us to speak on one service system instead of two co-operating prime service systems.

Definition 5 (Dual Service System) *Let C, P, T stand for Client, Provider, Target, respectively.*

Let us have a general service system such that two prime service systems could be identified in it using two main contexts. Let the first one is (C_1, P_1, T_1) *and the second one is* (C_2, P_2, T_2). *If it holds that*

1. $C_1 = P_2$ *and* $P_1 = C_2$
2. $T_1 = T_2$
3. *value proposition of* (C_1, P_1, T_1) *and value proposition of* (C_2, P_2, T_2) *are formulated such that their union creates a two-directional value proposition covering benefits for both* C_1 *and* C_2

than we say that the service system (C_1, P_1, T_1) *or the service system* (C_2, P_2, T_2) *is a* dual service system.

This means: a dual service system is a system containing two basic contexts. When switching the contexts we switch from the first agent's benefit to the second agent's benefit and vice-versa. There exists one common target. This help us to formulate comprehensible value proposition that balances benefits of both agents playing the roles of Client/Provider and Provider/Client.

Dual service systems can be utilized in customer - supplier relationships so as in an employee - employer relationships. Using this concept the contract between both two parties could be established in more visualized and understandable form. We will use this in chapter 9 on service systems evaluation.

A last comment in this chapter we will devote to a service system complexity understanding. Let us imagine a general service system composed of tens of agents, some pieces of technology and maybe of some environment. Without doubts we will consider such a system to be a complex one. Surely there will be lot of contexts in such a service system from which we can recognize particular service triangles client-provider-target. Simply the agents, maybe together with technology and/or environment, are composed in various ways.

Let's, on the other hand, we have a service system containing two agents in the roles of client and provider, respectively, and an agent or technology or environment playing the role of target. Suppose the agents are again service systems. Maybe the agents of those service systems are again service systems, and so on. Maybe those „second level“ service systems are complex ones as we mention above. Let there is one or maximally two interesting contexts from which the roles of client and of provider are spread on the agents. Is there any reason to think on such service system as on a complex one? I propose no. Such system is a simple one. The complexity is maybe hidden in the embedded service systems, but it is not manifested in the referred service system. So, an agent, which itself is a very complex service system can play a role in a service system which is a simple one. Remember that a typical example of this situation is any service system in which a human play some role.

Chapter 4

Cyberspace

⋇ [STA]$_{31}$:IN WORK. TO FINISH UP TO LANGUAGE CORRECTIONS ASAP. ?10/2009

Cyberspace is an often used term in modern epoch and we understand it intuitively well when used as is usual. Internet, e-mail, electronic documents, and a lot of others, are „inhabitants“ of this space.

But when I put a question: What is more important for our acting in real processes in business or in our lives–is it the real three-dimensional space together with the time dimension or is it the cyberspace? What will be the answer?

A very first idea for an answer is: the real three-dimensional space with time, i.e., the physical space, of course.

But let us turn our attention to Daniel Dennett book Kinds of Mind [1] and to Jeff Hawkins On Intelligence [**?**] (cf. section Inspiration in the Introduction and more in 4.2). Reading these books from the beginning to to the end and backward we can find only one solution for this question: Any decision or any action we take in any situation comes solely from our mind–from our images, our visions, our ideas. It does not depend directly on the state of affairs in the real world. It depends exclusively on our seeing of the state of affairs.

Of course, our seeing of the state of affairs depends on this state of affairs! But the state of affairs is the input only. All other processes after this input, i.e. reasoning, decision making, any kind of acting, are directly driven solely by our mind, i.e. by products and/or services of our brain.

So, ideas, images, concepts, visions, their records not only in our brains, but on any kind of electronic or any other computer accessed media are very important when thinking on service systems as composites of people, technology, other service systems, glued by information and knowledge.

I propose to precise the term „cyberspace“ (at least for our purposes in this book) meaning by it a virtual space populated by ideas, images, concepts, visions, their records in our brains or on any kind of electronic or any other computer accessed media.

Thus, cyberspace is a space of our ideas and concepts as well as of their records by means of information technology (IT).

Maybe you are surprised that I do not make a difference between this what dwells in our heads (in our brains) and this what is materialized in a kind of IT.

Lot of things here in this book is nontraditional. You will see that almost everywhere I[1] am going in opposite direction to the main stream. When others solve the problem bottom-up, we offer a top-down solution. When others speak on objects as the most important subject, we speak on connections. When others create boundaries and remind us to be careful to cross the boundaries as it can cause misunderstandings and chaos in thinking, we break the borders and take a benefit from this. Where others struggle with syntax to understand a message coping with semantics at least at the end of the process, we use semantics for message recognition and understanding from the very beginning while syntax sometimes will help us closing the process. And, when others separate us–people from systems–technology, I break this boundary. This is why I am so enthusiastic about Service Science.

This chapter could be named as well „Value co-creation, Information, Knowledge, Semantics, Cognition“. We would like to understand deeply what is value co-creation in general and what is value co-creation in each particular situation of a given service system. A plan for value co-creation is the value proposition formulated between a client and a provider. Shared understanding is necessary to formulate such a value proposition, and shared information, knowledge and cognition is necessary to evaluate whether the value co-creation is successful. And, above all, let us be aware of the definition of service system: it is a composite of people and technology. What portion is technology and what portion is people is not defined. This can change during a service system lifecycle. An ultimate requirement grows from this situation: we need such technology which allows to develop easy communicable agents as players in the service system game.

Thus we engage with information, knowledge, and cognition not for a

[1]together with all my and Filip's team

pleasure of an ancient theorist[2] but for better understanding what value co-creation is, how to specify it, how to formulate value propositions, how to evaluate benefits of interested parties, how to monitor progress, and for better understanding between us–people and artificial agents–computer systems.

This will be a non easy journey. This will be journey to ultimate semantics understanding, nevertheless we will deal with different conceptual systems and share this conceptual systems and the semantics understanding across different types of agents–at least human ones and artificial ones.

This will be a journey with lot of theoretical and very abstract contemplations, definitions and statements. Yes, this is very nontraditional in a book dealing with SSME. The usual approach is to speak on processes, providers, customers, their relationships, strategies, marketing rules. The usual approach is speak or write in natural language sometimes embroidered by tables, diagrams and pictures. Instead of this I will use lambda calculus and mathematical formulae.

So, by by my dear reader! I feel I will loose lot of you just here in this chapter. However, I am sorry, but there is no king's road to complex systems and complex situation understanding. My experience, and thus my opinion, is: All this is about intelligent coping with semantics; hence we need to understand well what concepts are, what conceptual systems are, how to model semantics, how artificial agents can work with semantics.

Well, I try to do the best for the business practice. I try to introduce a good theory, here.[3] I will try, where it is possible, to express concepts and thoughts naturally, and to add exact meanings in a formal apparatus as illustrations.

4.1 Value Co-creation and its issues

A hot problem was stated in the previous chapter in the section 3.3. Let's speak on this issue in a concrete way. A generalization will be obvious. The questions arising from the mentioned section are:

- How to recognize whether the value proposition of the IBM benefit system is in a good balance with the value proposition of the FI benefit system?

[2] when I was young, I was. By education I am mathematician.

[3] This is a paraphrase of the Dines Bjorner sentence from the ancient SOFSEM seminar: „There is nothing better for practice than a good theory“.

- How to recognize whether the deliverables of the IBM benefit system is balanced to the deliverables of the FI benefit system?

To solve these questions we need to compare the value added in the IBM benefit system to the value added in the FI benefit system.

To compare these added values we need

- to have a necessary information and a necessary knowledge to formulate adequate value propositions, maybe in different conceptual systems
- to share understanding to these value propositions, despite they are formulated in different conceptual systems
- to have such cognition capabilities that help to understand what we are monitoring

Thus, value co-creation is usually a matter of different conceptual systems mutual understanding.

The ongoing change in business becomes the only reliance we have. A Service System is a composite of people, technology, organization coupled with information and knowledge. A crucial point start to be a construction of the technology components of the service system, namely of SW systems embedded in the service system. The first business requirement is instant adaptability. People are adaptable. Embedded SW components have to be too.

Understanding is not only a problem of human beings but of systems supporting them, too. We need to avoid barriers of understanding between people of different professions and between people and systems, too. This is the crucial role of semantics. We need to share meanings without boundaries. Hence, the models have to cover semantics. On the other hand, models have to be in a way executable. A model execution must follow the semantics.

This considerations can be summarized into the following statement:

Statement 1 (Value co-creation and conceptual systems.) *Value co-creation provided by agents within a knowledge intensive Service System ultimately requires*

- *by all these agents shared understanding of semantics,*
- *by all agents shared information,*

- *by all agents shared knowledge.*

Conceptual systems of particular agents could differ each from other; understanding of semantics must be ensured across all boundaries of particular conceptual systems.

Conceptual systems have to be dealt with explicitly not implicitly or in a hidden manner.

The statement entitles me to continue with the non-easy chapter about content of cyberspace. I will start with quotations from a Jeff Hawkins book „On Intelligence“ which brings lot of intuitions about our brain functioning and thus about what knowledge is and how to handle with knowledge. Those intuitions will be supported and explicated from a logical point of view in the next section, very abstract one. An exact and effective way to understanding a domain and to recording obtained pieces of knowledge in a form of (conceptual) model will be introduced in the following section 4.4 on Domain Understanding and Modeling. The rest section of this chapter try to bring clarity into such topics as knowledge management and universal modeling, and to picture it from the Service Systems point of view.

Some portions of the content of cyberspace are stored in information systems. Information systems have to be developed so that they are prepared for storing of actually needed portions of the cyberspace content. This could be very dynamic and hardly predictable requirements.

The classical paradigm of IS development (requirements collection - analysis - design - programming and testing - running and evaluation) seems to be not sufficient for this purpose. The tool we need have to be rather a modeling tool than an information system.

We need to model the business, and using obtained model to analyze possibilities, choose the best way, and support this way of business by the model. The model is treated as it would be an information system in this case. When changes in business will occur we record them into the model immediately and go on with supporting the business by this way changed model.

Changes in such a business model could be very unexpected. Often a new concepts and new methods are introduced. Thus we need a powerful modeling tool, enough general to cope with new concepts, new methods or operations, and new connections, not known in time the modeling tool is created.

The stumbling block for achieving this by existing modeling tools is their basic principle: the „world" which is modeled is strictly separated from the „world" by which the modeling is done. When new concepts and new operations appear in the modeled business world we have no possibility to push them into the modeling world and manipulate them without re-programming our modeling tools. This separation is currently seen as a natural and necessary thing to achieve an understandable and systematic view of a particular business. The reason can be that if we do not separate the modeling and modeled worlds, we can lose our focus and fall into unmanageable chaos.

But, when speaking on new concepts and connections and new operations in business using our natural language and our natural way of thinking, we are able to manage both–the modeled world and the modeling world. We define this new concepts, new connections, and new operations by means of our natural language and our natural conceptual system, and immediately we start to use them, again by means of our natural language and our natural conceptual system. This could be our inspiration for an universal modeling tool that do not separate the modeled and the modeling worlds.

Thus some ideas of theory of concepts and logical analysis of natural language expressions will help us. The fundamental element will be a notion of category. We will speak of them–mention their various connections–and together with this we will use them to speak of their members and of those members connections. Facts and rules describing the „world of categories" will help us to propose an architecture of universal SW component that can serve as above mentioned powerful modeling tool.

My opinion is: when one wants to use methods and procedures effectively, he/she needs to understand why the methods and procedures look as they look, and he/she needs to understand underlying principles. Thus, regardless of low popularity of theory, I will lead you through it.

4.2 Jeff Hawkins Hypothesis on Intelligence

⋇ [STA]$_{32}$:TODO: WILL BE REMARKABLY SHORTEN TO THIS WHICH IS NECESSARY TO UNDERSTAND NEXT CHAPTERS.

Next chapters' explications will often built on Jeff Hawkins hypothesis published in his nice book „On Intelligence". As I cannot suppose you red this book or you break the reading now and turn to Jeff Hawkins before next chapters, I put there lot quotations which will be useful in future. And as our brains are containers to lot of cyberspace elements it is natural to put this quotations here.

„Complexity is syndrome of confusion, not a cause.“ ([?], p. 5)

„A fully mature theory will take years to develop, but that doesn't diminish the power of the core idea.“ ⋇ [STA]$_{33}$:cite Hawkins, p. 7([?], p. 7)

„The most important attribute of a computer program or a neural network is whether it gives the correct or desired output. As inspired by Alan Turing, intelligence equals behavior. But intelligence is not just a matter of acting or behaving intelligently. Behavior is manifestation of intelligence, but not the central characteristic or primary definition of being intelligent.“ ⋇ [STA]$_{34}$:cite Hawkins, p. 29([?], p. 29)

Jeff Hawkins had proposed three essential criterions to understand the brain.

Inclusion of time in brain function. „Real brains process rapidly changing streams of information. There is nothing static about the flow of information into and out of the brain.“ ⋇ [STA]$_{35}$:cite Hawkins, p. 25([?], p. 25)

Importance of feedback. „... for every fiber feeding information forward into the neocortex, there are ten fibers feeding information back toward the senses. Feedback dominates most connections throughout the neocortex as well. ... It must be important.“ ⋇ [STA]$_{36}$:cite Hawkins, p. 25([?], p. 25)

Physical architecture of the brain. „... any theory or model of the brain should account for the physical architecture of the brain. The neocortex is not a simple structure ... it is organized as a repeating hierarchy.“ ⋇ [STA]$_{37}$:cite Hawkins, p. 25([?], p. 25)

The architecture of brain: —

Areas of brain: „brain regions are semi-independent and seem to be specialized for certain aspects of perception or thinking. Physically they are arranged in an irregular patchwork, which varies only a little from person to person. Rarely are the functions cleanly delineated. Functionally they are arranged in a branching hierarchy.“ ⋇ [STA]$_{38}$:cite Hawkins, p. 44([?], p. 44)

Next quotations give us the sense of hierarchy as Jeff Hawkins work with. The hierarchy is a crucial concept and phenomenon. We need to understand it from various points of view. We need to be prepared to compare the hierarchy of neocortex with constructions and with ramified type system in TIL and with hierarchies constructed using diamonds from the Diamond Path Framework.

„The human brain has an incredible capacity to learn and adapt to thousands of environments that didn't exist until very recently. This argues for an extremely flexible system, not one with a thousand solutions for a thousand problems.“ ⋇ [STA]$_{39}$:cite Hawkins, p. 53([?], p. 53)

„Cells were not born to specialize in vision or touch or hearing.“ ⋇ [STA]$_{40}$:cite Hawkins, p. 54([?], p. 54)

„If you connect regions of cortex together in a suitable hierarchy and provide a stream of input, it will learn about its environment.“ ⋇ [STA]$_{41}$:cite Hawkins, p. 55([?], p. 55)

At the beginning of this chapter I introduced a meaning of the word cyberspace as it will be helpful in our next considerations. At the same time I argued for the importance of cyberspace in our lives. The following quotation deals with this: „This is not to say that people or objects aren't there (STA: in the real three-dimensional space with time). They are really there. But our certainty of the world's existence is based on consistency of patterns (STA: in our brains) and how we interpret them.“ ⋇ [STA]$_{42}$:cite Hawkins, p. 55([?], p. 55)

Jeff Hawkins cites Vernon Mountcastle. Mountcaslte suggested that since all regions of brain look the same, perhaps they are actually performing the same basic operation! He proposes that the cortex uses the same computational tool to accomplish everything it does.

In a field of anatomists looking for minute differences in cortical regions, he shows that despite the differences, the neocortex is remarkably uniform.

The thing that makes the vision area visual and the motor area motoric is how the regions of cortex are connected to each other and to other parts of the central nervous system.

The cortex does something universal that can be applied to any type of sensory or motor system. [4] ⋇ [STA]$_{43}$:cite Hawkins, pp. 50-51([?], pp. 50-51)

The memory. „The cortex creates what are called invariant representations, which handle variations in the world automatically.“

[4]Hawkins cites Mountcastles paper 'On Organizing Principle for Cerebral Function' (1978)

There are four attributes of neocortical memory: „(1) The neocortex stores sequences of patterns. It's almost impossible to think of anything complex that isn't a series of events or thoughts. (2) The neocortex recalls patterns auto-associatively. (3) The neocortex stores patterns in an invariant form. (4) The neocortex stores patterns in a hierarchy.“ ⁕ [STA]$_{44}$:cite Hawkins, pp. 69-70([?], pp. 69-70)

„Truly random thoughts don't exist. Memory recall almost always follows a pathway of association.“ ⁕ [STA]$_{45}$:cite Hawkins, p. 71([?], p. 71)

„At any time a piece can activate the whole. This is the essence of auto-associative memories.“ ⁕ [STA]$_{46}$:cite Hawkins, p. 74([?], p. 74)

The new framework of intelligence. „Your brain has made a model of the world and is constantly checking that model against reality. You know where you are and what are you doing by the validity of this model.“

„Intelligence is measured by the capacity to remember and predict the patterns in the world, including language, mathematics, linguistics, physical properties of objects and social situations. Your brain receives patterns from outside world, stores them as memories, and makes predictions by combining what it has seen before and what is happening now. ... Prediction and behavior are not completely separate. ... (From the historical point of view) behavior comes first, then intelligence. ... most of what we sense is heavily dependent on what we do and how we move in the world. Therefore prediction and behavior are closely related.“ ⁕ [STA]$_{47}$:cite Hawkins, p. 97([?], p. 97)

„Here then is the core of my argument on how to understand the neocortex, and why memory and prediction are the keys to unlocking the mystery of intelligence. We start with the reptilian brain with no cortex. Evolution discovers that if it tacks on a memory system (the neocortex) to the sensory path of the primitive brain, the animal gains an ability to predict the future. Imagine the old reptilian brain is still doing its thing, but now sensory patterns are simultaneously fed into the neocortex. The neocortex stores this sensory information in its memory. At a future time when the animal encounters the same or a similar situation, the memory recognizes the input as similar and recalls what happened in the past. The recalled memory is compared with the sensory input stream. It both 'fills in' the current input and predicts what will be seen next. By comparing the actual sensory input with recalled memory, the animal not only understands where it is but can see into the future. ... We don't even have to assume the cortex knows the difference between sensations and behavior; to the cortex they are both just patterns. When our animal finds itself in the same or a similar situation, it not only sees into the future but recalls which behavior led to that future vision. Thus, memory and prediction allow an animal to use its existing (old brain) behaviors more intelligently.“ ⋇ [STA]$_{48}$:cite Hawkins, pp. 99-100([?], pp. 99-100)

The memory-prediction framework. „Move your arm in front of your face. To predict seeing your arm, your cortex has to know that it has commanded the arm to move. If the cortex saw your arm moving without the corresponding motor command, you would be surprised. The simplest way to interpret this would be to assume your brain first moves the arm and then predicts what it will see. I believe this is wrong. Instead I believe the cortex predicts seeing the arm, and this prediction is what causes the motor commands to make predictions come true. You think first, which causes you to act to make your thoughts come true.“ ⋇ [STA]$_{49}$:cite Hawkins, p. 102([?], p. 102)

„... intelligence and understanding started as a memory system that fed predictions into the sensory stream. These predictions are the essence of understanding. To know something means that you can make predictions about it. ... The human cortex ... is constantly predicting what you will see, hear, and feel, mostly in ways you are unconscious of. These predictions are our thoughts, and, when combined with sensory input, they are our perceptions. ... To make predictions of future events, your neocortex has to store sequences of patterns. To recall the appropriate memories, it has to retrieve patterns by their similarity to past patterns (auto-associative recall). And, finally, memories have to be stored in an invariant form so that the knowledge of the past events can be applied to new situations that are similar but not identical to the past.“ ⋇ [STA]$_{50}$:cite Hawkins, pp. 104-105([?], pp. 104-105)

Please pay attention to this last quotation. Again I will repeat what I feel very important:

„To know something means that you can make predictions about it.“

„Predictions are our thoughts. Predictions combined with sensory input form our perception.“

The usual conviction is: first there is perception; on the base of perception there had been developed thinking as something pasted to the perception. Jeff Hawkins hypothesis, which I strongly belief, says no! The perception is pasted to the thinking. When a jump in evolution from reptiles to mammals (mammals have neocortex, reptiles haven't) were made, the perception made a giant step in the quality of perception. Our perception is based on our thinking not vice versa.

How the cortex works

Invariant representations. „...when you begin to realize that the cortex's core function is to make predictions, then you have to put feedback into the model; the brain has to send information flowing back toward the region that first receives the inputs (STA: from outside or from more peripheral regions). Prediction requires a comparison between what is happening and what you expect to happen. What is actually happening flows up (STA: from peripheral to core), and what you expect to happen flows down (STA: from core to peripheral).

The same feedforward-feedback process is occuring in all your cortical areas involving all your senses." ⋇ [STA]$_{51}$:cite Hawkins, pp. 113([?], p. 113)

On page 114 Jeff Hawkins brings a conceptual scheme of forming the invariant representation. This picture is a scheme showing how an invariant representation is „constructed" from hearing, vision and touch. I cite this figure here as fig. 4.1 as I consider it a very important to understand brain functioning as well as Service Systems functioning. My opinion is that the invariant representation forming is a general way of concepts discovering for both, the people and the Service Systems.

The „four floors" in the three columns schematically depicts cortex's first four regions in the recognition of objects (for the vision named by neuroscientists V1, V2, V4, IT), while the top of this picture is purely conceptual diagram. The feedforward of information from peripherals to the core is depicted by upward arrows, the feedback going from the core to more peripheral regions or to peripherals is depicted by downward arrows. Note, that the feedback is more than ten times rich than the feedforward.

But the picture on fig. 4.1 is a simplification made for didactic purpose only. In fact there is no reason to see upper regions of cortex receiving converging inputs from more regions, while lower regions (like V1, V2, V4) receive only one input from its predecessor. For this reason and other considerations Jeff Hawkins proposed an alternate view of the cortical hierarchy, which is on fig. 4.2

The regions from the figure 4.1 are in fact collections of many smaller

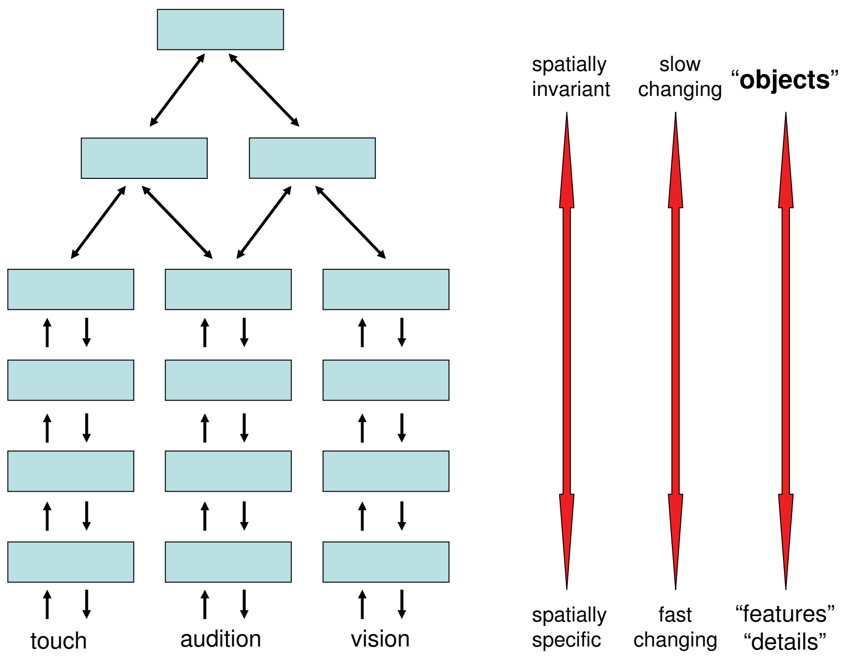

Figure 4.1: Forming invariant representations in touch, hearing, vision, and all together.

regions. The V1 from this figure have the largest number of small subregions. V2 is also composed of fewer, slightly larger subregions. The same is true of V4. But the top region, IT, is really a single region „which is why IT cells have a bird's-eye view of the entire visual world.“ ⋇ [STA]52:cite Hawkins, p.122([?], p. 122)

„There is a pleasing symmetry here on the picture of fig. 4.2 ... Note that now the cortex looks similar everywhere. ... the job of any cortical region is to find out how its inputs are related, to memorize the sequence of correlations between them, and to use this memory to predict how the inputs will behave in the future. Cortex is cortex. The same process is happening everywhere: a common cortical algorithm. ⋇ [STA]53:cite Hawkins, p.123-124([?], pp.

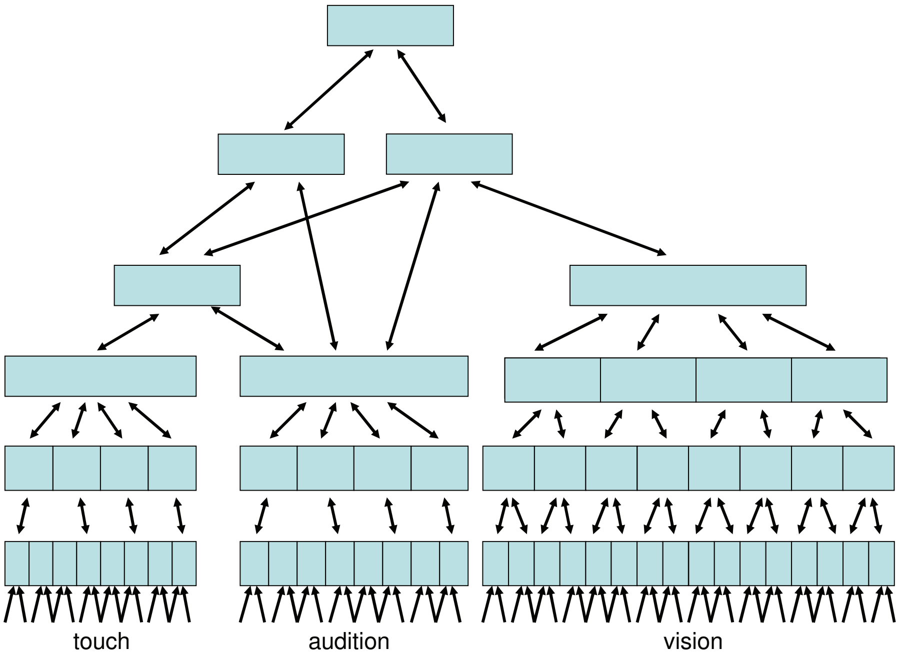

Figure 4.2: More real picture of the cortical hierarchy.

123-124)

And this algorithm is **prediction**. Nothing more and nothing less.

A model of the world. „All objects in your world are composed of sub-objects that occur consistently together; that is very definition of an object. When we assign a name to something, we do so because a set of features consistently travels together. A face is a face precisely because two eyes, nose and a mouth always appear together. ...

Every object in the world is composed of a collection of smaller objects, and most objects are parts of larger objects. This is what I mean by nested structure. Once you are aware of it, you can see nested structures everywhere. ... Large-scale relationships are stored at the top of hierarchy and small-scale relationships are stored toward the bottom. ...

The design of the cortex and the method by which it learns naturally discover the hierarchical relationships in the world. ... The cortex has a clever learning algorithm that naturally finds whatever hierarchical structure exists and captures it. If the structure is absent, we are thrown into confusion, even chaos.“ ⋇ [STA]$_{54}$:cite Hawkins, pp. 126-127([?], pp. 126-127)

I would like to add that this is why the Tony(?) Buzan's Mind Maps work so well. Lot of people and lot of organizations use them to describe their aims or plans and this seems to be very effective. Tony Buzan[5]in his widely known book on mind maps explains this fact as a result of similarity of mind maps hierarchy with brain hierarchy. We can confirm this is true. And, moreover, we can add: not only similarity with brain hierarchy, but also similarity with the world principal hierarchy, as the brain hierarchy is a copy of this world hierarchy.

„Since we can only touch, hear, and see a very small part of the world at any moment in time, information flowing into the brain naturally arrives as a sequence of patterns. ... A sequence is set of patterns that generally accompany each other but not always in fixed order. What is important is that patterns of a sequence follow one another in time even if not in fixed order. ...

The brain treats abstract and concrete objects in the same way. ...

Predictability is the very definition of reality. If a region of cortex finds it can reliably and predictably move among these input patterns using a series of physical motions (such as saccades of the eyes ...) and can predict them accurately as they unfold in time, the brain interprets these as having casual relationship. ...

Therefore, the brain can be said to store sequences of sequences. Each region of the cortex learns sequences, develops what I call "names„ for the sequences it knows, and passes these names to the next regions higher in the cortical hierarchy.“ ⋇ [STA]$_{56}$:cite Hawkins, pp. 127-129([?], pp. 127-129)

I recommend you to read Alva Noe's book called „Action in Perception“ [10]and to compare the ideas.

Maybe now my above[6] defense of the role of cyberspace starts to be more clear and understandable.

[5]the author of mind maps concept and conception; [?] ⋇ [STA]$_{55}$:Tony Buzan Mind Maps

[6]cf. the introduction to the chapter 4

Sequences of sequences. „As information moves up from primary sensory regions (of the neocortex) to higher levels, we see fewer and fewer changes over time. ... Each region of cortex has a repertoire of sequences it knows, analogous to a repertoire of songs. Regions store these song-like sequences about anything and everything ...

We have names for songs, and in a similar fashion each cortical region has a name for each sequence it knows. This "name„ is a group of cells whose collective firing represents the set of objects in the sequence.

It's as if region were saying: 'Here is the name of the sequence that I am hearing, seeing, or touching. You don't need to know about the individual notes, edges, or texture. I will let you know if something new or unpredicted happens.'

We can imagine region IT[7] at the top of the visual hierarchy relaying to an association area above it, 'I am seeing a face. Yes, with each saccade the eyes are fixating on different parts of the face; I am seeing different parts of the face in succession. But it is still the same face. I will let you know when I see something else.'

By collapsing predictable sequences into 'named objects' at each region in our hierarchy, we achieve more and more stability the higher we go. This creates invariant representations. ...

The opposite effect happens as a pattern moves back down the hierarchy: stable patterns get 'unfolded' into sequences. ...

The lower you look in the hierarchy, the faster the patterns are changing.

The memories of the words are handled as invariant representations; it doesn't matter whether you will speak, type, or handwrite them. ...

Representations of simple objects at the bottom of the hierarchy can be reused over and over for different high-level sequences." ⋇ [STA]$_{57}$:cite Hawkins, pp. 129-131([**?**], pp. 129-131)

To give evidence that the principle is the same in case of our perceiving of the time-space world and in case of our perceiving of the cyberspace I will give an example from a domain which is completely made-up.

Let us compare the above cited reasoning with a usual way of materializing a strategy of an interrelated set of projects. Such a program (i.e., a collection of interrelated projects) strategy can be visualized by the so called logical framework matrix (LFM)[8] consisting of four rows (goals, objectives, deliver-

[7]a biologists label for the top visual region of neocortex

[8]see e.g. TODO cite LFM

ables, and actions) and four columns (name of the goal/objective/deliverable/activity, indicators of success, data sources for the indicators, and outside conditions).

The four rows and the four columns of LFM represent a framework very consistent with the above described functioning of cortex. We can metaphorically map the matrix cells into cortex regions and to see the same approach is emerging. To understand what the program is about and what particular project in the program is we need to make in a way series of mental motions on the matrix cells. Without this we cannot „see" what the program really is.

⋇ [STA]$_{58}$:TODO maybe needs more explanation.

„The process of forming sequences pay off when an input is ambiguous. ... You use the context of known sequences to resolve ambiguity." ⋇ [STA]$_{59}$:cite Hawkins, p. 134([?], p. 134)

I would like to emphasize that context is much more important than we usually think of. And the context is nothing more than a known sequence of names, which in turn are collapsed sequences, etc.

„The memory of sequences allows you not only to resolve ambiguity in the current input, but also to predict which input should happen next. ... By recognizing a sequence of patterns, a cortical region will predict its next input pattern and tell the region below what to expect.

A region of cortex not only learns familiar sequences, it also learns how to modify its classifications." ⋇ [STA]$_{60}$:cite Hawkins, p. 135([?], p. 135)

„In cortical regions, bottom-up classifications and top-down sequences are constantly interacting, changing throughout your life. This is essence of learning. ... Forming new classifications and new sequences is how you remember the world." ⋇ [STA]$_{61}$:cite Hawkins, p. 136([?], p. 136)

„The hierarchy of the cortex ensures that memories of objects are distributed over the hierarchy; they aren't located in a single spot. Also, because each region of the hierarchy forms invariant memories, what a typical region of cortex learns is sequences of invariant representations, which are themselves sequences of invariant memories. You will not find a picture of a cup or any other object stored in your brain.

Unlike a camera's memory, your brain remembers the world as it is, not as it appears. ... The sequences by which you experience the world is the real structure of the world, and that is what the cortex wants to remember." ⋇ [STA]$_{62}$:cite Hawkins, p. 137([?], p. 137)

Regions of cortex are on one hand hierarchically interconnected and on the other hand they are formed from columns of cells. The term column is used because these clusters of cells are in a way perpendicular to the horizontal layers.

„You can think of columns as being vertical 'units' of cells that work together.“ ⋇ [STA]$_{63}$:cite Hawkins, p. 139([?], p. 139)

„The large number of synapses connecting cells in a column to other parts of the brain provide each column with the context it needs in order to predict its activity in many different situations.“ ⋇ [STA]$_{64}$:cite Hawkins, p. 141([?], p. 141)

„Columns learn to fire in the right context and in the correct order.“ ⋇ [STA]$_{65}$:cite Hawkins, p. 149([?], p. 149)

If the column could speak it will say: „I (column) represent our interpretation of the world, regardless of whether it is true or just imagined.“ ⋇ [STA]$_{66}$:cite Hawkins, pp. 154-155([?], pp. 154-155)

This is a very important piece of knowledge. It again shows that cyberspace is this what matters in all our actions and decisions.

„This mechanism not only make specific, predictions, it also resolves ambiguities from the sensory inputs. Very often the input to a region of cortex will be ambiguous, when you hear a semi-garbled word, e.g. This bottom-up/top-down matching mechanism enables you to decide between two or more interpretations. And once you decide, you relay your interpretation to the region below.

Every moment in your waking life, each region of your neocortex is comparing a set of expected columns driven from above (regions hierarchy) with the set of observed columns driven from below (regions hierarchy). Where the two set intersects is what we perceive. ....

Finally, in addition to projecting to lower cortical regions, layer 6 cells can send their output back into layer 4 cells of their own column - when they do, our predictions become input. This is what we do when daydreaming or thinking. It allows us to see the consequences of our own predictions. We do this many hours a day as we plan the future, rehearse speeches, and worry about events to come. Longtime cortical modeler Stephen Grossberg calls this 'folded feedback'. I prefer 'imagining'.“ ⋇ [STA]$_{67}$:cite Hawkins, p. 156([?], p. 156)

Four Hawkins' predictions

In the last chapter „The future of intelligence" of his book Jeff Hawkins answers some peculiar questions: „Can we build intelligent machines?" „Should we build intelligent machines?" „Why build intelligent machines?" And he makes important predictions on aspects of artificial memory-prediction systems in which they will scale dramatically beyond our biological brains. These attributes that will exceed our own abilities are: speed, capacity, replicability, and sensory system.

Speed. „While neurons work on the order of milliseconds, silicon operates on the order of nanoseconds (and is still getting faster). That's a million-fold difference, or six orders of magnitude. ... Intelligent machines will be able to think as much as a million times faster than the human brain. ... If an intelligent machine conversed or interacted with a human, it would have to slow down to work at human speed. ... But when interfacing with the electronic world [9], it could function much more quickly. ... Two intelligent machines could hold a conversation a million times faster than two humans." ⋇ [STA]$_{68}$:cite Hawkins, p. 224([?], p. 224)

Capacity. „Despite the impressive memory capacity of a human cortex, intelligent machines can be built to surpass it by large margins. The size of our brains has been constrained by several biological factors ... But we can build intelligent memory systems of any size ...

As we build intelligent machines, we could increase their memory capacity in several ways. Adding depth to the hierarchy will lead to deeper understanding–the ability to see higher-order patterns. Enlarging the capacity within regions will allow the machine to remember more details, or perceive with greater acuity ... And adding new senses and sensory hierarchies permits the device to construct better models of the world ..." ⋇ [STA]$_{69}$:cite Hawkins, p. 225([?], p. 225)

Replicability. „Each new organic brain must be grown and trained de novo, a process that takes decades in human beings. Each human must discover for himself or herself the basics of coordinating the body's limbs ..., the names of things and the structure of the language; and the rules of family

[9] I say the cyberspace

and society. ... Every individual must trudge up the same set of learning curves in life ... all to build a model of the world in the cortex.

Intelligent machines need not undergo this long learning curve, since chips and other storage can be replicated endlessly and the contents transferred easily. ... intelligent machines could replicate like software. Once a prototype system has been satisfactory honed and trained, it could be copied as many times as we please. ...

It should be possible to share components of learning the way we share components of software. An intelligent machine of a particular design could be reprogrammed with a new set of connections to lead to different behavior, as if I could download a new set of connections into your brain and instantly change you from an English speaker to a French speaker ... Sharing expertise in this way is simply impossible for humans. ... Reprogramming an intelligent machine will not be too different from running a new video game or installing a piece of software.“ ⋇ [STA]$_{70}$:cite Hawkins, pp. 226-227([?], pp. 226-227)

Sensory system. „Humans have a handful of senses. ... We can't change them. Sometimes we use technology to augment our senses, such as with night-vision goggles, radar, or the Hubble space telescope. These high-tech instruments are clever tricks of data translation, not new modes of perception. ...

Our intelligent machines could perceive the world through any sense found in nature as well as new senses of purely human design. ...

It is in the realm of exotic senses that, I suspect, the revolutionary uses of intelligent machines lie. ...

Imagine weather sensors spaced every fifty or so miles across a continent (e.g.). ...

Other large distributed sensory systems could allow us to build intelligent machines that understand and predict animal migrations, changes in demographics, and the spread of disease. Imagine having sensors distributed over a country's electrical power grid. An intelligent machine attached to these sensors would observe the ebb and flow of electricity consumption in the same way you and I see the ebb and flow of traffic on a road, or the movement of people at an airport. ... We might combine sensors for weather and for human demographics, in order to anticipate political unrest, famines, or disease outbreaks. ... You might think intelligent machines would need emotions to foresee patterns involving human behavior, but I don't think so.

We are not born with a set culture, a set of values, and a set religion; we learn them. And just as I can learn to understand the motivations of people with values different from mine, intelligent machines can comprehend human motivations and emotions, even if the machine doesn't have those emotions itself. ... Finally, we might unite a bunch of intelligent systems in a grand hierarchy, just as our cortex unites hearing, touch, and vision higher up the cortical hierarchy. Such a system would automatically learn to model and predict the patterns of thinking in populations of intelligent machines." ⁕ [STA]$_{71}$:cite Hawkins, pp. ?-?([?], pp. ?226-227)

—

Why this attempt to rewrite a Hawkins book into this book? Was it necessary to do it? I think yes, it is useful. As I told in the Introduction, the coincidence of the processes of research in knowledge and information robots domain at university and of development of technologies based on these research results at the university spin-off on one hand, and thoughts about Service Science and about what intelligence is based on the Hawkins book, on the other hand, had brought fruitful issues. The following reasoning will be less comprehensible without the Jeff Hawkins ideas.

From the service system point of view these Hawkins four predictions immediately lead to intelligent service systems. To describe such systems, or similar systems with a kind of memory-prediction model involved we need to understand deeply what are the objects dwelling in the cyberspace. Thus, we will continue in the next section completely different. As logic is the science which explains what do we talk about, i.e. what do we think about, we will try to watch cyberspace from the logical point of view.

4.3 From The Logical Point of View

This section could be named „Data, Information, Semantics, Concepts", as well, as data, information, semantics and concepts are the key words for the following text. But I want to stress the overall logic behind these key words and connecting them into solid and useful framework.

Data are uninterpreted records of symbols chosen from a finite set of symbols, the so called alphabet. Such a record or sequence of such records when transmitted or could be transmitted forms a message. Records are finite sequences of symbols, messages are finite sequences of records. Please, remember the Jeff Hawkins hypothesis on sequences here.

According to Shannon the information is a measure of uncertainty decreasing with respect to a result of an action which is obtained when the action is performed. Entropy is related to information. Entropy is a measure of uncertainty. When a receiver obtain information the entropy is decreased.

In the Shannon conception the information is relative to the state of receiver's cognition. If the receiver knows the coming message, this message gives him/her/it no information.

Intuitively, information is connected with natural language. Without natural language there is no possibility to cope with uncertainty and uncertainty decreasing. We cannot explicate what means „receiver knows the coming message" without natural language.

Expressions of natural language (NL) are connected to objects of the real world or the cyberspace, to which we focus, by means of concepts. In fact, expressions of NL code (or we can say represent) these concepts, and at the same time they also denote the respective objects of our focus. For example, the concept of prime number is the same be it coded as „prime number" in English, or as „prvočíslo" in Czech.

Any case it denotes each such number which has just two divisors. Any case it is the same mental procedure using which we can recognize whether a given number is a prime one or not.

Concepts are identification procedures comprised of certain mental steps that lead to the identification of objects on which we focus. These mental steps are very close to invariant representations (see previous section 4.2). I propose to consider these mental steps to be a natural continuation of the process of invariant representations creating in our cortex. It is valid because the cortex regions do not care whether inputs come from outer environment or from inner memories. The important message is: if we are able to describe concepts precisely and to understand them well, we will be able to simulate them and to build intelligent artificial agents which in turn help us

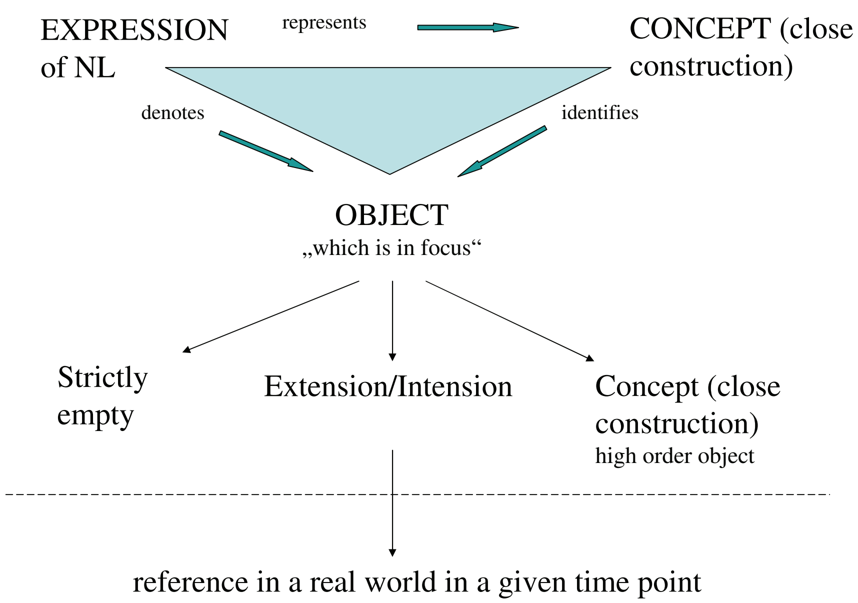

Figure 4.3: Expressions, Concepts, Objects

to establish smarter Service Systems.

Back to the concepts. Without such identification procedures it would not be possible to communicate what we have in our minds, and we could not translate from one natural language into another because we would lose notion of the objects which we focus on. Concepts help us to understand the uncertainty and they help the uncertainty decreasing, at the same time, as was mentioned above.

Concepts may also identify other concepts as the objects of our interest. Such concepts are then called higher order objects. The situation can be seen in figure 4.3.

As we can see on the figure the concepts could be referred to as to close constructions. The meaning of construction will be explained in the following formal description paragraphs. **Concepts are first citizens of cyberspace**.

What could be an object we focus on? And what is the nature of a concept identifying such object? The object to which we focus our attention (and which is denoted by a NL expression and identified by a concept) can be:

- something very common we used to talk on as it is an object of interest– any arbitrary item or a set of such items, a number, truth value or their set, i.e., something independent of the state-of-the-world–the so-called extension,

- something like „the highest building in the world", or „companies with more than one hundred employees", or „students"–i.e., a sort of individual offices or classes of objects dependent on the state of the world (the object of interest is not one particular assignment of these objects to the state-of-the-world, but the very function that assigns particular objects or sets of objects to the states-of-the-world), etc.; in general, the function that maps the states-of-the-world to something–the so-called intension,

- no-object, which means no object of given qualities, regardless of the state-of-the-world, exists–for example „the highest natural number" (in such case we say that the concept identifying this „no-object" is strictly empty)

- another concept, i.e. an identification procedure itself, or a function arguments of which and/or result of which are concepts. An example of it is the „concept of the highest natural number", or the function „The (Truth Value) of the proposition that the (Concept X) is strictly empty", i.e. a function mapping concepts to truth values ((Concept X) $\rightarrow$ (Truth Value)).

The fourth case tell us that there is an infinite hierarchy of concepts: items of the real world are captured by concepts of (in a sense) 1st order. These concepts could be seen as focused objects again. They are captured by concepts of second order. And we can continue without any limits: hawing concepts of order n, when we like or need to conceive them as focused objects, we can capture them by concepts of order n+1. I hope this reminds you the Hawkins' hierarchy mentioned in the previous section.

The second case can give two special possibilities again; if the result is not an usual intension, as was mentioned in this case, it could be one of the following special cases:

- empty set (an extension); a set of objects that have the given qualities is, regardless of the state-of-the-world, empty one–for example „a round square" (we say that the concept that identifies this empty set is quasi-empty in this case),
- not existing object (an intension); the object does not exist in the given state-of-the-world, but it may exist in another state-of-the-world–for example „the present Czech king" (a concept that identifies this „not existing object" is called empirically empty).

Concepts are very important to be understood precisely. When we want to make mutual understanding between people and artificial agents, forming both together a Service System, easier, we do not make do with intuitive comprehension of what concepts are, only. And, moreover, when we want to build up Service Systems on principles similar to architecture and functioning of our brains, we need to catch basic notions, basic ideas, and basic principles with enough precision not to fall into chaos within a big complexity of affairs.

So the serial of formal paragraphs follows describing exact meaning of those topics. This serial is interlard by little bit more human explications. It is possible to read this chapter without these formal description paragraphs and to understand introduced notions intuitively, only. But for the real exact understanding it is necessary to read these formal paragraph, too. But you can postpone this for the second reading if you want.

Formal Description 1 (SIMPLE TYPES) *To explain a nature of concepts we started with functions following the restrictions of the simple hierarchy of types.*

Let B be a base (i.e. a finite collection of non-empty and pairwise disjoint sets). Than we say:

I. *every member of B is a type over B (so called node type),*

II. *if $T_1, \ldots, T_n$, $n > 1$, are types over B, then the cartesian product of $T_1, \ldots, T_n$, denoted by $(T_1, \ldots, T_n)$, is a type over B (so called tuple type or sequence type),*

III. *if T_1 and T_2 are types over B, then $(T_1 \rightarrow T_2)$, i.e. the collection of all partial functions from T_1 into T_2, is a type over B (so called functional type),*

IV. *nothing except I. - III. is a type.*

An element X of a type T is called an object of type T or an T-object, denoted X/T.

A term „sequence“ we will use for finite sequences only, here. It is in accord with the Jeff Hawkins[10] utilization of this term.

A fundamental operation of any agent in the service game, no matter if the agent is a human being or an artificial one–a piece of software, is identification of what is in his/her/its focus. That is the reason why constructions have to be introduced.

Simply saying, a construction is an identification procedure consisting in sequence of mental steps we need to perform to recognize what a particular object of the real time-space world or of the cyberspace, we are focused on, is. A formal description of what is construction in general follows and it is amended in the following formal paragraphs by an application of this general notion to our real world. The aim is to understand elements from which a knowledge is composed.

I will recognize eight modes of constructions. The first four are

1. trivial

[10] in the book „On Intelligence“

2. atomic
3. execution
4. double-execution

A trivial construction is simple pointing an object. I point now: „this current page you are reading now is this what I want to speak of with you". And this was the construction of the current page.

The atomic construction help us to point something not so univocally. E.g. I can say „a page in this book" and I point by this one page of this book but now you don't know which one exactly. We use this construction to express general judgement. The expression „a page of this book" is a variable to which any of this book's pages can be substituted.

The execution construction is not so simple. Let us consider we know what constructions are for a moment. Then I can tell you: take a construction, use it and point the result of using this construction. This means deal with its USE mode (see definition of Service System in def. 2). This execution construction when applied to a construction, e.g. construction C, gives you this what is constructed by C.

Even more peculiar is the double execution construction. Let's start as in case of execution construction. I tell you: take a construction, use it, take the result of its execution and if the result is a construction again, please execute it; this what you obtain is the result of double execution of the first construction.

These last two operations help us to jump freely on the hierarchy of higher order objects. This is very close to operations of our brains–let's remember the previous section quotations.

The second four modes of constructions are

5 sequence

6 projection

7 abstraction

8 application

The sequence construction is a natural image of the Jeff Hawkins sequences. Simply having some objects I put them together and start to handle them as a whole, i.e. not distinguishing the particular objects. This is the most important construction according to my opinion. I am sure this is the most frequent operation we do and Service Systems do. This is the base of our thinking. And, considering the Jeff Hawkins' hypothesis, this is the base of the hierarchical fundament of the world around us.

The projection construction is a reverse operation. Having a sequence we often want to know particular objects of what the sequence is composed of. And this we obtain by projection construction.

The most difficult for understanding is the construction of abstraction. This construction help us to express a generalization of a set of facts consisting in particular assignments of one set of things to another (maybe the same) set of things, i.e. generalization of a set of sequences of the length 2. Each member of such sequence can be a sequence again, but these sequences must be of fixed length and each of them must be of given fixed kind. The abstraction construction constructs function. If we have some prescription or pattern how to assign one thing/item/sequence to another thing/item/sequence, and if we collect the first things/items/sequences into a set called domain and the second things/items/sequences into set called range, we can establish a function mapping elements of the domain to elements of the range. And this is what (construction) abstraction does. As the name of this construction suggests it is really abstract operation.

Once having functions we want to use these functions. The USE mode of functions is important, too. Using a function means to substitute concrete argument(s) for function's variable(s) and compute the result of the assignment represented by this function. As we usually say we apply the function to concrete argument(s). And this is the application construction.

I know that the eight modes is redundant. Theoretically we can manage with a less number of construction modes. Tichy[11] manages with 6 modes–he excludes sequence construction and projection, and whenever a sequence or projection could be appropriate he constructs each such particular object by means of abstraction or application. Materna and Duzi[12] in logic area tried to manage with 4 modes–they exclude from Tichy's 6 modes more execution and double-execution, but they told me recently they start to account for these two modes; Duzi at the domain of informatics uses sequence and projection so as me.

[11] Pavel Tichy a founder of Transparent Intensional Logic, see [17]

[12] Pavel Materna and Marie Duzi are Czech logicians, promoters of TIL

My choice is a pragmatical one. As sequence and projection are so common when working with information and knowledge, and as they are fundamental from the brain construction point of view (cf. Jeff Hawkins ideas in previous section) I prefer to have a construction mode for this purpose and not to derive in each particular case again and again the required result by means of abstraction and application.

Contrarily any function given by finite table can be constructed by using sequences and projections. It is simple: each row in a table is a sequence, and focusing on chosen item (items) is a kind of projection or projections. But avoiding abstraction and application operation from construction modes we lose a comprehensible abbreviation to generalized views of state of affairs which will make the Service System's functioning description more difficult.

Formal Description 2 (CONSTRUCTIONS) *The key notion to understand what dwell in the cyberspace is the notion of* ***construction****. I will introduce this eight modes in a natural order of their didactical development.*

trivial construction *An object may be identified by means of pointing the object itself. This is called a* trivial construction *(*trivialization*). If X is an object (of any type or it may be even a construction), then* ^0X *is a construction.* ^0X *constructs simply object X without any change. If X is a construction it gives us a possibility to* ***mention*** *it or to speak of it.*

Comments and examples: *The number 6 is trivially constructed by a construction* ^{06}*,* ^0Venus *trivially constructs the individual Venus. The trivial construction constructs its own object independently of any valuation (exact meaning you will see below). The main reason for this construction is that it enables us to distinguish between objects and their constructions while the transparent approach is adopted. Without this kind of construction misunderstandings can arise from the fact that constructing a complex object we need to use some simpler objects. This means the trivial construction helps us to switch between MENTION and USE modes, see definition of Service System in chapter 3.*

variables and valuation; atomic constructions *The second simplest constructions are* atomic constructions*. They are in fact variables for which we can substitute objects of a given type. Atomic construction constructs an object substituted for a variable in a certain valuation v. For every type T let* *a denumerable infinite number of variables be at our disposal. Each variable is distinct from any object of any type T. A variable is an incomplete construction which constructs a definite object of the respective type dependently on valuation. Valuation is a total function associating any variable of a type T with just one T-object. We say that variables v-construct respective objects, where v is a parameter for valuation. If a variable x v-constructs objects of a type T we say that x is ranging over T, denoting* $x :: T$.

The atomic constructions are open constructions, since variables are not bound in any way–we may arbitrarily substitute for them using valuations.

Comments and examples: *Variables are not letters, or characters, as standard logicians consider. Variables, being constructions, are language independent entities. The letters standardly used for variables (like x,y,* x_1, *...,* x_n, *...) are conceived to be names of variables. If v is such valuation which assigns to a N-variable y, where N is a type of natural numbers, the object (number) 6, then the atomic construction y v-constructs the N-object (a number) 6. If v is such valuation which associates an U-variable z, where U is a type of individuals, with the object (individual) George Bush Jr , then the atomic construction z v-constructs the U-object George Bush Jr.*

And now the atomic construction and trivial construction together: If x is a T-variable and v is any valuation, then ^0x *v-constructs the atomic construction x. Note, that what an atomic construction constructs depends on a given valuation, while what the trivial construction constructs is independent of a given valuation. Thus the trivialization bounds in some way variables being in its scope, see below. If X is constructed by* ^0X *then it means that X is identified without the use of any other construction. Pavel Materna gives in* ⁕ [STA]72:cite[MAT98]*[9] a very illustrative example: A child is able to identify circularity without exploiting other geometrical constructions like 'point', 'equidistant', etc. What the child „possesses“ is just the trivialization* ^0Circle*. The trivial construction binds the meanings (concepts) to the NL expressions very tightly. If, e.g., 'catastrophe' and 'disaster' (i.e. two distinct but synonymous NL expressions) denote one and the same object (a property of events), then we say that* ^{0}*catastrophe and* ^{0}*disaster are equivalent constructions. This claim contributes to the synonymous problem in NL. The importance of trivialization will be obvious as soon as a ramified hierarchy of types will be used.*

execution *Having a construction X, we may execute it (* execution construction*). By executing the construction X we construct this, what X constructs. If X is not a construction, executing X does not construct anything. Exactly: If X is an object of any type (it may be even a construction), then* ^1X *is a construction. If X is not a construction, then* ^1X *is v-improper construction, i.e. it constructs nothing for any valuation v. If X is a construction and v is any valuation, then* ^1X *v-constructs simply this object, which is v-constructed by X. Execution of a construction makes the effect of this construction. If X is a construction this brings a possibility to* ***use*** *it.*

Examples: *Let us have an object–a number 6. Then* ^{1}6*, i.e. the execution of this object, constructs nothing since „6“ is not a construction. But having a variable x the situation is different. The execution* ^1x *constructs this object, which is constructed by the atomic construction x dependently on chosen valuation v. The execution construction is something like instruction „do it!“. While other constructions describe a way by which step-by-step the by them identified object could be reached, execution says: do these steps now!*

double execution *If the construction X constructs another construction, then we can execute the constructed construction. This is called* double execution*. If X is an object of any type (it may be even a construction), then* ^2X *is a construction. If X is not a construction or X v-constructs an object, which is not a construction, then* ^2X *is v-improper construction for any valuation v. In other case* ^2X *v-constructs the object, which is v-constructed by the construction which is v-constructed by X.*

Comments and examples: *Double execution can be seen as a pattern of executing of operations which are stored in a database or in a kind of memory, maybe an artificial one, maybe a human one, in the same way as usual data or other images are stored.*

Double execution will help us to overcome possible difficulties with high order objects and their executions, cf. formal paragraph 5.

Let us take, e.g., a benefit for IBM in the year 2009 from the co-operation with FI. Let the benefit is given by concrete students working in this year on projects in Brno Global Service Delivery Center. Let the annual contract supplement is a specification of concrete students who will work for IBM in the given year. We can conceive this contract supplement as a construction *that when executed gives us the concrete students. Let us turn our attention to the general contract describing rules of how to formulate the annual contract supplements. Conceiving this general contract as construction that this when executed returns a concrete annual contract supplement for a given year; we can by double execution of this general contract construction obtain concrete students in the given year.*

What is what, and what provides what, is absolutely clear not dependently on our experience or other mysterious features of human world. All this could be „understandable" for a kind of memory-prediction machine that can be connected into a complex Service System.

sequence *If* $X_1, ..., X_n$ *are constructions then the construction* $(X_1, ..., X_n)$ *is a finite sequence (tuple) construction.*

Precisely: Let $X_1, ..., X_n$ *are constructions which v-construct elements of* $T_1, ..., T_n$*, respectively. Then* $(X_1, ..., X_n)$ *is a construction called sequence (or tuple) construction. If any of* $X_1, ..., X_n$ *is v-improper, the sequence construction is v-improper. Otherwise the sequence construction v-constructs the finite sequence of elements constructed by* $X_1, ..., X_n$ *(i.e. a member of Cartesian product of* $T_1, ..., T_n$*).*

projection *Contrarily, when X is a sequence construction then* $X_{(i)}$ *is a construction of its i-th projection.*

Precisely: Let X is a construction which v-constructs a sequence of a type $(T_1, ..., T_n)$*. Then for i = 1 .. n the* $X_{(i)}$ *is a construction called projection (or projection on the i-th component). If X is v-improper, then each of its projections is v-improper. Otherwise the projection v-constructs the i-th component of the sequence constructed by X.*

Examples and comments to both two construction modes will follow together with examples of the following modes of constructions.

abstraction *The construction of abstraction is used for creating functions. If* $x_1, ..., x_n$ *are variables ranging over types* $T_1, ..., T_n$*, respectively, and X is a construction constructing objects of the type T, then* $\lambda x_1...x_n(X)$ *(abstraction or closure, since it closes the possibility for a variable to be assigned a value by valuations) constructs a function of a type* $(T_1, ..., T_n) \rightarrow T$*.*

Precisely: Let $x_1, ..., x_n$ be distinct variables ranging over the sets $T_1, ..., T_n$, respectively. Let X be a construction that v-constructs members of a set T. Then $\lambda x_1...x_n(X)$ is a construction called abstraction. Abstraction v-constructs the following function F: Let $\langle t_1, ..., t_n\rangle$ be an n-tuple of the members of $T_1, ..., T_n$ respectively. Let valuation v' associate x_i with t_i (i=1 .. n) and be in other respects the same valuation as v, i.e. $v' = v(x_1 = t_1, ..., x_n = t_n)$. Then if X is v'-improper, F is undefined on $\langle t_1, ..., t_n\rangle$. Otherwise, the value of F on $\langle t_1, ..., t_n\rangle$ is the T-object v'-constructed by X.

Comments and examples: *Abstractions (or closures) produce functions. Functions as mappings are „flat", i.e. structureless. But their constructions are structured. They may be conceived as instructions how to create a function. The arguments of this function are the n-tuples (sequences of the length n) of objects v-constructed by x_1, ..., x_n, where v is appropriate valuation. The values of these function are given by X, the „body" of the functional construction. The values are T-objects v-constructed by the construction X.*

Let us denote the type of real numbers by $\Re$. Let valuation v associates $\Re$-variables x_1, x_2, x_3 with numbers 1, 2, 3, respectively. Let $[^0\!: x_1x_3]$ denotes the division of x_1 by x_3 (explanation of this denotation will come in the paragraph about application below). The abstraction

$\lambda x_1x_2([^0\!: x_1x_3])$

v-constructs the function (of a type $((\Re, \Re) \rightarrow \Re)$) that associates every pair $\langle i, j\rangle$ of numbers with the result of division of i by 3. Thus the function does not depend on the valuation of x_1 and x_2 as they are „bounded" by the λ-closure. As the variable x_3 is „free", the function depends on its valuation. The variable x_2 is not contained in the body of this construction. Thus the constructed function does not depend on the argument j.

More complex example: *Let us have constructions from the previous example:*

x_3,

$\lambda x_1x_2([^0\!: x_1x_3])$, *and*

$^0(\lambda x_1x_2([^0\!: x_1x_3]))$

The construction

$(^0(\lambda x_1x_2([^0\!: x_1x_3])), x_3, \lambda x_1x_2([^0\!: x_1x_3]))$

v-constructs (we assume the same valuation as in the previous example) a following sequence:

$(\lambda x_1 x_2([^0\!: x_1 x_3]), 3, \lambda x_1([^0\!: x_1 3]))$

application *The construction of application constructs objects as functional values by application of a function to given arguments. Arguments have to be of the same type as the function variables. If the function is not defined on the arguments given by the valuation v, or if one of the arguments is given by a v-improper construction*[13]*, the application is also v-improper.*

More precisely: Let X is a construction which v-constructs a function with values in a set T whose arguments are n-tuples of elements of $T_1, ..., T_n$*, i.e. an object of type* $((T_1, ..., T_n) \rightarrow T)$*. Let* $X_1, ..., X_n$ *are constructions which v-construct elements of* $T_1, ..., T_n$*, respectively. Then* $[XX_1...X_n]$ *is a construction called application. If the function v-constructed by X is not defined on arguments v-constructed by* $X_1, ..., X_n$*, the application is v-improper. If any of* $X_1, ..., X_n$ *is v-improper, the application is again v-improper. Otherwise the application v-constructs the value of the function v-constructed by X on arguments v-constructed by* $X_1, ..., X_n$*.*

Comments and examples: *The construction minus, i.e.* $^0-$*, constructs the standard arithmetic function 'subtraction' in a domain of real numbers. Let the type of real numbers be* $\Re$*. It is a function of a type* $((\Re, \Re) \rightarrow \Re)$*, i.e. it is one of* $((\Re, \Re) \rightarrow \Re)$*-objects. The application*

(c1).......... $[^0-{}^0 4 {}^0 2]$

constructs the number 2. Another $((\Re, \Re) \rightarrow \Re)$*-object is a function 'addition' constructed by a trivial construction* $^0+$*. The application*

(c2).......... $[^0+{}^0 1 {}^0 1]$

also constructs 2. The fact that both these constructions (c1) and (c2) are equal in the sense that they construct one and the same object, namely the number 2, can be expressed by using of a function =, so called identity of real numbers, being a $((\Re, \Re) \rightarrow Boolean)$*-object:*

(c3).......... $[^0\!= [^0-{}^0 4 {}^0 2][^0+{}^0 1 {}^0 1]]$.

It is obvious that (c1) and (c2) are distinct constructions. If we want to write down this fact that they are distinct constructions using our apparatus of constructions and we write

(c4).......... $[^0\neg[^0\!=^0 [^0-{}^0 4 {}^0 2]^0[^0+{}^0 1 {}^0 1]]]$,

we will come to problems. While the formula (c3) is correct, the formula (c4) is not. The type of the identity function = in (c4) is undetermined within

[13]such construction constructs nothing in the valuation v

our simple hierarchy of types (cf. definition in formal paragraph 1), as the constructions $^0[^0{-}^04^02]$ *and* $^0[^0{+}^01^01]$ *are not type-theoretically determined. To solve this problem we need the ramified hierarchy of types which covers constructions as a special kind of objects.*

An important comment has to be stressed: The part of cyberspace each of us (not excluding artificial agents) has in his/her/its brain or brain-like memory-prediction system is full of such constructions. Remember that any our action, regardless of who we are, comes as a result of a thinking step– a prediction. Thus, the above discussion and the following explications are legitimated.

Examples and comments for projections and sequences: *Let us have construction*

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))$.

Then the projection

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(1)}$

returns

$^0(\lambda x_1 x_2([^0\!: x_1 x_3]))$,

the projection

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(2)}$

returns

x_3,

and the projection

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(3)}$

returns

$\lambda x_1 x_2([^0\!: x_1 x_3])$.

In the valuation v as above the construction

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(1)}$

v-constructs

$^0(\lambda x_1 x_2([^0\!: x_1 x_3]))$,

as all variables here are o-bounded (cf. formal paragraph 3); the construction

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(2)}$

v-constructs the number 3, and the construction

$(^0(\lambda x_1 x_2([^0\!: x_1 x_3])), x_3, \lambda x_1 x_2([^0\!: x_1 x_3]))_{(3)}$

v-constructs

$\lambda x_1([^0\!: x_1 3])$.

Remember that in memory-prediction systems sequences play a crucial role. Any thought has a form of sequence, any perception comes as input in a form of sequence. Giving a name to sequence in the neocortical algorithm or in an other memory-prediction system means execution of the sequence construction. Focusing on a given item in a sequence means execution of the projection construction.

Thus, if wanted to construct artificial agent being able to simulate our thinking in a way we need to understand deeply what the constructions principles are.

Constructions not dependent on valuation, i.e., constructions which construct the same objects in any situations, are very important. Such construction has to have no such variable to which a real value can be assigned. Constructions of this kind are these identification procedures by using which we can capture objects (cf. the beginning of this section). Constructions could be seen as those patterns forming models in our brains or other memory-prediction systems.

And what do we do with such constructions within a process of performing a memory-prediction activities? The answer is simple. There are only two thing we can do:

- We can USE the constructions to identify appropriate objects by them, or
- we can MENTION the constructions to understand better how they identify the objects and to combine them into other constructions identifying another objects.

This MENTION–USE duality is a cornerstone of all „games“ in cyberspace. I will explain this specific game more deeply in next chapter–chapter 5.

As constructions are composed objects, not a flat ones, we need to identify what is a sub-construction of a given construction. Intuitively it is a compact part of a construction which itself is a construction.

Constructions could be equivalent. In this case these constructions either both construct the same object or both are improper (they construct nothing).

Formal Description 3 (MORE ABOUT CONSTRUCTIONS) *Let C be a construction and x a variable. Let C contains at least one occurrence of x. Then we define:*

(a) If C is x, then x is free in C.

(b) If C is ^0X, then x is o-bound in C.

(c) If C is $[XX_1...X_n]$, then x is free in C iff at least one occurrence of x is free in X or X_1 or ... or X_n.

(d) If C is $\lambda x_1...x_n(X)$, then x is free in C iff it is distinct from $x_1, ..., x_n$ and is free in X. The variables $x_1, ..., x_n$ are λ-bound in C iff they are not o-bound in X.

(e) If C is $(X_1, ..., X_n)$, then x is free in C iff at least one occurrence of x is free in X_1 or ... or X_n.

(f) If C is $X_{(i)}$, then x is free in C iff at least one occurrence of x is free in $X_{(i)}$.

(g) The variable x is free or o-bound or λ-bound in C only due to (a) - (f).

(h) A construction C is called a closed construction iff C does not contain any free variable. In other case it is called open construction.

Equivalency of constructions *The following definition is based on the very intuitive and wide adopted principle of extensionality. Moreover, computing principles for constructions are added. These principles say what part of a construction can be rewritten without any effect to the constructed objects.*

(1) A construction C_1 is equivalent to a construction C_2 iff for every valuation v holds: C_1 v-constructs the same object as does C_2 or both of them are v-improper.

(2) Let C_1, C_2 be closed constructions. Let C_2 arise from C_1 by correctly (i.e. without any collision in variables names) renaming of all λ-bound variables. Then C_1 is α-equivalent to C_2.

(3) Let C be an open construction that constructs T-objects. Let $x_1, ..., x_n$ be its free variables, $x_i :: T_i$ for $i = 1 .. n$. Let $B_1, ..., B_n$ be constructions of T_1-, ..., T_n-objects, respectively. Let

$C(x_1, ..., x_n := B_1, ..., B_n)$

denotes such operation, that substitutes each occurrence of x_i in C by the construction B_i and is admissible only if these substitutions do not lead to collision in the sense that a free variable in B_i after this substitution starts to be a λ-bound variable in B_i. Then the construction

$C(x_1, ..., x_n := B_1, ..., B_n)$

and

$[(\lambda x_1...x_n(C))B_1...B_n]$

are β-equivalent. This is usually denoted by the formula

$[(\lambda x_1...x_n(C))B_1...B_n] = C(x_1, ..., x_n := B_1, ..., B_n)$.

This is the fundamental computing rule in the lambda-calculus.

(4) Let C be a construction. Let $x_1, ..., x_n$ be any variables, $x_i :: T_i$ for $i = 1 .. n$. Then the construction

$[(\lambda x_1...x_n(C))x_1...x_n]$

and the construction C are η-equivalent. This is usually denoted by the formula

$[(\lambda x_1...x_n(C))x_1...x_n] = C$.

It is known that each two α-equivalent constructions are equivalent, each two β-equivalent constructions are equivalent, and each two η-equivalent constructions are equivalent.

The second claim follows from the fact that for each valuation v the construction
$C(x_1, ..., x_n := B_1, ..., B_n)$
v-constructs the same T-object as
$[(\lambda x_1...x_n(C))B_1...B_n]$
does, or both are v-improper. The third claim follows using the β-conversion
$C = C(x_1, ..., x_n := x_1, ..., x_n) = [(\lambda x_1...x_n(C))x_1...x_n]$
and the fact that if a variable x_i is not free in C the rewriting $x_i := x_i$ makes nothing.

Subconstructions. *Let C be a construction.*

1. *C is a subconstruction of C*

2. *Let C be* ^0X *or* ^1X *or* ^2X*. If X is a construction, then X is a subconstruction of C.*

3. *Let C be* $[XX_1...X_m]$*. Then X,* X_1*, ...,* X_m *are subconstructions of C.*

4. *Let C be* $\lambda x_1...x_m(X)$*. Then X is a subconstruction of C.*

5. *Let C be* $(X_1, ..., X_m)$*. Then* X_1*, ...,* X_m *are subconstructions of C.*

6. *Let C be* $X_{(i)}$ *for i = 1 .. n. Then X is a subconstruction of C.*

7. *If A is a subconstruction of B, and B is a subconstruction of C, then A is a subconstruction of C.*

8. *Anything is a subconstruction of C only due to 1. - 7.*

Note, that $X_{(i)} = X_i$ *for any sequence construction X.*

Important mathematical and logical constructions. *Important constructions for modeling purposes are some constructions of mathematical and logical (i.e. analytical) functions. Note that these functions are simply objects. The arithmetic operations are functions of a type*

$((Time, Time) \rightarrow Time)$

and we will use their trivial constructions using the traditional symbols +, -, ×, ÷, i.e. we will use their constructions $^0+$, $^0-$, $^0\times$, $^0\div$*, respectively.*

For the logical connectives being of type $((Boolean, Boolean) \rightarrow Boolean)$ *(or* $(Boolean \rightarrow Boolean)$ *for negation) we will use the same approach, as well as for less then and equal tests over numbers, which are of a type* $((Time, Time) \rightarrow Boolean)$*. Thus we have constructions* $^0\wedge$, $^0\vee$, $^0\neg$, $^0<$*, and* $^0=$ *of conjunction, disjunction, negation, less then, and equal, respectively.*

The identity function is constructed simply by abstraction for each type T. If x::T for any T, then the identity of type T is constructed by

$\lambda x(x)$.

Note that identity function is nothing trivial or simple. The construction is quite simple, but its meaning not. Without abstraction it is impossible to understand what identity is. Practice brings lot of experiences with non easy judging whether two focused objects are one and the same or not. The problem very often is: we watch a phenomenon, than we watch another phenomenon, and than after some mental work is done we recognize these two watched *phenomenons were one and the same object. An issue could be a little bit philosophical question: what means that two watched phenomenons are one and the same item? My opinion is that 80 percent of our mental work is spent in solving identity issues (but it is a hypothesis only).*

The constant function for each type T is constructed by
$\lambda x({}^0 e)$,
where $e \in T$.

The often used general and existential quantifiers and singularizer are again objects defined for each type as follows:
Let x::T be a variable, C a construction of Boolean-objects. Then

- *objects* $\bigwedge^T / ((T \to Boolean) \to Boolean)$ *are general quantifiers. The construction* $[{}^0\bigwedge^T \lambda x(C)]$ *constructs True if* $\lambda x(C)$ *constructs the class of all members of T, otherwise it constructs False. Instead of* $[{}^0\bigwedge^T \lambda x(C)]$ *we will write* $\forall x(C)$.
- *objects* $\bigvee^T / ((T \to Boolean) \to Boolean)$ *are existential quantifiers. The construction* $[{}^0\bigvee^T \lambda x(C)]$ *constructs True if* $\lambda x(C)$ *constructs the class containing at least one member of T, otherwise it constructs False. Instead of* $[{}^0\bigvee^T \lambda x(C)]$ *we will write* $\exists x(C)$.
- *objects* $I^T / ((T \to Boolean) \to T)$ *are singularizers. The construction* $[{}^0 I^T \lambda x(C)]$ *constructs the only member of* $\lambda x(C)$ *if* $\lambda x(C)$ *constructs a T-singleton (the class containing just one member of T), otherwise it is improper. Instead of* $[{}^0 I^T \lambda x(C)]$ *we will write* $\iota x(C)$ *and read it „the only x that C“.*

Note that all these mathematical and logical objects are defined for each type T separately. This is the tax which is paid for the pigeon hole order brought by type theory approach. And note that it is impossible to make, e.g., equal tests for objects of distinct types. This is not surprising: nobody will test whether a pencil is the same thing as a number. Maybe you are afraid this bound us too much. Will it be enough rich apparatus to deal with all the world complexity? Don't worry, please. Almost every comparisons and tests of equality we do in our heads are tests or comparisons of constructions, not of objects. More about this you will find in the formal paragraph 5 where constructions will be introduced into the world of objects.

The β-rule is the fundamental computing rule in the lambda-calculus. ⋇ [STA]$_{73}$:cite Barengret?: Lambda Calculus[?] I would like to note that according to my opinion the β-rule or β-conversion is something very closed to firing cells in columns of neocortex as we spoke on in the previous section. I think it is a formal expression of the auto-associative mechanism our brains use when recall anything from the memory. Thus, the λ-abstraction, the β-rule, the wrapping of items into a sequence, and the focusing one item of a sequence by projection, are patterns of operations the software memory-prediction systems be able to perform to be equal participants in Service Systems.

One comment I would like to make to the identity function. As discussed above it is almost the one of most often used functions in thinking and cognition processes. We know we have separate identity functions for each type T. For all of them we will use one and the same symbol: the „=“. But we will be very careful not to mix types on both sides of this symbol. Moreover, when using the identity function, denoted by „=“, in a construction the correct expression is „$^0 =$“, not the „=“ only. But for the convenience we will omit the superindex zero as using it leads to non transparent expressions. We will do this for the identity function only. This will be a honor of its exceptionality. But all the time we must have in mind what is the right meaning of the expression within a construction and we must be aware we violate a rule.

Now we apply the general constructions to real world understanding. When trying to understand something, very often we use to derive this subject from a known things or presumptions. This process of derivation can continue backward: to understand A, we derive it from B and C; to understand B or C, we derive it form B1, B2, B3, or from C1 and C2, respectively. And so on. Where this process stops? Will it stop anytime or not? To fix our understanding we need such process to be a finite one. Thus there must be some base, from which this understanding process starts. And for all of us the same one. The „for all of us“ means for all Service Systems which will be connected into an intelligent game.

Thus, a so called epistemic base will be introduced and notions of extension and intension explicated. The epistemic base is chosen such that we can analyze any sentence of our natural language over it and we can find the meaning of this sentence. The epistemic base is formed by

- a set of all individuals (or items) we are able to distinguish from anything other,
- by a time,

- by two truth values „true“ and „false“, and
- by possible state-of-affairs. These state-of-affairs help us to understand what semantics of a message or a sentence is.

A precise description of the epistemic base is done in the following formal paragraph. The Leibnitz's notion of possible world in Tichy's modification is used in the following formal explanation. In this approach the world is not conceived as a collection of all things, but as a collection of all facts. More details you can find in Pavel Materna's book „Concepts and Objects“ ⋇ [STA]$_{74}$:cite Materna98 [9].

Formal Description 4 (EPISTEMIC BASE AND THE REASONING) *The theory of concepts and natural language analysis is carried out over the so-called epistemic base (EB).*

The epistemic base is a base EB = $\{$*Universe, Boolean, Time, PossibleWorld*$\}$ *consisting in four elementary types.*

Type Boolean is a set of truth values $\{True, False\}$*, type Time is a set of time moments (or real numbers).*

Type Universe (precisely Universe of Discourse) is a set of individuals (maybe better items), i.e. of material entities possessing no non-trivial property necessarily. Material entity is something we can focus on, we can distinguish from anything else, and whose identity we understand a priori. The set Universe is shared by all possible worlds. There are no „possible items“. Items (individuals) are carriers of properties/features/traits. Each such item is diverse from any of these properties/features/traits. One item (individual) cannot transmute into another item. It can only take over various features/traits in various states of the world.

Type PossibleWorld is a set of possible worlds, i.e., a set of logically consistent states-of-affairs relative to a given (natural) language.

To understand the possible world better, let us go through several steps:

- *First, let us assume that there is a collection of (intuitively and pre-theoretically) given traits assigned to objects (dispersed on those objects) of our interest.*
- *The choice of the objects of our interest and of the basic traits depends on the domain we want to investigate.*

- *In any case each such object and each such trait is expressible in language we use for communication.*
- *As we want to deal with Service Systems and the domain of their operation can unpredictably change, let us suppose we take into account all objects, i.e., the whole set Universe, and all conceivable and at the same time in language expressible traits.*
- *The objects and traits are empirical. Hence the distribution of the traits among the objects is unpredictable and we have to apply an empirical procedure to find out which of the logically possible distributions is the actual one.*
- *But these possible distributions can change in time. Hence possible world is better to understand as the chronology of logically possible distributions of basic traits among objects.*
- *Finally, the set PossibleWorld can be viewed as a logical space of the (natural) language we are using to communicate.*

A chronology of real distributions of basic traits among objects is called actual world. Which of the above mentioned possible worlds is the actual one is impossible to compute. The only possibility is to discover empirically a part of the actual world. Note, that to know what of all possible worlds is the actual one means to be omniscient.

The couple $\langle w, t\rangle \in$ *(PossibleWorld, Time) represents a particular state of the world.*

Assertion, that a given piece of knowledge or information is true, means that the actual world is among possible worlds in which this piece of knowledge/information is valid.

Pieces of knowledge or information we convey by using sentences/propositions, i.e. ((PossibleWorld, Time)$\rightarrow$ *Boolean)-objects. Each proposition sets apart a subset of the set PossibleWorld such that this proposition is true in each possible world from this subset.*

Any set of propositions being really true in a given time moment sets apart a subset of the set PossibleWorld such that this subset definitely contains the actual world.

Let T, T' are types over EB. A T-object, where $T \neq$ *(PossibleWorld* $\rightarrow T'$*) for any T', is an extension. Each (PossibleWorld* $\rightarrow T$*)-object is an intension.*

Abbreviation: If X is a ((PossibleWorld, Time)→ T)-object or equivalently (PossibleWorld → (Time → T))-object, and w, t are PossibleWorld-, Time-variables, respectively, we will write

$^0X_{wt}$

instead of

$[^0X(w,t)]$ *or* $[[^0Xw]t]$.

Generic examples: *will be done by several couples of a type extension and to it corresponding intension.*

The extension ***class of T-objects*** *of the type* $(T \rightarrow Boolean)$ *could be associate to* ***property of a T-object*** *of the type ((PossibleWorld, Time)→* $(T \rightarrow Boolean)$*).*

The extension ***n-ary relation in-extension*** *of the type* $((T_1, ..., T_n) \rightarrow$ *Boolean) could be associate to* ***n-ary relation in-intension*** *of the type ((PossibleWorld, Time)→* $((T_1, ..., T_n) \rightarrow Boolean)$*).*

The extension ***Boolean value*** $\{True, False\}$ *of the type Boolean could be associate to* ***proposition*** *of the type ((PossibleWorld, Time)→ Boolean).*

The extension ***analytical function*** *of the type* $((T_1, ..., T_n) \rightarrow T)$ *could be associate to* ***empirical function*** *of the type ((PossibleWorld, Time)→* $((T_1, ..., T_n) \rightarrow T)$*).*

Information in Rudolph Carnap sense. *Let P be a set of propositions. Let* $R \subseteq Time$ *is arbitrary time interval. Let w ranges over PossibleWorld, t ranges over Time,* $\equiv$*/((Boolean, Boolean) → Boolean) is the identity function on boolean objects. Then*

$W_R^P = \lambda w\ (\ ^0q_{wt}\ ^0 \equiv\ ^0True \wedge q \in P \wedge t \in R)$,

i.e. the set

$\{w \in PossibleWorld \mid \forall q \in P\ \forall t \in R\ (^0q_{wt}\ ^0 \equiv\ ^0True)\}$

(using a common math and logic notation to simplify the constructions) is called the admissible logical space of the set of propositions P with respect to the time interval R. The smaller the W_R^P *is, the more information is contained in the set P.*

If W_R^P *is identical with PossibleWorld then the set P of propositions gives no information. If* W_R^P *is not identical with PossibleWorld then* W_R^P *is a proper subset of the set PossibleWorld, and the set P of propositions gives information in the time interval R.*

If $R = Time$, we speak of an admissible logical space of the set of propositions P.

Distinguishing between extensions–objects not depending on state-of-affairs, and intensions–objects depending on state-of-affairs is very important to understand any domain investigated. Intensions, i.e. functions assigning to states of the world objects of the real world, represent simply empirical facts which could become aware using an empirical procedure only, or in other words, using a pragmatically chosen data collecting or using a kind of perception.

So to evaluate an intension function requires two steps:

- the first one is to collect empirical data and, e.g., to store them into database tables or into a kind of memory; this must be done somewhere before the need of using this data appears,
- the second one is evaluating the intension function using those collected data, i.e. those database tables or those memory content.

It is obvious that value of such intention function changes according to performed database tables or memory actualizations.

The evaluation of extension function means simply to compute an appropriate algorithm. Result of such a function depends on arguments, only, regardless of the state of the world.

According to Carnap, an information is given to this extent to which a proposition, when true, excludes some possible worlds–those one in which this proposition is false.

Now this definition of information, which is not relative with respect to the receiver so as a Shannon one is, starts to be comprehensible. It will serve as an absolute measure of information contained in a proposition. This absolute approach to information we need to

- analyze and/or
- formulate, maintain and manipulate, and of course
- use

the value propositions defining benefits for clients in Service Systems.

First order objects are T-objects, where T is a simple type (a type from the simple hierarchy of types). Thus no construction is a first order object. To be able to focus on constructions as on other objects we need to extent our notion of types to the so called ramified hierarchy of types.

A motivation for to do it is on hand: every value proposition, every contract, every plan within business are full of sentences expressing constructions and handling with these constructions. Lot of work is wasted due to misunderstandings arising from not well and not shared comprehension of what was stated by these propositions and what was intended by authors of these propositions. So, to focus on constructions as on other objects is of great importance, namely in service paradigm driven situations.[14]

Ramified hierarchy of types is obtained by the following procedure: let's take all constructions constructing first order objects. This collection forms a new type. Denoting this collection $Cons^1$ we say that each its member is an object of order 2. Let's collecting all constructions of objects of order 2. We denote this collection $Cons^2$. The members of collection $Cons^2$ are called objects of order 3. This way we can continue with no limitations. Ramified hierarchy means an introduction of constructions into the world of objects. Till now we spoke about objects of the material world or better of the time-space. Now we extend our space by the whole cyberspace. Constructions are inhabitants of cyberspace.

Be aware that speaking of conditions of the future service provision we often mention what will be mentioned if only something is mentioned .... Now we can see that such formulations simply express constructions of constructions, i.e. denote objects of higher order.

Two points I'd like to stress: first, logical point of view can help us to make negotiations more clear and more unambiguous. Second, when combining our human capabilities with capabilities of some artificial agents everything which could be mathematized will help us being mathematized; artificial agent will be able to work with this.

An issue of such importance deserves an exact formal definition. Don't worry if it is not clear for the firs reading. This is abstract and non easy definition.

Formal Description 5 (RAMIFIED HIERARCHY) *Definition of ramified hierarchy of types has three parts. First, types of order 1 are defined*

[14]Note, please, that these musings are very closed to the neocortex regions hierarchy explicated in the previous section.

(paragraph denoted by T_1), second, constructions of order n are defined (paragraph denoted by C_n), third, types of order n+1 are defined (paragraph denoted by T_{n+1}). The definition is an inductive one:

T_1 : —

Simple types over the epistemic base EB = {Boolean, Universe, Time, PossibleWorld} are types of order 1.

C_n : —

Let α be a type of order n. Then:

1. *Any variable that v-constructs α-objects is a construction of order n.*
2. *If X is an α-object then ^0X or ^1X or ^2X is a construction of order n.*
3. *Let X, X_1, ..., X_m be constructions of order n. Then $[XX_1...X_m]$ is a construction of order n.*
4. *Let $x_1, ..., x_m, X$ be constructions of order n, x_i being variables. Then $\lambda x_1...x_m(X)$ is a construction of order n.*
5. *Let $X_1, ..., X_m$ be constructions of order n. Then $(X_1, ..., X_m)$ is a construction of order n.*
6. *Let X be a sequence construction of order n. Then, for each i = 1 .. m, $X_{(i)}$ is a construction of order n.*

T_{n+1} : —

Let $Cons^n$ be the collection of all constructions of order n. Then:

I. *$Cons^n$ and every type of order n is a type of order n+1.*

II. *Let $\alpha, \beta_1, ..., \beta_m$ be types of order n+1. Then $((\beta_1, ..., \beta_m) \rightarrow \alpha)$, i.e. the collection of all partial functions from $\beta_1 \times ... \times \beta_m$ into α, is a type of order n+1.*

III. *Let $\beta_1, ..., \beta_m$ be types of order n+1. Then the cartesian product of $\beta_1, ..., \beta_m$, denoted by $(\beta_1, ..., \beta_m)$, is a type of order n+1.*

IV. *Nothing is a type of order n+1 unless it is determined by I. - III.*

An object A of type α, denoted A/α , where α is a type of order n, is called an object of order n.

If x is a variable, i.e., an atomic construction, which constructs objects of a type α, we denote this situation by $x :: \alpha$ and we say that variable x ranges over type α. Let's understand that a variable x constructing objects of a type α of an order n is, conceived as an object, an object of type $Cons^n$, i.e. an object of order n+1. Thus, the two terms

$x::\alpha$ and x/α

*say a very different things. The first one tell us a type of objects we obtain by **using** this object as a construction, the second one tell us a type of this thing when this atomic construction is conceived as an object, i.e. when it is **mentioned**. Of course in this case the type α must be some $Cons^n$, otherwise the second term is a nonsense.*

Again we are seeing the MENTION-USE fundamental game in the cyberspace. Note that in the cyberspace we have constructions only and nothing more than constructions. This could be taken as a precise definition of what the cyberspace is.

Summarizing the formal definition of ramified type hierarchy we can say: the collection of all constructions of order n forms one type of order n+1. The formulation „every type of order n is a type of order n+1" in the formal paragraph T_{n+1}(point I.) makes it possible to include lower order types into the higher order types. This enables to ascribe types to complex constructions, the components of which are constructions of distinct orders. Higher order objects are α-objects, where the order of α is greater than 1.

The intuition behind the ramified type definition is: therein „near the infinity" we cannot recognize objects nor their types. The objects look like „all-order objects" for us. Going back from the very far not good observable regions to a more familiar regions we are accustomed to live in, we start step-by-step to recognize what objects around us are. And we could see that objects around us step-by-step loose their multi-type property. Some of these objects stop on some level and we know that an object is, e.g., a construction of construction of construction of an object–i.e. a construction of third order. Or we could see an object which is construction of construction of an object–i.e. a construction of second order, or another object which is a construction of an object–i.e. construction of first order. At the end of this process we will find objects that are merely objects and not constructions, i.e. objects of first order. These last ones are objects of the real time-space. All the others are objects of the cyberspace, only.

How we cope with real time-space objects in our memory-prediction systems? Simply by using names as labels on these objects. The names help us to execute trivial constructions of these objects. We point these objects in a way by using their names. In section 4.2 the role of names were stressed and here I want to argue that this approach and the one introduced here are again very tightly connected.

Let's take another examples. If the valuation v associates a variable x with an object, which is the construction of data model of the Bank EXPANDIA, then the atomic construction x v-constructs just the object being the construction of data model of Bank EXPANDIA.

Notice that neither constructions, nor functions whose values or arguments contain constructions are first order objects. Objects of types Boolean, Universe, Time, PossibleWorld, forming the epistemic base, are of first order. Extensions and intensions that were mentioned before are first order objects.

When creating database scheme for an usual information system (e.g. system for personal evidence, financial system, ERP system, etc.) we usually last with first order objects. But creating model for a metasystem that serves for example to maintain a datawarehouse, we need to work not only with classes, properties, data functions, etc. of the first order objects but with constructions of those objects too. For example we need to compare two database schemes (of primary systems) to find out which part of which scheme will be used in the integrated scheme of the datawarehouse. Each of those schemes can be conceived to be a kind of construction. Our metasystem then will contain not only entities being containers for usual objects, but some other entities, that have to serve as containers for schemes of primary systems, i.e. for higher order objects.

This again resembles the cortical hierarchy of regions, cf. fig. 4.2. High order objects introduced here represent well the subregions of cortex being involved in the cognitive process running in our brains.

Maybe you can ask why put I such considerations and such abstract theory into the book about Service Systems. My answer is: the reason is the problem of a correct, valid, mutually understand and plausible value proposition formulation. I do not want to say that the future service oriented business will be full of lambda terms and inductive definitions twice recursively rolled up. Most of people, and business people, too, in the coming service economy will live successfully with no idea about high order objects and ramified type hierarchy. But when facing the problem of a balance of two value propositions in a pair of co-operating Service Systems, cf. def. 4, they are leaved to their intuition only. To be better outfitted this theory is useful; I personally lived out many times the advantage of „knowing why“ within business negotiations. And I must say the knowledge of what we are speaking of is beyond price.

And, moreover, there is another reason. This logical foundations of the hierarchy of higher order objects, the inhabitants of our minds, thus the inhabitants of cyberspace, form the base for the so called Diamond Road, see chapter 5, which leads to the Reference Model of Service System.

Let's turn back to our logical point of view. Often we are hearing sentences like: „this is logical that ...“, or „... logically it must be ...“, etc. Often the speakers don't know the exact meaning of a logical consequence. As this notion is crucial for value proposition formulation, value proposition comparison, and value proposition evaluation, I devote to this issue one formal paragraph here. First we address a problem of what can be derived from a given set of propositions.

A next issue have to be known to all lawyers and all managers if they want to write a good agreement or a well done value proposition within a Service System. Each such document is a sequence of propositions. As such it gives a specific amount of information. A legitimate questions are:

- what are all consequences of this document (or contract)?
- having two documents (proposals or contracts) describing a value proposition of one and the same service system, which of them gives less information, or
- are these two documents comparable? have these two documents (contracts) the same consequences?
- what parts could be exchanged between these documents without changing the overall sense?

I am afraid there is a lot of lawyers and/or managers who don't know to answer such questions well. In service economy this problem will be multiplied compared to the goods economy epoch.

In section 3.3 of the previous chapter on service systems we were discussed a problem of value proposition formulation and value propositions comparing to decide whether two service systems are co-operating or not. It is hard to solve these issues without knowledge of a kind of information ordering. We need to compare two or more sets of propositions and determine which of them gives us more information or which two are not comparable.

One of the most important theoretical results influencing managerial methods and co-operating strategies between business subjects (especially in service provision environment) is the so called „reversed triangle inequality“ which holds for measuring of information capacity of documents or reports. It says, simply, that if we take together several reports from isolated parts of the whole process (done by operational management), then derive from each such report all its logical consequences (done by middle management), and then we make a union of all this information into a final report (done for top management), we obtain as a result less information then if we had taken together all the reports of operational managers, made a union of them, and after this derived all consequences and formed the final report.

This topic I will deal with more deeply in chapter 5 when speaking on the Diamond of agents team organization, namely on the functioning of such organization. I think the reversed triangle inequality for information capacity is one of crucial mathematical supports of the reference model of Service System which I present in sect. 5.4 named Diamond of Predictive Behavior.

I would like to cause you understand there is mathematics behind the management process, not your volition only. You are free, you can decide whether you will go the way of value added in GTM (see fig. 1.1) or you can decide to turn to zero-sum games, but you cannot try to hide yourself behind a big complexity of information.

Formal Description 6 (INFORMATION ORDERING) *To order sets of propositions according to what information they provide the distinction between logical implication and material implication has to be emphasized.*

Logical implication (logical following). *In propositional calculus there is material implication defined as a function (operation) of a type*

((Boolean, Boolean) $\rightarrow$ Boolean)

such that

$x \Rightarrow y$

gives False, if x is True and y is False, otherwise it gives True.

A proposition q logically follows from a proposition p, denoted

$p \supset q$*,*

if

$\forall w \in PossibleWorld\ \forall t \in Time((^0p_{wt}\ ^0\equiv\ ^0True) \Rightarrow (^0q_{wt}\ ^0\equiv\ ^0True))$.

If $^0p_{wt}\ ^0\equiv\ ^0False$ *or* $^0p_{wt}$ *is improper we can say nothing about truth-value of q in the state of world (w,t).*

Quasi-ordering on sets of propositions. *Let P, Q be sets of propositions such that for each* $q \in Q$ *holds either* $q \in P$ *or there exists* $p \in P$ *such that* $p \supset q$. *Then, according to the definition of the logical following and the Carnap's definition of information (cf. above), it holds*

$W_R^P \subseteq W_R^Q$

for each time interval $R \subseteq Time$. *We say in such a case that the set Q of propositions gives less or at least the same information as the set P of propositions, and denote it* $Q \lessdot P$. *As the relation* $\lessdot$ *is homomorphic with the set theoretical inclusion on subsets* W_R^P *of the set PossibleWorld, it is reflexive and transitive.*

Let $P = Q \cup \{p\}$, *where* $p\neg \in Q \wedge \exists q \in Q(q \supset p)$. *Evidently* $P \lessdot Q$ *and* $Q \lessdot P$. *But* $P \neq Q$. *Thus the relation is not antisymmetric.*

The relation $\lessdot$ *on the sets of propositions, i.e. on the power set of the set of all propositions, is quasi-ordering. To derive from this relation an ordering relation (exactly a partial ordering) we have to introduce a function consequence, denoted Cn.*

Logical consequence. *Let us denote the type of a proposition by Proposition. Thus,*

Proposition = ((PossibleWorld, Time) → Boolean).

Let p, q be variables, p::(Proposition→Boolean), q::(Proposition→Boolean). A function Cn of the type $(Proposition \rightarrow Boolean) \rightarrow (Proposition \rightarrow Boolean)$ *is called a consequence (or a logical consequence) iff*

$Cn = {}^1(\lambda p(^0\bigcup \lambda q(q \lessdot p)))$.

To see how it works, let's take a set of propositions P/(Proposition→Boolean). The application of Cn yields:

$[^0Cn\,^0P] = [(\lambda p(^0\bigcup \lambda q(q\ ^0\lessdot p)))\ ^0P]$

$= {}^0\bigcup \lambda q(q\ ^0\lessdot\ ^0P)$.

Reading this construction in human words we have: „A logical consequence of a set P of propositions is an union of all sets of propositions which give less or at least the same information as the set of propositions P". The function Cn yields on a set P of propositions a set of all its possible logical consequences.

The Cn function has several interesting and important properties.

***(1) The operation (function) Cn is idempotent one**, i.e.*
$[{}^0Cn[{}^0Cn^0P]] = [{}^0Cn^0P]$.

Thus, it can be applied with some effect only once. The second application makes nothing.

Proof: $[{}^0Cn[{}^0CnP]] =$
(substitution for the inner application)
$= [{}^0Cn({}^0\bigcup \lambda q(q^0 \lessdot^0 P))]$
(substitution for the Cn)
$= [(\lambda p({}^0\bigcup \lambda q(q^0 \lessdot p)))\ ({}^0\bigcup \lambda q(q\ {}^0\!\lessdot\ {}^0P))]$
(alpha-reduction)
$= [(\lambda x({}^0\bigcup \lambda y(y^0 \lessdot x)))\ ({}^0\bigcup \lambda q(q\ {}^0\!\lessdot\ {}^0P))]$
(beta-reduction)
$= ({}^0\bigcup \lambda y(y^0\!\lessdot\ ({}^0\bigcup \lambda q(q\ {}^0\!\lessdot\ {}^0P)))))$
(the outer union applied on λy abstraction leaves the result of the inner construction unchanged, as when in the inner construction we derive everything what derived could be, we cannot to add something new to it in the outer construction)
$= {}^0\bigcup \lambda q(q\ {}^0\!\lessdot\ {}^0P)) = [{}^0Cn^0P]$.

(2) Cn operation is connected with the $\lessdot$ relation by the formula

$\forall P, Q \in (Proposition \rightarrow Boolean)(({}^0Q^0\!\lessdot^0P)^0 \Leftrightarrow ([{}^0Cn^0Q]^0 \subseteq [{}^0Cn^0P]))$
Proof: *The formula $\forall P, Q \in (Proposition \rightarrow Boolean)((Q\lessdot P) \Rightarrow ([CnQ] \subseteq [CnP]))$ follows from the Cn definition. The reversed formula could be easily proven by contradiction, and I leave it to you as an exercise.*

***(3) The set** $\Pi = \{[CnP] \mid P \in$ **(Proposition** $\rightarrow Boolean)\}$,*

*i.e. the set of all with respect to the consequence closed sets of propositions, **can be partially ordered by a relation induced by the $\lessdot$ relation.***

According to the point (2) the induced relation is simply the set-theoretical inclusion. And we know form algebra that a structure $(\Pi, \subseteq)$ is partially ordered. Moreover, we know that this structure forms a complete lattice as the set-theoretical inclusion is a natural ordering.

(4) Following formula is valid
$\forall P \in (Proposition \rightarrow Boolean)(P \subseteq^1 [{}^0Cn^0P])$

Information capacity of a set of propositions. *Let*

P / (((PossibleWorld, Time) $\rightarrow$ Boolean) $\rightarrow$ Boolean)
be a set of propositions. Then
$^1[^0Cn^0P]$
represents its information capacity. We say that P generates this information capacity.

Each information capacity has a denumerable infinity count of generators.

The information capacities of sets of propositions form a complete lattice with respect to the amount of information embedded in these proposition sets.[15]

The zero-element of this lattice is generated by any set of tautologies. The one-element of this lattice is generated by any set of propositions which contains a contradiction.

For any two sets of propositions P, Q there exists their lattice join
$P \curlyvee Q$
which is the smallest set of propositions with the property
$(P \lessdot (P \curlyvee Q)) \wedge (Q \lessdot (P \curlyvee Q))$.
For any finite set of proposition sets there exists its lattice join.
For any two sets of propositions P, Q there exists their lattice meet
$P \curlywedge Q$
which is the largest set of propositions with the property
$((P \curlywedge Q) \lessdot P) \wedge ((P \curlywedge Q) \lessdot Q)$.
For any finite set of proposition sets there exists its lattice meet.

These algebraic properties of sets of propositions have a great impact to business area. A consequence of the properties of information capacity of a set of propositions is: any situation can be negotiated unless the parties play a zero-sum game in GTM.[16]

Thus, all results from algebra concerning complete lattices are applicable.

(Let's do it, dear managers, entrepreneurs, and lawyers. I mean let's apply algebraic and logical approach. Our co-operations, value proposition formulations, contracts, co-creation evaluations, etc. would be more accurate! Instructions how to do it you can find in chapter 5 and when and why to do it you will find in chapter 6).

Reversed triangle inequality for information capacities. *Let*

---

[15] The proof of this claim you can find in several articles, namely [5, 2, 4]
[16] More about this will be in chapter 6

$P_i/(Proposition \to Boolean)$ for $i = 1 .. n$, i.e. each P_i is a set of propositions. Then the following inequality holds:

$\bigcup_{i=1}^{n} {}^1[{}^0Cn^0 P_i] \subseteq^1 [{}^0Cn^0(\bigcup_{i=1}^{n} P_i)]$.

Proof: *Let q be a proposition, $q \in \bigcup_{i=1}^{n} {}^1[{}^0Cn^0 P_i]$. Thus there exists at least one P_i such that $q \in^1 [{}^0Cn^0 P_i]$. Two possibilities exists: either $q \in P_i$ or $q\neg \in P_i$. In the first case the proof is done as if $q \in P_i$ then $q \in \bigcup_{i=1}^{n} P_i$ and, thus, $q \in^1 [{}^0Cn^0(\bigcup_{i=1}^{n} P_i)]$. The second case means $\exists p \in P_i$ such that $p \supset q$. This we can reformulate as $\exists p \in \bigcup_{i=1}^{n} P_i$ such that $p \supset q$. Hence, $q \in^1 [{}^0Cn^0(\bigcup_{i=1}^{n} P_i)]$, which finishes the proof.*

This reversed triangle inequality tells us what we intuitively know, but very often use to forget. This is the fact that when first generate information capacities of particular messages (sets of propositions) and then make their union, we lose connections going through particular messages boundaries. When operating in reverse order–i.e. first make a union of all messages and then generate information capacity, we have all what is possible to have.

The last comment to this formal paragraph is: let's be aware that „leadership from the front“[17], method of forming seminars and forming in large extent (FILE)[18] and network organizational structure (NOS)[19] are justified. They are an organizational response to the above explained theoretical principles.

The most important inhabitants of the cyberspace are concepts. Concepts are identification procedures comprised of certain mental steps that lead to the identification of objects we focus on. This was stated at the beginning of this section together with the Expression-Object-Concept triangle diagram. Let's focus on figure 4.3 once again. Understanding what construction is, and understanding the epistemic base, now, we can interpret this picture better and obtain much more relevant visualization.

Pavel Materna in his book [⁕ [STA]75:cite MAT98] [9] starts with a definition of the so called concept* and continues to the notion of concept. Simplifying this approach (similarly as Marie Duzi does in [⁕ [STA]76:cite DUZI96] [3] and [⁕ [STA]77:cite DUZI99a])[?] we will define concepts as Materna's concepts* as it is sufficient for our purpose. The difference consists only in that two concepts identifying one and the same object in all states of affairs can be taken as one concept in Materna's approach. Than, simply saying, concepts are the

[17] see the last paragraph in subsect. 5.3.2 of sect. 5.3

[18] see paragraph 5.3.2 in subsect 5.3.2 of sect. 5.3

[19] see paragraph 5.3.2 in subsect 5.3.2 of sect. 5.3

closed constructions as is shown on fig. 4.3. A precise definitions follows in the next formal paragraph.

Formal Description 7 (CONCEPTS) *A concept of order n is a closed construction of order n. A concept is a concept of order n for some n.*

A concept C is a strictly empty concept iff C is an improper construction.

A concept C is a quasi-empty concept iff C constructs an empty set.

Let C be a concept that constructs an $((PossibleWorld, Time) \rightarrow \alpha)$*-object A for any type* α*. C is a concept empty with respect to* $W \in (PossibleWorld, Time)$ *iff A is not defined in W, or its value in W is an empty set. In this case C is called an empirical empty concept (cf. fig. 4.3 and the text following this figure).*

A concept C is a first order concept iff C constructs a first order object. A concept C is a higher order concept iff C constructs an higher order object.

These definitions give precise meaning to the expression-concept-object diagram from fig. 4.3. These definitions help to better understanding what a regions hierarchy in neocortex or in any memory-prediction system (cf. section 4.2) is about.

A brief summarization according to Marie Duzi shows the important features of concepts (see ⋇ [STA]$_{78}$:cite [DUZI99a][**?**]):

- Concepts are objective, independent of a particular (natural) language. Their objectiveness is of the same nature as the objectiveness of real numbers or any other mathematical objects.
- Concepts are 'abstract procedures' - sequences of steps leading to the identification of an entity (which can be first order object or higher order object or an empty set). The sequence of steps is expressed in the formal description by lambda terms.
- Concepts can fail to identify any entity - in case of strictly empty concepts.
- Concepts are timeless and spaceless. Concepts are first citizen of the cyberspace.
- Concepts are structured non set-theoretical entities–'complexes'. They consist of parts. The way of composing these parts together is important so as the parts themselves.

- People do not create concepts, they just discover them. In next pages we start to speak on categories which are tightly connected with concepts but not identical to concepts. Categories in opposite to concepts are created by people and manipulated by them.

- Concepts are meanings of NL expressions. Thus the sentence „The meaning of a concept changes“ is a non-reasonable one. The meaning of a concept cannot change, but the assignment of certain concept to a particular NL expression can change.

To work with concepts, i.e. to derive one concept from other concepts, or to explain what a definition precisely is, we need something like a conceptual system, intuitively. First we define a simple concepts, and using the notion of subconstructions we explicate what is meant by extension of a concept and by content of a concept. All these terms are widely used and almost often without a precise and exact meaning connected to them. From a vague interpretation of these notions a lot of misunderstandings in business starts.

In subsections 7.1.4 and 7.3.3 on integration agents and Service Systems usability in chapter 7 we will deal lot about integration of pieces of information from various existing data sources into one comprehensible view which will help us–service system's agents–in decision making and/or in acting in a proper way. Transformations from several conceptual systems into one integrating conceptual system will be discussed. The understanding of one professional group of agents by another group of agents being involved in different professions depends on these conceptual systems transformations heavily. Let's remember here the definition 2 of Service System, namely the claims dealing with contexts from which the agents' behavior is evaluated.

So, I would like to persuade you to read this following formal paragraph, although this will be not an easy reading.

Formal Description 8 (CONCEPTUAL SYSTEM) *The following explications are again based on ramified hierarchy of types. It will utilize a proper understanding of variables (atomic constructions) and trivial constructions.*

Simple concepts. *A simple concept is a construction* ^0X*, where* X *is a variable of any type* α *of the ramified hierarchy or an* α*-object which is not a construction.*

Thus, the simple concept is trivial construction of an object which is not a construction or of an variable, i.e. of an atomic construction. The invariant representations (see sect. 4.2) based on our perceiving of the real time-space help us to discover simple concepts.

Let X be a first order object of any complexity (i.e. an object of any type from simple types). Then ^0X is a simple concept.

If X is a construction of order n then ^0X is not a simple concept. But, any variable of type α when o-bounded creates a simple concept.

A function having constructions as arguments and/or value (i.e. the higher order object) gives us a simple concept only in the particular situation when it is trivially constructed.

Simple concepts point objects and this is clear without any additional explication. Not simple concepts point objects, too, but additional explication is needed to understand what the matter is.[20]

Concept's extension and concept's content. *The extension of a concept C is the object constructed by C. If the object constructed by C is an intension, i.e. a ((PossibleWorld, Time) $\to \alpha$)-object, we speak also about an extension of the concept C with respect to the state of world W $\in$(PossibleWorld, Time), which is the value of this intension in W.*

The content of a concept C is the set of all simple concepts that are subconstructions of C. The content of a simple concept is a singleton containing this simple concept. Thus, simple concepts are such concepts that their content is a singleton.

*This definition looks very similar to the standard way the philosophers explicate a notion of concept. But it makes it more precise, and as is shown in Materna's book [**?**] this notion of the content of a concept is in a way different from the traditional one.*

Having those basics we can define a conceptual system.

[20]Excuse me for the analogy, but we are living an „a theory of relativity realm". Thus, what is a simple concept depends on the context from which we judge the situation. And this context is given by the conceptual system we decided to move in. What is a simple concept in one conceptual system could be a complex one (not simple concept) in another conceptual system. Unfortunately for understanding these topics the decision is usually not made explicitly. But the good message is: having a conceptual system it is absolutely clear which concept is a simple one.

Conceptual system. *Let* C_1, ..., C_m *be simple concepts (constructions) of orders* $k_1, ..., k_m$*, respectively, where each* $k_j \leq n$*. Let* C_{m+1}, C_{m+2}, ... *be concepts, all of them distinct from any of* C_1, ..., C_m*, such that the subconstructions of* C_{m+i}, $i > 0$*, are only*

(a) members of the set $\{C_1, ..., C_m\}$*, and*

(b) variables ranging over those types that are composed of types given by C_1, ..., C_m*.*[21]

Then the set $\{C_1, ..., C_m\} \cup \{C_{m+1}, C_{m+2}, ...\}$ *is called a conceptual system of order n, and denoted as* CS^n*. A conceptual system, denoted CS, is a conceptual system of order n for some n.*

The set $\{C_1, ..., C_m\}$ *is the set of primitive concepts of CS, and denoted as PCS.*

The set $\{C_{m+1}, C_{m+2}, ...\}$ *is the set of derived concepts of CS, and denoted as DCS.*

Notice that among derived concepts there are concepts (constructions) of any order. Higher and higher order concepts arise by simply cumulating trivial construction. This will be in a way materialized in the so called R-edges in the diamonds introduced in chapter 5[22]*.*

The order of a conceptual system is determined by orders of its primitive concepts, only.

An important piece of knowledge is that the count of concepts in any conceptual system, whatever complex it is, is at most denumerably infinite. Not more! And this is a good message. From this point of view the ancient Greek philosophers endeavor (Pythagoras, etc. ...), to transfer every knowledge to natural numbers, doesn't look so unwisely as somebody could think of. From the Service Systems point of view this says that even the artificial agents could be able to work in a way with conceptual systems (as they are denumerable).

[21]Note, that „types that are composed of types given by C_1, ..., C_m“ means: types given by a construction are types of objects constructed by the construction or any of its subconstruction; type composed of types is a type given by a construction being any composition (i.e. any of the 8 construction modes can be used) of constructions which construct objects of these types.

[22]namely the Diamond of Attention Focusing and the Diamond of Cognitive Elements

Remembering Jeff Hawkins hypothesis the images of derived concepts could be conceived as inhabitants of higher levels of the cortical hierarchy while the primitive concepts dwell in lower levels, see fig. 4.2.

It is intuitive that the construction/concept is a universal modeling tool. Moreover, it is natural to see concepts and conceptual systems as a formal explication of the neocortex regions functioning, regions hierarchy organization and all the other musings of the section 4.2.

When we, people, perform any action the problems usually are not in performing the action but they are in speaking, thinking, evaluating, planning, etc., of this action. Shortly the problems are not with USE mode but with the MENTION mode. The side CONCEPT-OBJECT of the denotational triangle from fig. 4.3 could be clear now. As the side EXPRESSION-OBJECT is obvious, the remaining issue is the side EXPRESSION-CONCEPT. When this will be exactly comprehended we can expect we are prepared to value propositions effective formulating, to Service Systems proper planning and evaluation and to intelligent value co-creation.

But one problem still remains: it is the one with practising of such universal modeling tool in real business situations. This problem can be divided into two (sub)problems.

The first one is that it is practically impossible to lead each requirements analysis to the epistemic base. The standard analysis breaks the process on a level of description of something, which is called entity. These entities are more or less exactly described and a structure and/or behavior of a system is modeled over them. This is a problem of „short cuts“. A focused process or action or structure, shortly anything in focus, builds on lower level items of a detailed scope, which are mentioned in „constructions of the focused thing“ by their names = short cuts. Cf. the concept of hierarchy in section 4.2.

The second problem is that the world of constructions is „too ramified“ to be effectively and correctly used in modeling practice.

One possible solution of both of these problems is presented in the next section (cf. 4.4 about domain understanding and modeling.

However, to explicate this we need to understand deeply what a notion of definition means exactly. And not only the definition itself. Definitions and conceptual systems are very tightly connected to natural language and to languages in general. My opinion is that there we can find the roots of semantics and demystify the question what semantics is. At the same time this is the explication of the remaining side EXPRESSION-CONCEPT of the Expression-Concept-Object triangle, see again the fig. 4.3.

The next formal paragraph deals with definitions, with explicating what a definition itself is, with conceptual systems and their languages, and with linguistic definitions helping to build a hierarchy of languages with respect to a given conceptual system. This can look cumbersome but the idea is simple. The only issue is to stop to think „we have a completed language at our disposal and we only use it to express what we need in particular domain“. Instead quite opposite paradigm will help us: this what is called natural language (NL) is a big container for a huge set of particular domain languages more or less prepared to use them in respective domains and all the time being continuously developed. These particular languages depend on conceptual systems of the domains. In this sense we can say the domain is a context giving exact meanings to language expressions used within it. So as we discover concepts identifying objects of a studied domain we enrich our particular domain language with new terms and expressions representing the discovered concepts. Of course we use the pool of terms of the container called NL within this process. Moreover, there could be more special languages for one domain: the experts in the domain use conceptual system which primitive concepts are seen as derived concepts of the conceptual system used by learners in this domain. Thus the particular language used by experts differs from the particular language used by learners.

One term could be used in more conceptual systems as a simple expression, in one conceptual system representing a simple concept, but in other conceptual system of the same domain representing a derived concept. The same term could be used in other more or less distinct domain again as a simple expression representing either a simple concept of this new domain or a derived concept or both. Hence, it is not surprising that NL contains homonyms and synonyms. In turn, the presence of homonyms and synonyms in NL confirms this above introduced conception of NL.

Formal Description 9 (LANGUAGES AND CONCEPTUAL SYSTEMS.) *First, what is a definition, what means that something is defined. A definition is always tackled with respect to a conceptual system.*

Definition, definability. *Let CS be a conceptual system. Let a concept C be a member of DCS, and let C constructs an object A. If A is not constructed by a member of PCS, then C defines A or C is a definition of A.*

An object A is definable in the conceptual system CS iff some member of CS, precisely of its DCS, defines A.

Note, that objects constructed by primitive concepts of a given conceptual system are not definable in this conceptual system. It is not contradictory to our intuition as we assume them to be given a priori. The following is a trivial consequence of the above definitions: Every non strictly empty complex concept (i.e. not a simple one) defines some object in some conceptual system. In other words: Every non strictly empty complex concept is a definition in some conceptual system.

This notion of definition is not a standard one. The standard explication of what a definition is speaks about „definiendum" (i.e., what is defined) and „definiens" (i.e., by which it is defined). A standard approach is a linguistic one, as it is based on NL (natural language) expressions and natural language semantics. Well, standard definitions have the form

'Definiendum = Definiens'.

Some definitions here in this book have this form. Let's be aware of the fact, that it could be difficult to define something comprehensibly without this form. Thus, I try[23] *to bind the construction-based notion of definition with the linguistic-based notion of definition.*

First we define what can be considered to be an (in a way artificial) language which can serve as communication tool for expressing a given conceptual system.

Language of a conceptual system. *Let CS be a conceptual system based on $PCS = \{C_1, ..., C_m\}$. A language L_{CS} of conceptual system CS is a language satisfying the following conditions:*

1. *There are simple expressions in L_{CS} that represent C_1, ..., C_m.*

2. *If E is expression of L_{CS} that represents construction X of α-objects, then there is grammatical rule of L_{CS} that makes it possible to create expressions E_{vi}, i = 1, 2, ..., that represents variables ranging over the type α.*

3. *If E is expression of L_{CS} that represents construction X of α-objects, then there are grammatical rules of L_{CS} that make it possible to create expressions E_0, E_1, E_2 of L_{CS} that represent trivial construction, execution and double execution, respectively.*

[23] in alignment with Pavel Materna approach in his book Concepts and Objects – [9]

4. *If E_1, ..., E_k are expressions of L_{CS} that represent constructions X_1 (of β_1-objects), ..., X_k (of β_k-objects), respectively, then there is a grammatical rule (or a set of such rules) of L_{CS} that makes it possible to create an expression $E_{(X1,...,Xk)}$ of L_{CS} that represents the sequence construction $(X_1, ..., X_k)$.*

5. *If E_X is an expression of L_{CS} representing the sequence construction X (of $((\beta_1, ..., \beta_k)$-objects, then there is a grammatical rule (or a set of such rules) of L_{CS} that makes it possible to create an expression $E_{(i)}$ of L_{CS}, $i = 1 .. k$, that represents a projection $X_{(i)}$.*

6. *If E_X is an expression of L_{CS} representing a construction X, then there is a grammatical rule (or a set of such rules) of L_{CS} that makes it possible to create an expression E_λ of L_{CS} that represents an abstraction $\lambda x_1...x_k(X)$.*

7. *If E, E_1, ..., E_k are expressions of L_{CS} that represent constructions X (of $((\beta_1, ..., \beta_k) \rightarrow \alpha)$-objects), X_1 (of β_1-objects), ..., X_k (of β_k-objects), respectively, then there is a grammatical rule (or a set of such rules) of L_{CS} that makes it possible to create an expression $E_{[XX1...Xk]}$ of L_{CS} that represents the application $[XX_1...X_k]$.*

The language L_{CS} may not contain any 'linguistic definition'. Thus our second step is to create an artificial hierarchy of languages, each of which adds some new simple expressions, i.e. names, via linguistic definitions to the previous one.

Hierarchy of languages based on linguistic definitions with respect to CS. *Let SE_i^j denotes a simple language expression, i.e. not structured expression (not composed of other language expressions). It is usual to take such expression as a name of this item which is denoted by this simple expression.*

Let CE_i^j denotes a complex language expression, a structured expression composed of other language expressions.

Linguistic definitions in an (in a sense artificial) language of i-th level with respect to a conceptual system CS are introduced inductively:

(i) *Let L_{CS}^0 be L_{CS}.*

(ii) *Let* L^i_{CS}, $i > 0$, *results from* L^{i-1}_{CS} *by adding a set of simple expressions* SE^i_1, *...*, SE^i_k, $k > 0$, *together with expressions interpreted as true sentences:*

$SE^i_1 = CE^i_1$

...

$SE^i_k = CE^i_k$

where CE^i_1, *...*, CE^i_k *are complex expressions that contain only expressions occurring in* L^{i-1}_{CS}.

The expressions $SE^i_j = CE^i_j$, $i > 0$, $j = 1 .. k$, *are called linguistic definitions expressed by the language* L^i_{CS}.

Simple expressions SE^i_1, *...*, SE^i_k *are definienda, complex expressions* CE^i_1, *...*, CE^i_k *are definiens expressed by the language* L^i_{CS}.

Comparing these definitions with our experience in natural language using, we can add the following important remarks:

- *NL (natural language) contains a wide set of languages* L^j_{CSi} *of various conceptual systems* CS_i. *These could be various professional conceptual systems (like for mathematics, biology, law, financial business, retail business, IT services, ...) and to them connected hierarchies of languages; the upper index j denotes here a level of a particular language in the appropriate hierarchy.*

- *The whole hierarchy of languages based on linguistic definitions originated in a given conceptual system is embedded in NL. The hierarchy grows up to higher levels during the time in accordance with the step-by-step growing cognition of a given domain.*

- *The essential connection between a language and a conceptual system is given by the definition „Hierarchy of languages based on linguistic definitions with respect to CS".*

- *All languages belonging to the family* $\{L^i_{CS} \mid i > 0\}$ *are based on one and the same conceptual system CS. This conceptual system defines a domain. The domain is this what we try to understand and very often to model in a way in various situations in business. The domain is this, what have to be understood and modeled appropriately, when a Service System is created and value proposition is under deep analysis and precise formulation.*

- *Let us be aware of the fact, that the linguistic definitions create nothing new; they only introduce 'a short cuts', names, that enable our proper thinking and that enable improve our communication in the whole agent team (containing both, human agents and artificial agents) within a Service System. Each such 'short cut' is an (linguistic) abbreviation of some construction that is a derived concept of a given conceptual system.*

  It is, as I said, nothing new, but it is practically everything we need. These 'short cuts' recalls naming of sequences within the neocortex functioning (cf. section 4.2).

I would like to express my very opinion here: the above formal description of hierarchy of languages connected to a given conceptual system, which is based on linguistic definitions utilization, gives us an effective way to understanding between us–people mutually, and a between us–people and technology–artificial agents. In chapters 6 and 7 I will propose a way by which it could be done.

A most important issue is that we do not, more, waste our time by trying to create a kind of automatized speech recognizer to start a mutual comprehension between us–people and artificial agents. Understanding the hierarchy of languages based on linguistic definitions with respect to a given conceptual system we stop to try to analyze and to understand the current „final“ result of a longlife and ongoing evolution of NL, whilst we start to simulate this evolution process. Our benefit will be (and already is[24]) that in any step of this process we are happy that to some level the people-people and people-technology understanding is done, and we are looking forward to the step-by-step improvement of this understanding, which goes together with each new member to the language hierarchy added.

In any case the mystery of semantics was overcome. Simply saying, the semantics of a sentence, i.e. this, what gives to the sentence a sense, are concepts (= closed constructions) expressed by NL expressions forming the sentence. At the same time we have an universal modeling tool–the apparatus of constructions together with the hierarchical organization of a conceptual system and the appropriate hierarchy of connected languages based on linguistic definitions. These hierarchies are in natural alignment with the

[24] The so called Mycroft technologies developed in a spin-off of Masaryk university are providing this benefit.

cortical hierarchy of our brains and they together are in alignment with the hierarchical nature of the world.

Let's turn back to definition 2 of Service System. Please read it again carefully and with your mind full of context of conceptual systems and the hierarchy of languages of conceptual systems. As was said the conceptual system defines a domain. A domain is one of the most important contexts from which a Service System can be described, operated, evaluated. Thus to understand Service System really means to know its conceptual system and to be familiar with some languages of this conceptual system. The agents of a Service System have to share the Service System's conceptual system and have to share at least one common language of the conceptual system. The reversed triangle inequality for information capacities from the formal paragraph 6 is the plausible confirmation of this assertion. There are no doubts that if agent A understands language L_A and agent B understands language L_B than there must be a sequence of agents $A_1, A_2, ..., A_k$ of the Service System such that

- $A_1 = A$,
- $A_k = B$,
- the pair (A_i, A_{i+1}), i = 1 .. k-1, shares at least one common language.

Otherwise the system cannot function at all. But this is not enough despite of lot of companies and lot of managers and leaders in business think it is. And they behave according to this conviction. Actually, if agent B is a chief and derive conclusions from reports coming from set of agents containing agent A and A and B do not share one language then B can do nothing more than to make union of all logical consequences prepared by some agent A_j, who shares language with agent B and understands this what was prepared and maybe translated from agent A, with all other reports. This means B works on the left side of reversed triangle inequality.

All these considerations lead to a crucial statement of this chapter about cyberspace. (Remember please the GTM principle from the fig. 3.9.)

Statement 2 (Agent's mind mandatory content.) *Let SS be a Service System, $A_1, ..., A_k$ its agents. Let each one of agents $A_1, ..., A_k$ is aware of GTM principle*

Let the success of SS operation is defined as no agent expecting in the Client role some benefit has a reason to declare „my benefits are not realized“.

The probability of success of SS operation will decrease if any of the following conditions is not satisfied:

MMC1 *Agents $A_1, ..., A_k$ all together share at least one conceptual system CS.*

MMC2 *Agents $A_1, ..., A_k$ all together co-operatively create hierarchy of languages L^i_{CS} based on linguistic definitions with respect to CS and share this languages.*

MMC3 *All communications and information sharing within the SS life-cycle is provided by means of the most recent language from the hierarchy.*

Note that a situation can appear in which contexts of SS are such that benefit of one agent logically implies loss of another agent of SS. If the presumption on GTM principle is valid the only possibility to reach such situation is a misunderstanding in formulation of some value proposition. This problem[25] will be deeply discussed in chapter 6 about Service System modeling. In any case a probability of such misunderstanding increases when one of conditions MMC1 - MMC3 is not satisfied.

Hence the conclusion for Service Systems from the logical point of view can be expressed as follows: The aim of Service System's domain understanding and modeling is

1. to discover the relevant concepts of the domain,
2. to fix the conceptual system being used by all agents involved in the Service System, and
3. to develop and approve one language of the fixed conceptual system which all these agents will share.

[25]remember the GTM picture on fig. 3.9

4.4 Domain Understanding and Modeling

In this section you will find pragmatic basics covering needs of next chapters and the purpose of this book, only. These basics form excerption from the wider method of conceptual modeling, called HIT method of conceptual modeling [5, 4] The amendments to a complete description of the whole HIT method of conceptual modeling you will find in Appendix.

Starting points of this section are as follows:

A usual approach to modeling a part of reality for the purpose of IS building or business process improvement is based on an assumption that semantics cannot be captured and thus it is necessary to try to substitute this problem by various syntactical features.

My very opinion is that semantics is the core building block of understanding. Nothing can be more important than semantics when trying to co-operate and co-create a value in an environment of various stakeholders, various agents within a Service System. The barrier of natural dealing with semantics within models which aim is to support shared understanding of a domain has to be removed.

Till now the best known way to share semantics within a group of people is to communicate in natural language (NL). It is almost impossible to convince people to start to use and to share a common artificial language which could help the mutual understanding. Lot of attempts were done no emphatic results obtained. Recent development of , e.g., use-cases writing[26] or other documenting tools validates the NL unavoidability. The means we have to understand NL expressions and to formulate well understandable messages, statements and value propositions seems to be sufficient „from the logical point of view" (cf. the previous section).

I have almost 30 years of practice in semantics modeling of the needs of customers connected with their IS or business processes improvement, and 10 years of practice in teaching semantics modeling my students at university. Compared with other approaches the results are quite positive.

That is why I propose here an approach based on logical analysis of NL expressions and consisting in per each case development of appropriate sublanguage of NL–the language of the respective conceptual system, the conceptual system corresponding to the given domain.

[26] very good arguing you can find in Alistair Cockburn's book „Writing Effective Use Cases", see [**?**]

To understand a domain is not a passive reflection of structures, behavioral patterns, etc. of this domain; rather it is active discovering of underlying concepts and conceptual system fixing, and then creation of the effective language of the respective conceptual system. This sublanguage of NL will in turn possess several positive features:

- each term used in the language of conceptual system (L_{CS}) expressions is well defined and is unambiguous,
- each connection of some elements in the domain is expressed with precise and undoubted semantics,
- synonymy and homonymy of NL, which both currently make problems, are intelligently avoided for the domain under discussion.

Such sublanguage L_{CS} cannot be one universal for all domains as follows from the previous section 4.3. Rather it is a special language based on the respective conceptual system of this domain. But what universal is, it is the pattern according to which the particular domain language is created.

We can summarize: domain understanding consists in

1. discovering of the respective domain's conceptual system CS,
2. creation of a sublanguage of NL, the L_{CS}, which enables effective unambiguous communication and effective understanding of the domain,
3. items 1. and 2. are performed with full respect to the logical point of view explained in sect. 4.3 .

The process of and necessary competences to domain understanding and modeling I will explain in two steps: First, conceptual modeling–a practical approach–will be explicated. This part covers approximately what traditional conceptual data modeling practice does: development of conceptual database scheme. Second, knowledge modeling and management–again a practical approach–will be explicated. A switch from conceptual model to model will be explained and practical use case of modeling introduced. A model to modeling tool relation will be introduced together with knowledge management principles and principles of universal modeling.

4.4.1 HIT Method of Conceptual Modeling

HIT is an acronym for Homogeneous, Integrated, Type-oriented. The method is based on the HIT data model developed in years 1980-82 in Czech– see Appendix. The method itself was developed and established in the period 1981 - 1986. It is essentially an object-function model [**?**][⋇ [STA]$_{79}$:Scholl 1990 see Duzi], the specification and manipulation tool of which is not an object algebra but the 'language of constructions' introduced in the previous section.

This approach is not meant as a theoretical one only. On the contrary it is a pragmatic, in practice tested, method yielding material results. Thus the first issue is: To work with constructions and to leave all this anchored in the epistemic base according to the previous logical point of view is not practical henceforward. It will be 'too expensive' to reduce everything to the primitives over epistemic base. Thus we will introduce so called *basic sorts* as pragmatic short cuts of the semantics derivation process. As we want to have everything pigeon holed in a studied domain I will introduce a special form of linguistic definitions explaining what dwells in particular basic sorts.

The second issue is: To create a conceptual system together with its language hierarchy and to maintain it by means offered in logic is again not practical. But there is a time-tested practice of creating conceptual schemes in database and data modeling community. So, our way will be to follow the tested practice and to support this process by the introduced good logical background.

Let's start with definition of conceptual model.

Definition 6 (Conceptual model.) *Conceptual model of a domain under discussion is defined as a triple*

*(**BS**, **Attr**, **CC**),*

*where **BS** is a set of basic sorts, **Attr** is a set of concepts (constructions) of the so called HIT-attributes over the set of basic sorts **BS**, and **CC** is the set of constructions of consistency constraints connected with attributes of **Attr**.*

The components of conceptual model are explicated in the following.

The process of finding basic sorts is called sortalization. From the pragmatic point of view the process of modeling of a particular part of world being in interest is similar to focusing when making photos: There is everything before our objective but our snap-shot pictures only those things clearly, that are focused on. Sortalization is a kind of attention focusing.

When creating a database schema or visualization or conceptual model, we need to determine nodes of this schema/model which stand for some basic classes of objects of our interest. When creating a process model in a business area, we again need to determine nodes of this model which stand for elementary processes, events, branching and joining blocks. These are again basic classes of objects of our interest in this case. Speaking about objects of our interest we do not use formulations like

„the class of individuals (members of Universe) which have the property of being an employee“

or

„the class of such higher order objects (members of $Cons^n$ for an appropriate n) which have the property of being database schema of primary information systems of our DWH (data warehouse)“

or

„the class of objects which have the property of being an event relevant to our business interest“

or

„the class of objects which have the property of being a function executed within a given process or the property of being a process branching block or the property of being a branches joining block“.

Instead we simply speak about entity set EMPLOYEE or entity set DATA SCHEME or entity set EVENT or entity set PROCESS ELEMENT. Note that this corresponds to naming of sequences in the cortex hierarchy of regions (as was discussed in sect. 4.2) and to the *process of creating the hierarchy of languages based on linguistic definitions with respect to given conceptual system.* The only problem is, we do not know what conceptual system is the given one; we have to discover this in course of the requirement analysis in parallel with this language hierarchy creating.

Expressions like EMPLOYEE, DATA SCHEME, EVENT, PROCESS ELEMENT denote various properties of individuals or other objects of any order, and we demand an information system (or another model) in question should offer information on any object which has the selected property. On the other hand this information system or other model must not offer information on any object which has not the selected property. This means that the attributes of a model should concern - and thus be restricted to - only objects with this property.

Two kinds of basic sorts will be distinguished similarly as in other current data models it is usual: entity sorts and descriptive sorts.

entity sorts: Let $R \subseteq Time$ be a reasonable time environs (was-is-will be) of the present. Let $r \in R$ be a time point, and $w_a \in PossibleWorld$ be the actual possible world (i.e. this one in which we actually live, regardless we don't know which one it is). Let T be a type of ramified type hierarchy, and

$P/((PossibleWorld, Time) \rightarrow (T \rightarrow Bool))$

be a property of T -object. Denoting $C(P, r, w_a)$ the class of T -objects generated by a property P in a time point r in the actual possible world w_a, i.e. a class containing each T-object having the property P in actual world w_a in the given time point r, the *entity sort* S is defined as

$\cup_{r \in R} C(P, r, w_a)$.

Classes $C(P, r, w_a)$ are called populations of the entity sort S in particular time points r.

Each entity sort has a name which used to be written in a form (#Name of entity sort).

Rule 1 (Entity Sort Definition) *Each entity sort in **BS** must be defined according to the following pattern:*

„An element of the sort (#Name of entity sort) is each object possessing property P.“

or

„Entity sort (#Name of entity sort) is defined in such a way that it contains each particular element possessing property P.“

Property P has to be expressed by appropriate NL expression. The word „object“ can be substitute by a name of more specific category which is generally known and used in modeled domain.

Entity sorts definitions are special kinds of linguistic definitions. They introduce simple terms or almost simple terms as names of something which could be conceptually complex.

Definitions of this kind help us to anchor our mind in required conceptual system. The entity sort's definitions hide the complexity of constructions constructing properties which determine these sorts. At the same time they hide a little bit bizarre handling with actual possible world and reasonable time environs of the present within the theoretic explication of what an entity sort is.

The following are examples of entity sort definitions:

(#Article). „An element of the sort (#Article) is each such object that is a product or service or law and which is or can be a subject of purchase or sale and all these properties inclusive of products, services, laws not existing till now but possibly created for the purpose of company's investment activities."

(#Document). „An element of the sort (#Document) is each such object that is a record or message and the recording of which makes sense for the company under discussion."

(#Business Partner). „An element of the sort (#Business Partner) is each such object that is or was or can be a business partner of our company and that is interesting from the company's development activities point of view. "

Other examples you will find in next chapter 5. Entity sorts definitions form a glossary or term used list. It is widely recognized that when describing a new system or producing a plan of an extensive action it is necessary to have such a term used list.

To be sure you know how to formulate the definitions, try to define particular entity sorts corresponding to terms EMPLOYEE, DATA SCHEME, EVENT, PROCESS ELEMENT.

descriptive sorts: A class C is representable iff there is a recursive injection[27] of C into a set A*, where A* is the set of all finite strings over a finite alphabet A. A descriptive sort is any recursive, i.e. (in computer) representable class.

Each descriptive sort has a name which used to be written in a form (Name of descriptive sort).

Rule 2 (Descriptive Sort Definition) *Each descriptive sort in **BS** must be defined according to the following pattern:*

„An element of the sort (Name of descriptive sort) is represented by a string satisfying the conditions C1 and/or C2 and/or ... "

where C1, C2, ... are constraints limiting admissible values.

[27]injection means if f(a) = f(b) then a = b; recursive function means (simplifying a little bit) there exists a computer program containing only finite for-loops, no while-loops and no goto instructions, which computes values of this function

For the sake of convenience we will assume with no other reference that **BS** includes sorts representing the type *Time* and the type *Boolean* of the epistemic base EB.

An example of descriptive sort definition is:

(Date). „An element of the sort (Date) is represented by a string satisfying the following conditions: (1) it is an integer 8 bytes long, and (2) it is of the form RRRRMMDD."

Type* system instead of Type system (Ramified Hierarchy). Definitions of basic sorts **BS** all together claim the domain. At the same time they help us to discover concepts and to create the first picture of a conceptual system of the domain. And in parallel the first steps of the hierarchy of languages of this domain conceptual system starts to be built.

Note that basic sorts form a set of non-empty sets. The only difference comparing it to the base of a proper type system is that basic sorts are not pairwise disjoint, cf. the paragraph Subsorts, Supersorts below in this subsection. We can use the set of basic sorts in analogical way as if it was a base and create a huge collection of objects over these basic sorts by means of the eight modes of constructions we are familiar with. But we have to be aware of the fact that some functions over such set **BS** will fall into more than one class constructed analogically to a type system construction over a proper base. This is „a tax" paid for the pragmatic simplification performed by sortalization.

Nevertheless, we do it! The set of basic sorts will be called *base* of sorts*, so the symbol **BS** means both, set of basic sorts or base* of sorts. The hierarchy of types* is constructed over **BS** analogically as a type system is constructed over proper base. So we have:

I. every member of **BS** is a type* over **BS** (so called node type*),

II. if $T_1, \ldots, T_n$, $n > 1$, are types* over **BS**, then the cartesian product of $T_1, \ldots, T_n$, denoted by $(T_1, \ldots, T_n)$, is a type* over **BS** (so called tuple type* or sequence type*),

III. if T_1 and T_2 are types* over **BS**, then $(T_1 \to T_2)$, i.e. the collection of all partial functions from T_1 into T_2, is a type* over **BS** (so called functional type*),

IV. nothing except I. - III. is a type*.

An element X of a type* T is called an object of type* T or an T-object, denoted X/T. Thus all we prepared in the section 4.3 can be used with base* of sorts.

More about the sortalization in comparison with another current approaches to this issue you will find in Appendix.

HIT-attributes. Relations between the individual elements of the sorts are described by the so-called HIT-attributes, which constructions are elements of the set **Attr**.

HIT-attributes are empirical functions (see formal paragraph 4) of the so called simple types*. Such function can either map a cartesian product of sorts into another cartesian product of sorts or into the set of all subsets of the another cartesian product. The sorts can be both entity sorts and descriptive sorts; at least one of these sorts must be entity sort.

If a HIT-attribute maps into cartesian product, we call it a singular attribute, if it maps into the set of all subsets we call it a plural attribute.

Usually in practice the second cartesian product, i.e. the range of this function, is reduced into one sort, entity one or descriptive one.

Precisely, HIT-attributes are (empirical) functions of the type*

(a) $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow (S_1, ..., S_m)))$

or

(b) $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow ((S_1, ..., S_m) \rightarrow Boolean)))$,

where $T_1, ..., T_n, S_1, ..., S_m$ are sorts from the set of basic sorts **BS** and at least one of T_i, S_j is an entity sort. These sorts need not be necessarily distinct.

The number n+m is called *complexity* of the HIT-attribute.

Let $T_1, \ldots, T_n, S_1, \ldots, S_m$ be sorts from **BS**. Let $S = (S_1, \ldots, S_m)$ be so called tuple-sort (sequence construction used). Let x_i be a variable constructing T_i-objects, y be a variable constructing S-objects, C be a construction of Boolean-objects containing as free variables only w::(PossibleWorld, Time), $x_i :: T_i$, and y::S. Then a HIT-attribute A is defined by a construction

(a') $\lambda w \lambda x_1 ... x_n \iota y (C_w)$,

or

(b') $\lambda w \lambda x_1 ... x_n \lambda y(C_w)$,

according to whether it is of the first kind (single-valued) or of the second kind (multi-valued).

Rule 3 (Attribute semantics specification) *Each construction of a form (a') or (b') from the list* ***Attr*** *has to be associated with a just one formalized natural language expression that expresses a notion of function, namely*

(a") $text_0(S)text_1(T_1)text_2(T_2)...text_n(T_n)text_{n+1}$
in case (a'), and

(b") $text_0(S)$*-s* $text_1(T_1)text_2(T_2)...text_n(T_n)text_{n+1}$
in case (b'),

where S *and* T_i *are conceived to be names of respective sorts.*

Being read the expressions of a form (a") or (b") give us a notion of function giving a value of S *or of* $(S \rightarrow Boolean)$*, respectively, on arguments from a tuple* $(T_1, ..., T_n)$*. Only the* $text_0$ *and/or* $text_{n+1}$ *can be omitted, here.*

Expressions (a") and (b") we call slightly formalized names of HIT-attributes in natural language or, simply, HIT-attribute formalized names.

Recording a conceptual model of a domain it is enough to fill the list ***Attr*** *by expressions of the form (a") and/or (b"), only, as the respective constructions are obvious.*

The rule is O.K. in course of practice modeling. In this case the HIT-attribute formalized name usually stands for its construction. But when dealing with both, the names of attributes and the constructions of attributes, we have to distinguish what is the name and what is the construction. Speaking of attribute A we mean the 'A' is name of this attribute; than we must write $^0A \in$ **Attr** to express the attribute belongs to conceptual model. Speaking of $A \in$ **Attr** we mean the 'A' is construction of an attribute; than to speak about the attribute itself we have to write ^1A which denotes execution of the construction A and yields the attribute constructed by A.

The following examples show the practical utilization of this naming of attributes.

Examples: Previous formal definitions can be illustrated in following examples.

1. A1 = Planned delivery (#Product) of given supplier (#Business Partner) to given (#Project) in given (Date).

2. A2 = Obtained deliverables (#Product, Amount)-s of given supplier (#Business Partner) to given (#Project) till given (Date).

The first case, A1, is a (partial) function that depending on a state of the world provides a mapping

(#Business Partner, #Project, Date) → #Product,

i.e. to each triple $\langle businesspartner, project, date \rangle$ it assigns either nothing or some product, that one which is planned to be delivered by this business partner to this project.

The second case, A2, is a function (again depending on the state of the world)

(#Business Partner, #Project, Date) → ((#Product, Amount) → Boolean),

that assigns a set of couples $\langle product, amount \rangle$, which can be empty, to each triple $\langle businesspartner, project, date \rangle$. Again only these couples $\langle product, amount \rangle$ will be assigned which represent obtained deliverables from the particular business partner to the particular project received up the particular date.

More examples from usual domain understanding and modeling you can find in Appendix. But to exercise this semantics recording it will be enough to continue reading as the chapter 5 is full of definitions using this semantics recording.

More about HIT-attributes. Comparing this approach to the Expression-Concept-Object diagram from fig. 4.3 we can see (slightly formalized) natural language expressions (a") and (b") representing constructions (i.e. concepts) (a') and (b'), respectively, and denoting simple tables of facts (empirical functions)–objects. These objects (tables of facts) are identified by constructions (a') or (b'), respectively.

When speaking on HIT-attributes generally, namely in statements formulation and their proofs, we will use one unifying lambda expression for them: Let w::(PossibleWorld, Time), C be a construction of Boolean-objects, x_i :: T_i, for each i = 1 .. n, and let y::S in case (a) of single-valued HIT-attribute, or y::($S \rightarrow Boolean$) in case (b) of multi-valued HIT-attribute.

Then a HIT-attribute A is defined in both cases (a) and (b) by a construction

$\lambda w \lambda x_1 ... x_n \iota y(C_w)$.

Using this convention we define: Two HIT-attributes

$A1 = \lambda w \lambda x_1 ... x_n \iota y(C_{1w})$

$A2 = \lambda w \lambda x_1 ... x_n \iota y(C_{2w})$

are identical iff the construction C_{1w} is equivalent to the construction C_{2w} for all $w \in (PossibleWorld, Time)$.

For the purpose of questions answering it is necessary to use tuple sorts $S = (S_1, \ldots, S_m)$ in the HIT-attribute range.

For the purpose of modeling a part of reality it is better to use non-tuple sorts only in the HIT-attribute range. You can see in further text that each attribute with a tuple sort in range can be transformed into an attribute with non-tuple sort in its range while the information given by this second attribute is the same as information given by the first one.

Consistency Constraints. Consistency constraints are propositions that specify admissible states-of-affairs. Thus consistency constraints are objects of a type

$((PossibleWorld, Time) \rightarrow Boolean)$.

There are some states-of-affairs that are 'a priori' excluded. For instance, such a possible world in which the age of a person would decrease is impossible. But we have also to exclude such states-of-affairs that are logically possible but that contradict some empirical laws or conventions. For example conventions of a given organization.

Consistency constraints excluding logically impossible states-of-affairs are called *analytical constraints*, constraints excluding logically possible but contradicting some empirical laws or conventions are called *empirical constraints*.

This distinction is significant, since empirical constraints can evolve in time.

Consistency constraints help us to check the correctness of the attribute values. When consulting the business reality with a user or a domain expert, the designer has to use a natural language. But since consistency constraints are the third constituent of the conceptual model, the task of a designer is to analyze these natural language assertions and to transform them into the respective constructions.

Rule 4 (Consistency constraint specification) *Each construction of a consistency constraint from the list **CC** has to be associated with a proposition explicating the constraint. The proposition is recommended to be in the form of the „if ... then ... “ statement, or a logical composition of such statements. When a constraint is a complicated one the proposition construction is added either using formal lambda term expression or using other math or logic or programming means.*

Examples:

(1) Analytical constraint 'the age of a person never decreases' connected with attribute

A3 = (Age) of given (#Person)

is recorded by the following construction:

$\forall wt_1t_2person((t_1 \geq t_2) \supset ([^0A_{wt1}person] \geq [^0A_{wt2}person]))$,

where variables w, t_1, t_2 ranging over possible worlds and time points, respectively, variable 'person' ranging over the sort (#Person).

(2) Empirical constraint 'For each material there is always a supplier' connected with attribute

A4 = (#Supplier)-s of given (#Material)

is recorded by the following construction:

$\lambda w \forall matr \exists supp[[^0A4_w matr]supp]$,

where variable w ranging over (PossibleWorld, Time), $supp \in (\#Supplier)$, $matr \in (\#Material)$.

(3) Let our conceptual model is

(

{(#Article), (#Business Partner), (#Business Case), (Date), (Amount) },

$\{^0A1, ^0A2\}$,

CC

)

from the examples above in paragraph about HIT-attributes. (Note that A1, A2 are HIT-attributes formalized names, while the set **Attr** being the second constituent of conceptual model triple is a set of constructions, not names!)

Let's now we want to express our knowledge of a modeled reality, i.e. to fill the **CC** by an item, that „each received delivery must be planned in advance". It is again an empirical constraint. This constraint is connected with both attributes, A1 and A2. Using natural language to formulate this property of our model we obtain:

C = If (a given instance of (#Article) with some number of (Amount) is in range of A2 for a given time moment $t \in (Date)$ being together with an instance of (#Business Partner) and an instance of (#Business Case) in the domain of A2), then (this instance of (#Article) is in range of A1 for the above instance of (#Business Partner), and the above instance of (#Business Case) together with time moment $t' < t$, $t' \in (Date)$).

Using a more formal expression, we obtain a formulation which may be more comprehensive:

Suppose w ranging over (PossibleWorld, Time), and product ∈ (#Article), t, t' ∈ (Date), a ∈ (Amount), supplier ∈ (#Business Partner), project ∈ (#Business Case). Then

C = $\lambda w \forall$ t product a supplier project
(if $((product, a) \in [{}^0A2_w(supplier, project, t)])$
then
$((\exists t' \in (Date)(t' < t)) \wedge (product = [{}^0A1_w(supplier, project, t')])))$.

Note the form of the constructions of consistency constraints: $\forall w$... concerns analytical constraints, λw... concerns empirical constraints.

Ratio as a special consistency constraint An attribute A can be provided with the so-called ratio of the attribute that mirrors the appropriate singular and total integrity constraints. The ratio is written in the form

/ p,m : q,n
where $\forall w \in (PossibleWorld, Time)$

- p = 0 means that the ${}^0A_w$ is a partial function
- p = 1 means that the ${}^0A_w$ is a total function
- m = 1 means that the ${}^0A_w$ is singular
- m = M means that the ${}^0A_w$ is multivalued.

The values of q and n express the same constraints for the reversed function, i.e. function of the type*

$(PossibleWorld, Time) \rightarrow (S \rightarrow T)$

or

$(PossibleWorld, Time) \rightarrow (S \rightarrow (T \rightarrow Boolean))$

if A has been of the type*

$(PossibleWorld, Time) \rightarrow (T \rightarrow S)$

or

$(PossibleWorld, Time) \rightarrow (T \rightarrow (S \rightarrow Boolean))$,

where T, S are (tuple) sorts.

Rule 5 (Cardinality.) *Each formalized NL expression expressing a concept of a HIT-attribute has to be followed by the record of ratio to visualize the cardinality of the connection.*

Examples of a way of ratio recording are the following (these are extensions of the previous examples):

A3 = (Age) of given (#Person) / 1,1:0,M.

The part 1,1 of the ratio tells us that each person has at least one age and at most again one age, i.e. there is exactly one number representing an age of a person.

The part 0,M of the ratio tells that taking a number from descriptive sort (Age) there could be nobody who is of this age, but there could be several persons of this age.

A4 = (#Supplier)-s of given (#Material) / 0,M:1,M.

The part 0,M of the ratio means: to a given material there could be no known supplier (the zero marks it) but on the other hand there can be several suppliers. This example is intentionally in contradiction with the above example of consistency constraint.

The part 1,M of the ratio means: taking a supplier from (#Supplier) there must be at least one material supplied by him, as in other case he will be no supplier, and there can be more materials supplied by him.

But we can, in another context, to see the situation differently:

A4' = (#Supplier)-s of given (#Material) / 0,M:0,M.

In this case the second part 0,M of the ratio tells that there can be suppliers to whom we do not know by them supplied material.

Important message is that which of these both, A4 and A4', is correct depends on the point of view, i.e. on context of the analysis performed. And the context in turn depends on purpose of this analysis. Any case we cannot decide which of these both is better without experiencing the situation in „the real life“!

This (logically) follows from the fact that this HIT-attribute, so as each HIT-attribute, depends on a state-of-affairs.

A1 = Planned delivery (#Product) of given supplier (#Business Partner) to given (#Project) in given (Date) / 0,1:0,M.

This attribute represents a more complex connection: to each triple (business partner, project, date) a concrete project is assigned. The part 0,1 of the ratio says there can be a triple (business partner, project, date) to which no product is assigned, but if is assigned than only one, not more.

The part 0,M of the ratio says a given product can be not assigned to some triple (business partner, project, date) but it can be assigned to more such triples.

Attribute A1 can be rewritten to

A1' = Planned deliveries (#Product)-s of given supplier (#Business Partner) to given (#Project) in given (Date) / 0,M:0,M.

In this case a business partner can deliver to a given project in a given date more than one product. The contemplation could be similar to the one with attributes A4 and A4'.

A2 = Obtained deliverables (#Product, Amount)-s of a given supplier (#Business Partner) to a given (#Project) till a given (Date) / 0,M:0,M.

This is again a more complicated connection: a couple of items (one entity and one description) is assigned to a triple (two entities and one description).

The first 0,M denotes that to given triple (business partner, project, date) there can be no deliverable (expressed as a couple (product, amount)) but there can be more of such deliverables.

The second 0,M denotes that in the 'inverse' function, i.e.

(#Product, Amount) → ((#Business Partner, #Project, Date) → Boolean), the same is true: to a given couple (product, amount) there can be assigned no triple (business partner, project, date) but there can be assigned more such triples.

And these previous examples at the same time show the right way how to write attributes when specifying a conceptual model.

Subsorts, Supersorts, Inheritance. In case of descriptive sorts the relation of 'being a subsort' is defined as the set-theoretical inclusion.

But it is not correct to define the same way this relation for entity sorts. In case of entity sorts the relation is determined by the fact that some properties are not logically independent.

An entity sort $E1$ which is determined by a property $P1$ is a subsort of an entity sort $E2$ which is determined by a property $P2$ iff the property $P1$ necessarily implies the property $P2$, i.e.

$\forall wt \forall x([^0 P_{1wt} x] \supset [^0 P_{2wt} x])$,

where w, t, x range over possible worlds, time-points and individuals (or generally objects of a type α), respectively. In turn we say the sort E_2 is a supersort of the sort E_1.

The consequence of this dependency is the fact that in all the states-of-affairs the population of $E1$ is a subset of the population of $E2$. Hence in conceptual modeling the relation of 'being a subsort' is a necessary relation. This contradicts to the approach applied sometimes in artificial intelligence, where a not necessary relation taken as a fact is sometimes considered to be the so called ISA relation (e.g. Hemingway is a novelist).

Generalization and specialization. Let $E_1, ..., E_n$ be entity sorts defined by properties $P_1, ..., P_n$, respectively. Let each P_i, i = 1 .. n, concept (construction) contains[28] concepts of some properties $Q_1, ..., Q_m$. We can define a new entity sort E specified by properties $Q_1, ..., Q_m$ as a *generalization* of entities $E_1, ..., E_n$.

For example generalizing entity sorts (#Car), (#Motorcycle), (#Bicycle) we obtain a new entity sort (#Vehicle), generalizing further (#Vehicle) together with a (#Plane) and (#Ship), we obtain (#Transport means).

It usually holds in case of generalization that in every state-of-affairs the populations of sorts $E_1, ..., E_n$ do not overlap and the set-theoretical union of these populations covers the population of the new sort E. As all sorts

[28] as a content, see concept content and concept extension in sect. 4.3

$E_1, ..., E_n$ possess properties $Q_1, ..., Q_m$ defining the superentity E, we say that each E_i, i = 1 .. n inherits properties of E.

On the other hand, specialization is an 'opposite' way of defining a new entity sort. Having an entity sort E specified by a property P, we define new entity sorts $E_1, ..., E_n$ by specifying some additional properties $P_1, ..., P_n$ which an individual having the property P may present or which it may lack to present. These new sorts are *specializations* of the sort E.

As all sorts $E_1, ..., E_n$ possess property P defining the superentity E, we can say again that each E_i, i = 1 .. n inherits properties of E.

For instance, having the entity sort (#Bird), we can define (#Penguin) as a bird that does not fly, or (#Water bird), (#Domestic bird), etc. Or, from the entity sort (#Person) we can specialize sorts (#Employee), (#Student), (#Lecturer), etc. These sorts may overlap, and an object may change such roles without changing its underlying or fundamental identity. E.g. a student might become an employee or cease to be an employee without losing his/her underlying property to be a person.

Conceptual model diagram. People like to watch pictures when trying to understand a domain. Thus it is usual in data modeling practice to depict a conceptual scheme in a kind of diagram. Lot of techniques exist but when you are familiar with one of them you will have no problems to understand pictures produced in another one. There are, except others, classical ER diagrams and there is UML (Unifying Modeling Language) which tries to be a standard and almost is a standard.

Which kind of diagram you will choose to visualize your conceptual model is absolutely free. It is important to understand that the diagram is not the conceptual model itself. The conceptual model is this what was explicated above and the diagram is only a visualization means.

I prefer the so called crow's food notation which was very popular in 80's and 90's of the last century and which seems to me very transparent. An example of conceptual model diagram is on fig. ⋇ [STA]$_{80}$:TODO: figure of a simple conceptual model

⋇ [STA]$_{81}$:figure Example of Conceptual model diagram TODO

Particular icons have common meanings ⋇ [STA]$_{82}$:TODO: explanation of particular icons.

Entity sorts of our HIT conceptual model we depict by boxes with small square representing kernel entities. Attributes of complexity 2 are represented either by edges of the diagram either they are hidden in boxes. They are hidden in boxes representing particular entity sort if such attribute contains (except of this particular entity sort) a descriptive sort. Complicated attributes of complexity 3 or more, namely if at least three entity sorts are connected, are represented by boxes with small diamond, which in this notation represent relationship entities. ⁕ [STA]$_{83}$:TODO

4.4.2 Operations in Cyberspace

Trying to understand a domain we use not only empirical functions of simple types, i.e. HIT-attributes, but to express an algorithmic behavior we use analytical functions, too. The semantics of an analytical function (operation) is described by usual means of mathematics, logic and/or informatics (some kind of programming language or pseudocode or script is appropriate). We will suppose we have a complete programming language at our disposal when trying to understand a domain or to model it. An example of such analytical function can be

$Avg(V) = Sum_{vi \in V}(v_i)/Count_{vi \in V}(v_i)$,

which gives an average value on each argument V which is a set of values. Often we will use analytical functions in combination with empirical ones. This means simply computing with empirical values from a database or another kind of memory.

In the following considerations we will use a special class of analytical functions–the so called surjections. Analytical function

$f/A \rightarrow Z$

is a surjection iff

$\forall z \in Z(\exists a \in A([fa] = z))$.

Function which is not a surjection can collapse lot of points from the space A into one point in the space Z. When such collapsing function is applied to a construction constructing Boolean values it can yield either for all arguments True or for all arguments False. In the first case it constructs tautology, in the second one contradiction. Transformations changing everything into tautology or into contradiction are useless.

In the following I will omit the prefix 'HIT-' for the convenience often. Thus, everywhere in the following text in this chapter 'attribute' means HIT-attribute.

Definability is a fundamental concept of HIT method of database design.

Let $A, A_1, ..., A_n$ be HIT-attributes. Let w, f be variables ranging over (PossibleWorld, Time) and surjections, respectively. We say that a HIT-attribute A is *definable* over a set of HIT-attributes $\{A_1, ..., A_n\}$ iff

$\exists f \forall w([{}^0 Aw] = [{}^0 f([{}^0 A_1 w], ..., [{}^0 A_n w])])$,

and denote it $A \leftarrow \{A_1, ..., A_n\}$.

An attribute A is definable over an attribute B iff it is definable over the set $\{B\}$.

Let $\mathbf{A} = \{A_1, ..., A_n\}$, $\mathbf{B} = \{B_1, ..., B_m\}$ be sets of attributes. A set $\mathbf{A}$ is definable from a set $\mathbf{B}$, denoted $\mathbf{A} \leftarrow \mathbf{B}$, iff every member of $\mathbf{A}$ is definable over a subset of $\mathbf{B}$.

Let $\mathbf{A} \leftarrow \mathbf{B}$ and $\mathbf{B} \leftarrow \mathbf{A}$. Then we say that sets of attributes $\mathbf{A}$ and $\mathbf{B}$ are informationally equivalent, and denote $\mathbf{A} \approx \mathbf{B}$.

Two attributes A and B are informationally equivalent iff $\{A\} \approx \{B\}$.

Examples. Defining attributes over other attributes or from sets of attributes is a way of answering queries over a database or another kind of memory.

⋇ [STA]$_{84}$:TODO Examples from DM1 lectures

Information capability of HIT-attributes. Let's have two informationally equivalent sets of attributes $\mathbf{A}$ and $\mathbf{B}$, $\mathbf{B} \approx \mathbf{A}$, i.e. it is true that both are valid, $\mathbf{A} \leftarrow \mathbf{B}$ and $\mathbf{B} \leftarrow \mathbf{A}$. It is easy to see that the relation $\approx$ is an equivalence relation. A trivial consequence of the definition of 'definability' is the following, in business practice important, statement.

Statement 3 (Redundant attributes.) *Let $A \leftarrow \{B_1, ..., B_n\}$. Then $\{A, B_1, ..., B_n\} \approx \{B_1, ..., B_n\}$. In verbose form: Adding a definable attribute to a set of attributes we obtain nothing new and removing a definable attribute from a set of attributes we lose nothing.*

The relation of definability between sets of attributes is a quasi-ordering, as it is reflexive and transitive but not antisymmetric (if $\mathbf{A} \leftarrow \mathbf{B}$ and $\mathbf{B} \leftarrow \mathbf{A}$ then it is not necessary that $\mathbf{A} = \mathbf{B}$, see the above statement). From algebra it is known that in such a case a partial ordering on the set of equivalence-classes can be defined.

Let **BS** be a base* of sorts, **AA** be a set of all attributes that can be constructed over **BS**. Then the factor set $\mathbf{AA}/\approx$, i.e. the set of all equivalence-classes of attributes, is partially ordered by a relation $\angle$, which is induced by the definability relation. The relation $\angle$ of partial ordering on equivalence-classes of attributes is called an *information ordering.*

Each equivalence-class from $\mathbf{AA}/\approx$ determines an *information capability.* The information ordering is a partial ordering of information capabilities.

Consequences of information capability definition. Let $\mathbf{A}$ be a set of attributes from $\mathbf{AA}$. Then there is exactly one class

$C_{\mathbf{A}} \in \mathbf{AA}/\approx$

such that

$\mathbf{A} \in C_{\mathbf{A}}$.

Simply said: each set of attributes belongs to exact one equivalence class of attributes. Such class can be called class generated by the set of attributes. All sets of attributes which are informationally equivalent generate the same equivalence class of attributes.

Let's have two sets of attributes $\mathbf{A}$ and $\mathbf{B}$. Let $C_{\mathbf{A}}$ and $C_{\mathbf{B}}$ be respective equivalence classes generated by $\mathbf{A}$ and $\mathbf{B}$, respectively. Than the following is true:

1 $C_{\mathbf{A}} \angle C_{\mathbf{B}}$ iff $\mathbf{A} \leftarrow \mathbf{B}$.

2 Each set of attributes from $C_{\mathbf{B}}$ has a greater information capability then any set of attributes from $C_{\mathbf{A}}$.

3 If for $C_{\mathbf{A}}$ and $C_{\mathbf{B}}$ neither $C_{\mathbf{A}} \angle C_{\mathbf{B}}$ nor $C_{\mathbf{B}} \angle C_{\mathbf{A}}$, then information capability of any set of attributes from $C_{\mathbf{A}}$ is incomparable with the information capability of any set of attributes from $C_{\mathbf{B}}$.

In [6] I proved that information capabilities form a congenial algebraic structure–a complete lattice. The proof you can find in Appendix.

Note that information capability of HIT-attributes is defined in parallel with information ordering of sets of propositions in section 4.3 From the Logical Point of View. The function Cn, logical consequence, plays the same role as the equivalence class of attributes.

Rotation of attribute. The simplest way to define an attribute over another attribute is the so called rotation. Let A be an attribute given by the construction

$\lambda w \lambda x_1 .... x_m \Box x_{m+1} ... x_n([{}^0A_w(x_1, ..., x_m)] * (x_{m+1}, ..., x_n))$

where w ranges over (PossibleWorld, Time), $x_1, ..., x_n$ ranges over sorts $T_1, ..., T_n$, respectively, $\Box$ stands for λ or ι , and $*$ stands for application or equality.

Let $(i_1, ..., i_n)$ be any permutation of indexes (1, ..., n).

Then an attribute rotA given by construction

$\lambda w \lambda x_{i1}....x_{ik} \Box x_{ik+1}...x_{in}([{}^0A_w(x_1, ..., x_m)] * (x_{m+1}, ..., x_n))$

is called a rotation of attribute A.

If $\Box$ in the construction of rotA stands for λ we say it is a plural rotation. If $\Box$ in the construction of rotA stands for ι we say it is a singular rotation.

An attribute A will be called a plural attribute if it is given in plural rotation and it will be called a singular attribute if it is given in singular rotation. In other words, a plural attribute is a multivalued attribute while a singular attribute is a singlevalued attribute.

In fact it is not necessary to distinguish between two rotations if they differ only in permutation of $(x_{i1}, ..., x_{ik})$ or $(x_{ik+1}, ..., x_{in})$. Speaking of attribute rotation we will assume distinct rotations only.

Examples. ⁕ [STA]$_{85}$:TODO examples from DM1 lectures

How to reduce a sequence-sort range of attribute to a simple sort range.

As was said above when practising database design it is useful to employ a notion of HIT-attribute ratio. The ratio mirrors the respective singular and/or total constraint of a HIT-attribute and of the reversed HIT-attribute. Now we can say in other words (and maybe better) what a reversed attribute is: Let S, T be sorts from **BS** or sequence sorts constructed over **BS**. The reversed attribute A' is a special rotation of a given attribute A, such that while A is of type

$((PossibleWorld, Time) \rightarrow (S \rightarrow T))$

or

$((PossibleWorld, Time) \rightarrow (S \rightarrow (T \rightarrow Boolean))),$

the A' is of the type

$((PossibleWorld, Time) \rightarrow (T \rightarrow S))$

or

$((PossibleWorld, Time) \rightarrow (T \rightarrow (S \rightarrow Boolean))).$

Examples. ⁕ [STA]$_{86}$:TODO examples from DM1 lectures

Admissible rotation. A rotation of an attribute A is called admissible rotation, iff $A \leftarrow rotA$.

Thus each admissible rotation rotA of an attribute A is informationally equivalent to A. If rotA is not admissible, then rotA is not informationally equivalent to A.

An important question is what are the conditions causing a rotation of an attribute is admissible one.

One condition is: Each plural rotation of a plural attribute is admissible. With singular rotations or singular attributes it is not so easy. The problems come from the singularizer definition an from the possible partiality of attributes.[29]

The fact that a singular rotation of a given attribute A is admissible cannot be derived by no formal procedure from the construction of the attribute A. It is simply an empirical finding out. Studying admissibility of an attribute rotation we determine the ratio of the attribute.

Examples. ⋇ [STA]$_{87}$:TODO examples from DM1 lectures

Rule 6 (Admissible Singular Rotations Examination) *Modeling a domain each proposed HIT-attribute have to be examined in all its (distinct) rotations whether there is an admissible singular rotation or not. If admissible singular rotations do exist, one of them will be recorded by its construction into the* ***Attr*** *set of created model and the rest singular rotations are recorded as singular consistency constraints by their constructions into the* ***CC*** *set.*

Trivial derivation. Rotation of attribute is a special case of the so called trivial derivation. Another special case is: Having in mind that for any attribute A and any state of world $w \in (PossibleWorld, Time)$ the ${}^0A_w$ is simply a table, the second most simple derivation from attribute A is pointing one or more columns in this table, exclude it/them and take the remaining columns as ${}^0B_w$ of a new attribute B.

Thus trivial derivation can be defined as follows: An attribute B is trivially derived from an attribute A iff

- $B \leftarrow A$, i.e. for all $w \in (PossibleWorld, Time)$ ${}^0B_w = [{}^0f\,{}^0A_w]$
- the surjection f contains at most identity function, admissibly used singularizer and/or existential quantifier.

[29] I will provide you with more information in Appendix

Examples of trivial derivation. ⁎ [STA]$_{88}$:TODO examples from DM1 lectures

Decomposability and Subattributes. Having a conceptual model (**BS**, **Attr**, **CC**) we will aim to minimize the set **Attr** and to simplify particular attributes. The former can be realized by using the concept of definability. The latter needs a concept of decomposability. To obtain elementary attributes we decompose complex attributes into simpler ones, so called „subattributes" in such a way that informational capability remain the same. This means that the decomposition operation is an operation preserving information capability. The process of decomposition of attributes of a given conceptual model is analogous to the process of reaching normal forms in relational database theory.

Seeking the simplest form of a conceptual model is important from the Service Systems modeling and execution point of view: real world situations are complex enough itself and it is not wise to add more complexity by unnecessarily complex descriptions of the underlying semantics structures.

Subattributes. Let's have a HIT-attribute of a type*

$(PossibleWorld, Time) \to ((R, S) \to T)$,

where R, S, T are sorts from **BS** or tuple-sorts constructed over **BS** using sequence constructions, and let A be any of its admissible rotations.

Let variable w :: (PossibleWorld, Time), x :: R, y :: S, and z :: T if A is singular rotation or z :: $(T \to Boolean)$ if A is plural rotation. Then a construction of A is

$^0A = \lambda w \lambda x y \iota z([^0A_w(x, y)] = z)$.

An attribute B will be called a *proper plural subattribute* of a given attribute A if it is constructed by

$^0B = \lambda w \lambda x \lambda y \exists z([^0A_w(x, y)] = z)$.

An attribute C will be called a *proper singular subattribute* of a given attribute A if it is constructed by

$^0C = \lambda w \lambda x \iota y \exists z([^0A_w(x, y)] = z)$.

Proper plural subattributes and proper singular subattributes are called proper subattributes.

An attribute D is called a subattribute of a given attribute A if it is either A or B or C.

We precise here the notion introduced in paragraph Trivial derivation.

Obviously, a subattribute of an attribute A arises from A by a trivial derivation.

Examples: ⋇ [STA]$_{89}$:TODO from DM1 lectures

Decomposability. An attribute A is decomposable into attributes $B_1, ..., B_n$, denoted $A\diamondsuit(B_1, ..., B_n)$, iff

1) each of $B_1, ..., B_n$ is a proper subattribut of A, and
2) $A \leftarrow \{B_1, ..., B_n\}$.

The following claims are immediate consequences of the definition:

(1) If $A\diamondsuit(B_1, ..., B_n)$ then $A \approx (B_1, ..., B_n)$.

This is the information capability preservation of the decomposition operation.

(2) If $A\diamondsuit(B_1, ..., B_n)$ then each of admissible rotations of A is again decomposable into attributes $B_1, ..., B_n$.

An important question is how to recognize whether a given attribute is decomposable or not. This is important in practice, as we need to find the simplest form of database or alike memory for the implementation and subsequent maintenance.

Obviously, an attribute A of the type

(a) $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow (S_1, ..., S_m)))$

or

(b) $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow ((S_1, ..., S_m) \rightarrow Boolean)))$,

is decomposable if in all states of affairs $w \in (PossibleWorld, Time)$ its extension $^1(^0A_w)$ do not depend on some of the arguments of the function

$((T_1, ..., T_n) \rightarrow (S_1, ..., S_m)))$

or

$((T_1, ..., T_n) \rightarrow ((S_1, ..., S_m) \rightarrow Boolean)))$.

Examples: ⋇ [STA]$_{90}$:TODO examples from DM1 lectures

Comment: Let $A\Diamond(B_1, ..., B_n)$ and let f be the surjection which computes (in each state of affairs w) values of A from values of $B_1, ..., B_n$, i.e.

$^0A_w = [^0f(^0B_{1w}, ...,^0 B_{nw})]$.

Then the surjection f contains conjunctions and identities. If, moreover, A is a singular attribute then f contains a singularizer in addition.

A practical procedure to find whether a given attribute A of complexity 3 or more is decomposable one is based on the so called decomposition statement which you can find together with its proof in Appendix. The procedure has two variations. The first one is connected with single-valued attributes, i.e. with attributes given in singular rotation, and the second one is connected with multi-valued attributes, i.e. with attributes given in plural rotation.

The process applicable to multi-valued attributes is applicable to the single-valued attributes, too, but in practice it is easier to find an admissible singular subattribute of an attribute given in a singular rotation. That is why the decomposability seeking procedure has the two variations.

Decomposability seeking procedure.

1. Take an attribute A, $^0A \in \mathbf{Attr}$, with complexity $n + m \geq 3$.

2. Examine if there exists an admissible singular rotation of A. If yes, for each such singular rotation continue with item 3. If not, continue with item 8.

3. Examine each of the singular rotation proper subattributes with the same range in order from subattributes of complexity 2, then (if they exist) of complexity 3, etc., according to following steps:

4. Is the reality such that singular subattribute is the proper expression of? Note that this is an empirical discovering of the given domain.

5. If not, continue with another proper subattribute.

6. If yes, let B_1 is this proper singular subattribute; construct a new subattribute B_2 such that the function $^1(^0B_{2w})$ has the same domain as the function $^1(^0B_{1w})$ and the range of $^1(^0B_{2w})$ contains all remaining sorts of attribute A which are not included into subattribute B_1. Then $A\Diamond(B_1, B_2)$ and the attribute A is decomposable.

7. No decomposition possibility was found using admissible singular subattributes; continue with the next item.

8. For each plural rotation continue with item 9.

9. Examine each of proper subattributes of the attribute A with the same range in order from subattributes of complexity 2, then (if they exist) of complexity 3, etc., according to following steps:

10. Is the reality such that the subattribute gives just the same values as the examined plural rotation of A? Note that this is again an empirical discovering of the given domain.

11. If not, continue with another proper subattribute.

12. If yes, let B_1 is this proper subattribute; construct a new subattribute B_2 such that the function ${}^1({}^0B_{2w})$ has the same domain as the function ${}^1({}^0B_{1w})$ and the range of ${}^1({}^0B_{2w})$ contains all remaining sorts of attribute A which are not included into subattribute B_1. Then $A\Diamond(B_1, B_2)$ and the attribute A is decomposable.

13. If you do not finish in item 12. with decomposability of A, the attribute A is undecomposable.

Examples: ⋇ [STA]$_{91}$:TODO examples from DM1 lectures. Some examples for exercising.

Proper Conceptual Model. The notions of definability and decomposability allow to define a *data kernel* of a conceptual model.

For the sake of convenience I will extend the execution operation (see the formal paragraph 2 in section 4.3) from sequences to sets: A finite set of constructions **A** can be represented by each permutation of its elements which in turn is a sequence **A'** of its elements. Than ^{1}(**A'**) which in fact means execution of each construction in the sequence **A'** can be taken as execution of the set of constructions **A**. Thus ^{1}**A** will denote execution of each element of the set **A** which yields a set of objects being constructed by particular elements of the set **A**.

Now we can define: (Data) Kernel of a conceptual model (**BS**, **Attr**, **CC**) is a set of attribute constructions **Kernel** for which the following conditions hold:

(i) ^{1}**Kernel** $\approx$ ^{1}**Attr**,

(ii) each attribute of ^{1}**Kernel** is undecomposable,

(iii) there exists no such attribute $A \in {}^1$**Kernel** that $A \leftarrow {}^1$**Kernel** $- A$

The set of consistency constraints **CC** (rules of well formed model of a part of reality) can be transformed by substituting attribute constructions of **Kernel** for attribute constructions of **Attr** in respective consistency constraints constructions from **CC**. Executing this we obtain a set **Rules**.

Thus we can conclude this discussion with defining a *proper conceptual model* as a triple (**BS**, **Kernel**, **Rules**), where

- **BS** is a focused base of sorts,
- **Kernel** is a data kernel of the set of attribute constructions **Attr** designed by a data modeler and considered to be relevant and complete for the given purpose, and
- **Rules** is a set of consistency constraints constructions relevant to a modeled part of reality and formulated in terms of attribute constructions from **Kernel**.

4.4.3 Conceptual Systems and Conceptual Models

Propositions generated by an attribute. Let's compare now what we can express about a domain using appropriate conceptual system and its hierarchy of languages to what is possible to express about the domain using its conceptual model. The first case means expressing situations by means of propositions (constructed over the epistemic base), the second case means expressing situations by means of HIT-attributes (constructed over the base* of sorts). In the first case a message is constructed by constructions constructing $((PossibleWorld, Time) \rightarrow Boolean)$-objects, in the second case by constructions constructing $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow (S_1, ..., S_m)))$-objects or $((PossibleWorld, Time) \rightarrow ((T_1, ..., T_n) \rightarrow ((S_1, ..., S_m) \rightarrow Boolean)))$-objects, where $T_1, ...T_n, S_1, ..., S_m$ are sorts of some base*.

The first case is 'a world' of usual business documents, namely value propositions when speaking of Service Systems. The pluses and minuses are: (+) easy to write using NL for the message producer, (-) problems on the message receiver part–ambiguous expressions, misunderstandings.

The second case is 'a world' of proper specifications. Evidently the minus of this approach is it is not so easy for the message provider to write or to express the message properly. Will it be compensate by the pluses?

I will start with an example. Let A be a HIT-attribute 'SalaryOfEmployee'[30] which is given by a construction

$^0A = \lambda w \lambda x \iota y([^0 SalaryOfEmployee_w x] = y)$,

where w :: (PossibleWorld, Time), x :: (#Employee), y :: (Salary). The formalized name of this attribute is:

(Salary) of given (#Employee) / 0,1:0,M.

In each state of affairs w the attribute name denotes (and the construction ^0A constructs) a two columns table where in the first column there are particular employees represented by their names and in the second column there are numbers representing respective salaries. Taking our's pick we can read the row in the table:

'Salary of Ms. Smith is 8000 USD', e.g., or

'Salary of Mr. Brown is 15400 USD'.

These expressions are propositions decreasing by their truth-value True the uncertainty of the world.

But watching the table we can say more, for example:

'There is an employee having a salary greater than 15000 USD.'

Such sentences we will call propositions generated by the HIT-attributes. The first two propositions will be the so called basic propositions generated by a HIT-attribute, the last one is not basic; it is in a way derived from these basic propositions.

Basic propositions generated by an attribute. Let A be an attribute of type (a) or (b), i.e.

(a) A / $(PossibleWorld, Time) \rightarrow (T \rightarrow S)$, or

(b) A / $(PossibleWorld, Time) \rightarrow (T \rightarrow (S \rightarrow Boolean))$,

where $T, S \in \mathbf{BS}$ or they are sequence constructions over sorts from **BS**.

Let w :: (PossibleWorld, Time), x :: T, y :: S in case (a) and y :: $(S \rightarrow Boolean)$ in case (b), p :: ((PossibleWorld, Time)$\rightarrow$ $Boolean$), i.e. p ranges over propositions.

[30]we mean a month salary

The *set of basic propositions generated by the attribute A in the state of world w* is defined by the linguistic definition

$BP(A)^w = {}^1(\lambda p(\exists x(\exists y([{}^0A_w x] = y \wedge p = {}^1(\lambda w[{}^0A_w x] = y)))$.

This means $BP(A)^w$ is such a set of propositions p that there exist $x \in T$ and $y \in S$ or $y \in (S \to Boolean)$ such that (x, y) is a row of the table ${}^1({}^0A_w)$ and the p is the proposition which announces that $[{}^0A_w x] = y$ is true in the state of world w.

Note the using of execution constructions in the $BP(A)^w$ definition; they are necessary as we cannot put the identity symbol between an object and its construction. In other case we would mix the *use mode* with the *mention mode*. Indeed the $BP(A)^w$ is *mentioned* while the construction enclosed in the execution construction ${}^1(.....)$ is *used*. Mixing use mode with mention mode without explicit control is the most confusing mistake in descriptions of complex issues. And this occurs often within structure or behavior descriptions of a part of business reality.

Information capacity of HIT-attributes. By information capacity of an attribute A in a state of affairs $w \in (PossibleWorld, Time)$ we will call the set of all propositions generated by the attribute A in the state of world w, i.e. the set of all logical consequents of the set of basic propositions generated by the attribute A in w:

$P(A)^w =^1 [{}^0Cn^0BP(A)^w]$.

Information capacity of a set of attributes $\{A_1, ..., A_n\}$ in a state of affairs w is defined as a set of all logical consequences of the set of all propositions generated by attributes $\{A_1, ..., A_n\}$, i.e.

$P(A_1, ..., A_n)^w =^1 [{}^0Cn^0(\bigcup_{i=1}^n P(A)^w)]$.

A pragmatic conclusion: At the previous section 4.3 we started with understanding the world through NL. We built up a notion of conceptual system and the hierarchy of languages of this conceptual system. All this represents enough powerful gadget to understand deeply and unambiguously what is encoded in NL sentences describing particular business situation, e.g. value proposition of a given Service System. Even though the gadget is enough powerful, it is completely unusable in practice. We cannot analyze all expressions in value propositions up to epistemic base. But the previous considerations gives us possibility to simplify the analysis and still preserve advantages of the conceptual system and its languages hierarchy. It is known that a plausible requirements outline[31] contains a chapter 'Terms Used' defining terms of the domain under discussion. I propose here to extend this glossary to a conceptual model. A procedure is the following:

- We select a base* of sorts and write down the sorts definitions. This way we cover the usual glossaries and we obtain the set **BS**.
- Analyzing stories about a business of the domain under discussion we step by step fill the set **Attr** of constructions of HIT-attributes which grasps all important connections in the domain.
- In parallel with the **Attr** set construction we record all relevant constructions of consistency constraints which rule the game within the domain. Thus we obtain the set **CC**.
- Using definability and decomposition of HIT-attributes we create a data kernel, i.e. the set **Kernel**.
- Reformulating consistency constraints from **CC** in terms of **Kernel** we produce the set **Rules**.
- Now we have any relevant information of the domain under discussion nicely pigeon-holed in clear kinds of pieces of information.
- Propositions exactly and unambiguously describing situations in the domain will be propositions generated by the set of attributes **Kernel** or propositions derived from the set **Rules**.

The result of this procedure is in practice achievable and enough exact to avoid confusions arising from inappropriate formulations on the side of message producers and misunderstandings on the side of message consumers.

In fact we build this way a conceptual system of the domain together with its language.[32] The conceptual system is a set of primitive and derived concepts, i.e. closed constructions. The fact that dealing with constructions is formalized in the language of constructions based on typed lambda calculus entitles us to believe the concepts can be worked out within artificial agents.

[31] see, e.g. Alistair Cockburn book Writing Effective Use Cases [**?**]

[32] More precise explication of HIT-attributes and basic sorts in connection with conceptual system over epistemic base you can find in [4] and some remarks to this issue in Appendix.

The conceptual model in a way claims the conceptual system we are using in the given domain to mention all its elements and connections. We can say:

Statement 4 (Plausible Conceptual System/Model Relationship.) *Conceptual model of a domain is pragmatical anchor of the conceptual system in use together with its appropriate language.*

Thus I am sure the understanding a domain under discussion through HIT-attributes is worth in process of Service System modeling and services execution.

Binarization principle. Having a domain there is more than one possibility to describe it in a conceptual system. So as an object can be constructed by more than one construction we can describe one and the same domain in two or more distinct conceptual systems. Till now we used a conceptual system arising from the HIT method of conceptual modeling. This means we have at disposal constructions of HIT-attributes. A question is what will happen if we restrict the possibilities of HIT-attribute forms. Typically it could be restriction to less complex attributes only.

In section 4.3 I argued that it is possible to have more conceptual systems in which a given domain can be described and perceived. These conceptual systems differ in sets of primitive concepts: what is a primitive concept in one conceptual system could be a derived concept in another conceptual system. Namely the conceptual system in which a domain expert perceives the domain will differ from the conceptual system in which a beginner or learner perceives it. The same is valid when comparing conceptual systems in which two experts from distinct professions perceive the given domain.

Thus there are permanent differences in perceiving a domain between human agents of a Service System. Let's add artificial agents to this game. Some of the artificial agents within a Service System will be a primitive one, only just to possess the property to be an agent, some of them could be more developed, with higher capabilities. Again the conceptual system used by the primitive agent will differ from the conceptual system used by the more developed one. Working together these agents (both–the humans and the artificial) have to transform from one conceptual system into the other. Misunderstandings within these transformations cause fallacies in agents communication and result in bad Service System operating.

That is why we need procedures which help us and principles which lead us to perform conceptual systems transformations preserving information capacity and information capability.[33] I don't know what all such procedures and principles could be. I think this is a nice open research area which could bring a pretty results for practice. But I would like to present one such principle as a pattern. The conceptual system in which we perceive a given domain is pragmatically captured by conceptual model built up from HIT-attributes. The principle says that the same can be expressed by HIT-attributes of complexity 2 only. This is the binarization principle and I proved it first in 1985.

Using attributes of complexity 2 only means we use a kind of binary data model. The conceptual systems transforming procedure will be about transforming of HIT-attributes of complexity n, where n is arbitrary natural number, into HIT-attributes of complexity 2 only. Such attributes are easily implemented in various kinds of existing SW.

The reduction of attribute complexity to complexity 2 only is balanced by extension of the base* of sorts by the so called *concatenated sorts*. Then attributes of complexity greater than 2 are replaced by sets of attributes of complexity 2. The last step is reformulation of consistency constraints in terms of the new set of attributes. Of course this procedure applies to non-decomposable attributes.

Let's take an example from maintenance engineering. An important factor here is the „standard of material consumption for a maintenance intervention on a kind of device". This could be described by a HIT-attribute

A = Standard (Amount) of consumption of given (#Material) for given maintenance (#Intervention) on given (#Device category) / 0,1:0,M

This attribute is not decomposable as the amount depends on the material, on the intervention and on the device category. (By the 'intervention' we mean 'kind of intervention' not a a particular action in a given time point, here.) No singular subattribute is possible to find in this connection. The complexity of this attribute is 4. So, we introduce a concatenated sort

(@Norm) = concat(#Material, #Intervention, #Device category, Amount)

which contains these quadruples of cartesian product

(#Material, #Intervention, #Device category, Amount)

that are rows of the table ${}^1({}^0A_w)$, $w \in (PossibleWorld, Time)$. The definition of this concatenated sort is just the attribute A itself. By this step

[33]Attentive reader feel that information capacity of a set of propositions or information capacity of a set of HIT-attributes has a close relation to information capability of a set of HIT-attributes; more about this you can find in Appendix.

we made a transition from previous conceptual system based on sorts

(#Material), (#Intervention), (#Device category), (Amount)

to a new conceptual system based on sorts

(#Material), (#Intervention), (#Device category), (Amount), (@Norm).

Now we substitute the attribute A by four HIT-attributes of complexity 2:

B1 = (#Material) of given (@Norm) / 1,1:0,M,

B2 = (#Intervention) of given (@Norm) / 1,1:0,M,

B3 = (#Device category) of given (@Norm) / 1,1:0,M,

B4 = (Amount) of given (@Norm) / 1,1:0,M.

This way the previous situation expressed by attribute A of complexity 4 is now expressed by four attributes B1, B2, B3, B4, each of complexity 2.

The situation is depicted on fig. TODO

⋇ [STA]$_{92}$:TODO fig. of ERD of the situation

Let's describe the binarization principle exactly.

Let the *first conceptual system* is anchored by conceptual model

$CM_1 = (\mathbf{B}, \mathbf{A}, \mathbf{CA})$,

where

$\mathbf{B} = \mathbf{E} \cup \mathbf{D}$ is a base* of sorts, **E** is the set of entity sorts, **D** is the set of descriptive sorts. Let $S_j \in \mathbf{B}$ be particular sorts;

A is a sequence of constructions of HIT-attributes constructed over the base* **B** being of arbitrary complexity,

$\mathbf{A} = (A_1, ..., A_k)$,

and let **A** be a data kernel;

CA is a sequence of constructions of consistency constraints associated with the set **A**.

Each attribute constructed by construction A_j from **A** is either of type

$^1A_j/(PossibleWorld, Time) \rightarrow ((S_1, ..., S_m) \rightarrow (S_{m+1}, ..., S_n))$

or of type

$^1A_j/(PossibleWorld, Time) \rightarrow ((S_1, ..., S_m) \rightarrow ((S_{m+1}, ..., S_n) \rightarrow Boolean))$.

Let the *second conceptual system* is anchored by conceptual model

$CM_2 = (\mathbf{N}, \mathbf{F}, \mathbf{CF})$,

where

$\mathbf{N} = \mathbf{E} \cup \mathbf{D} \cup \mathbf{R}$ is a base* of sorts, **R** is the set of the so called concatenated sorts created by the binarization procedure (will be explained immediately). Let $T_i \in \mathbf{N}$ be particular sorts;

F is a sequence of constructions of HIT-attributes constructed over the base* **N** being of the complexity 2,

$\mathbf{F} = (B_1, ..., B_h)$;

CF is a sequence of constructions of consistency constraints associated with the set **F**.

Each attribute constructed by construction B_i from **F** is either of type
$^1B_i/(PossibleWorld, Time) \to (T_1 \to T_2)$
or of type
$^1B_i/(PossibleWorld, Time) \to (T_1 \to (T_2 \to Boolean))$.

The binarization procedure consists in three steps:

1. $\mathbf{B} \Rrightarrow \mathbf{N}$, i.e. deriving a new base* **N** from the original base* **B**:

   - $\mathbf{N} = \mathbf{B} \cup \mathbf{R}$;
   - The construction of **R** is the following: For all $A \in \mathbf{A}$ if ^1A is not of complexity 2 then ^1A is some ^1A_j (see above), where $n \geq 3$ and then put $concat(S_1, ..., S_n)$ into **R**, where
     $concat(S_1, ..., S_n) = {}^1(\lambda x_1 ... x_n(A_w(x_1, ..., x_m) * (x_{m+1}, ..., x_n)))$,
     where $w \in (PossibleWorld, Time)$ and * denotes function application or equality accordingly with plurality or singularity of the attribute ^1A.

2. $\mathbf{A} \Rrightarrow \mathbf{F}$, i.e. transition from complex attributes to binary attributes:

   - For all $A \in \mathbf{A}$ if ^1A is of complexity 2 then put A into **F**.
   - For all $A \in \mathbf{A}$ if ^1A is not of complexity 2 then (^1A is some ^1A_j, where $n \geq 3$, see above) denote
     R = $concat(S_1, ..., S_n)$
     and construct for all i = 1 .. n binary attributes ^1B_i such that
     $B_i = \lambda w \lambda r \iota x_i((x_i = r_{(i)}) \wedge (A_w(r_1, ..., r_m) * (r_{m+1}, ..., r_n)))$,
     where B_i and A are constructions of respective attributes, w :: (PossibleWorld, Time), r :: R, x_i :: S_i, i = 1 .. n, $r_{(i)}$ is a projection of r onto the component S_i, * denotes application or equality accordingly with plurality or singularity of the attribute ^1A.

     Put B_i into **F** for all i = 1 ..n.

3. **CA** $\Rrightarrow$ **CF**, i.e. transition from consistency constraints over complex attributes to consistency constraints over binary attributes:

   - For all $c \in$ **CA**, i.e. constructions over sets of A_j from **A**, do: substitute construction A over B_i for each A in c, where B_i are defined as in point 2;
   - Put the results into **CF**.

Following this procedure we make a transition from the first conceptual system anchored in conceptual model CM_1 into the second conceptual system anchored in conceptual model CM_2. A question is: what can be said about the information capability of CM_2 comparing it to CM_1? A good message is that it is the same.

Statement 5 (Binarization Theorem.) *For each conceptual model*

*CM_1 = (**B**, **A**, **CA**),*

*where **A** is a sequence of constructions of HIT-attributes (of arbitrary complexity) constructed over the base* **B**,*

there exists a conceptual model

*CM_2 = (**N**, **F**, **CF**),*

*where **F** is a sequence of constructions of HIT-attributes (of the complexity 2) constructed over the base* **N**, such that*

$^1\boldsymbol{F} \approx {}^1\boldsymbol{A}$,

i.e. the respective sets of attributes are informationally equivalent.

The above binarization procedure together with definitions of both conceptual systems is the procedure of construction of the conceptual model which existence is ensured by this statement. The only thing which need to be proved is

$^1\mathbf{F} \approx {}^1\mathbf{A}$.

This proof I postpone to Appendix.

The conclusion of these considerations is: No matter how complex is the reality in a domain, i.e. of what complexity will be attributes describing this domain, we can introduce concatenated sorts and than describe the domain using a network of sorts (entity or descriptive or concatenated ones) where each edge of this network represents a binary HIT-attribute.

Note that concatenated sorts are sorts bearing special constructions. These constructions construct connections between three or more sorts containing objects of order k in general. Thus the concatenated sort contains objects of order k+1 in general. According to the definition of higher order objects and the whole ramified hierarchy of types in the formal paragraph 5 both, the special constructions and the objects of connected sorts are objects of order k+1. This means that introduction of concatenated sorts, i.e. the transition into this new conceptual system being anchored by the binary HIT model represents a step up on the hierarchy of objects of higher order.

Now you can understand why the peculiar ramified hierarchy of types is unavoidable in our very practical thinking. Again here I want to remember that these logical roots are so much important not for the pure human communication only but for the communication within a Service System which is a composite of people, technology, artificial agents, another service systems, all „glued“ by shared knowledge and shared information.

Examples: TODO from lessons DM1 and DM2 (not DUM)

Another one point is worth to mention here: this is the terrible magic with execution construction and trivial construction during the previous explanation. Please, take it as an exercise in switching our minds between USE and MENTION modes of constructions. Let A be a construction. Then ${}^0A$ enables to speak of this construction as one speaks of any other object, i.e to *mention* the construction, while ${}^1A$ enables to speak of this object which is constructed by A, i.e. it enables to *use* the construction. Remember please the alternative solution of Russell's paradox (see Appendix and the comment in section 4.3). And prepare your mind to the next chapter about Diamonds where we utilize this mention-use switching very deeply.

Non-executable Modeling. Now we have at our disposal an apparatus which give us enough power to describe, i.e. to MENTION, what a given domain is about. Creating a conceptual model we start to understand the matter of facts within the domain.

What can we do with this description? We can read the conceptual model. We can develop it according to new circumstances. When new situations occur we can extend the model or we can modify the model. To USE such model is not a matter of the model itself. It is a matter of agents of some Service System.

A conceptual model shows what the things are. It tells us nothing about the behavior of a system under discussion. There is nothing what can be executed to see the change. That is why we will call the conceptual modeling to be a non-executable modeling.

Non-executable modeling is a very first step in our cognition process. In general all agents, the human ones and the artificial ones, need this first step. To perform any action with predictable result it is necessary to see the space the action will be performed in.

But we feel that it is not all. The most important issues of Service Systems are connected with their USE and thus with execution of something. That is why the following section is devoted to executable modeling, namely knowledge management as its very important part.

4.5 Executable Modeling and Knowledge Management

Service Systems are composites of agents (human or artificial or composed) and technology glued together by information and knowledge. We learned how to cope with information. Let's now focus attention to knowledge.

What is knowledge? I red lot of papers about knowledge engineering, knowledge acquisition, knowledge bases, knowledge management, knowledge mining, etc. but surprisingly nowhere I found a clear definition what the knowledge is. We have knowledge society, knowledge is taken as one of the most important assets in modern companies, we are living in knowledge economy, ... All this you can see or hear, but an answer to a simple question 'What is knowledge?' it is difficult to find.

Let's try to solve this problem naturally and in alignment with our conceptual modeling considerations. Conceptual model was defined as a triplet (**BS**, **Attr**, **CC**), where **BS** is a set of basic sorts, **Attr** is a set or sequence (it is not important to make difference between finite set and sequence for the purpose of conceptual model definition) of constructions of HIT-attributes over these basic sorts, and **CC** is a set (or sequence) of constructions of consistency constraints connected with attributes of **Attr**.

Conceptual model gives us a picture of modeled domain. But picture is only a picture: we can watch it but nothing else. Something is missing in our conceptual model to be really useful in our dealing with time-space and cyberspace. To perform any action, even a very simple one, it is not enough. Thus we need tools enabling actions. When I say 'we' I mean we–human beings so as we–all 'intelligent Service Systems'.

It is widely recognized that to memorize things or issues is important and helpful for any action to be performed. Our, human beings, functioning is fully dependent on memorizing. The same we can recognize when dealing with Service Systems. A level or 'intelligence' of Service System depends directly on its ability to memorize, to recall, and to use this what was memorized. A question arises: what is it? What are the 'things' or 'issues' which need to be memorized? And what is the appropriate form in which they will be memorized?

Any case the one who memorize is an agent.

Remember now the quotation from sect. 4.2: „To know something means that you can make predictions about it." Predictions are produced by memory prediction systems. Without memorizing no prediction can be made. Hence to know something, i.e. to have a knowledge, means first to memorize. Thus the knowledge belongs to cyberspace and as cyberspace is composed of concepts we can conclude: knowledge is memorized concept. On the other hand knowledge is inseparably connected to action: to say that something is knowledge requires there is a potential to use this something to make prediction. And as action cannot be executed without an actor knowledge is connected with actor, i.e. agent.

The basic property of agent is it can operate. This what we miss having only a conceptual model, i.e. non-executable model, is the possibility to operate. Thus, adding operations we turn the non-executable models into executable models.

Extending conceptual models into (executable) models we will precise what a modeling process is, what is its result and what tools help us in course of this process. These tools could be the enablers of agents intelligent behavior.

Being conceptually well prepared we can than explain what the knowledge is and what is its lifecycle within a Service System. The lifecycle of knowledge is a base of knowledge management. I will introduce here the co called cyclical paradigm which we use in our development teams and which I stand up against the classical application development templates.

This section I will finish with a contemplation on top-down and bottom-up approach to solve extremely complex issues like the brain functioning or Service System functioning and behavior. This will be a good motivation to the next chapter about the Diamonds Path Framework.

Chapter 5

Diamonds Path Framework

⋇ [STA]$_{93}$:DRAFT ALMOST FINISHED. PREPARE TO LANGUAGE CORRECTIONS.

My long-lasting ideas of models, modeling and modeling tools together with the idea of universal modeling presented in my PhD dissertation and my long-lasting opinion about co-operation based on deep business practice were very fruitful for acceptation of the ideas of Jeff Hawkins described in his excellent book „On Intelligence“. The Diamond path framework introduced in this chapter was discovered by me and my colleagues in the Mysroft project team before we have red this book. However, explication here is influenced by Jeff Hawkins ideas.

Cognition and modeling is about acquiring the space surrounding our current position. It can be in our physical world with its three dimensions so as in the world of concepts and ideas, messages–the cyberspace.

To be able to model, and thus to think on, such things that are not known till yet we need a kind of conceptual framework. Our natural languages are examples of expression of such a framework. But they are very redundant and ambiguous. Thus they help us in focusing our attention not very effectively. I propose here a new framework for this purpose. A framework which is very simple and unifying the natural diversity of our way of thinking. Moreover a framework proven in more than four year practice of development of solutions based on knowledge and information robots in my research team.

The framework consists of four basic models that are proven to help: (1) with understanding the space surrounding our current position in a situation, (2) with recognizing of principal movements in this space, (3) with organizing and planing possible actions in this space, and (4) with performing those actions. These four topics I call S-R-A-D scheme. This scheme can be interpreted as a basic imperative to an agent or Service System to be usable.

The „S“ means „see“ or maybe „watch“ what surrounds you.

The „R“ means „recognize“ what this watched is principally or „reason“ the movements, „reason“ the structures in the watched surroundings.

The „A“ means „advice“ to itself what a reaction could be, what behavior is the best in the situation.

The „D“ means „do“ or start to act and influence your environment.

Following sections will guide you through this S-R-A-D. The first one, called Diamond of Attention Focussing, helps you to understand what can be seen or watched by an agent. The second one, called Diamond of Cognitive Elements, give you a model of cognitive process an agent can use to recognize and to reason. The third one, called Diamond of Agent Team Organization, brings ideas how to organize a Service System to be able to advice itself in various situations. The fourth one, called Diamond of Predictive Behavior, is one possible model of acting and do it predictively. The fifth section, called Diamond of Diamonds–the Fifth Diamond, is about a synergy of those four Diamonds, about their practice utilization.

5.1 Diamond of Attention Focussing

At the beginning of the Diamond of Attention there was the integration of ideas belonging to two, seemingly remote, areas: the area of conceptual modeling and attempts to find a universal modeling tool for the work of analysts and data modelers [STDW], [SBJ97], [15], [16], and the area of modeling of cognitive processes and attempts to find a universal tool for modeling of cognitive agents [11]. The outcome of this integration was the specification of Diamond of Attention and implementation of the Diamond Modeling Tool. This tool can be used for modeling and simulation of any conceptual system, i.e., for example, for modeling of a conceptual system that an analyst aims to discover in the analysis of a given business area, or for modeling of the conceptual system of an agent simulating the behavior of a particular biological system.

The Diamond Modeling Tool was embedded into the first pilot implementation of a system belonging to knowledge and information robots class. This implementation was called UIR = Universal Information Robot.

TODO doplnit dle nasl. poznamek

ve SRAD je to to S

watch/see are key words

it is Diam of Attention but attention is not modeled directly

5.1.1 System Architecture Point of View

The Diamond of Attention Focussing (DoAF) is depicted in figure 5.1.

This figure can be explained in the following way:

If we want to get to know something, we have to focus first on individual objects (#Object) of the world we are getting to know. This world may be represented by anything: the material world around us, documents containing records about this world, but also the world of ideas, or the world of ants. To focus on an individual object means to differentiate it from everything else, to identify which phenomenal instances represent the same object, and finally, to be able to describe the properties of this object.

Objects alone are, however, usually less interesting for us than their particular connections (#Connection). These connections are, in a certain sense, stable, and may play the role of factual knowledge of these objects. Connections can be determined by a mere grouping of objects that shows visible stability (sequences of objects), or by a function that assigns objects of one type to objects of the same or a different type with a certain meaning, i.e. semantics.

If we want to work with objects and their connections in any way (record them, delete them, assign some properties to them, search for objects of given properties that participate in given connections, etc.), we have to introduce operations (#Operation). They are introduced, particularly, in order to be used. If we could only speak about operations, we could not use them for work. However, to speak about operations is also meaningful, for example, in order to clarify which operations are crucial for us.

But let us return to connections. We frequently speak about connections- we make different assertions about them. Further, we use connections. For example, the answer to the question „Which particular documents have been approved?“, is acquired by using the connection „documents that are placed into the category Approved documents“.

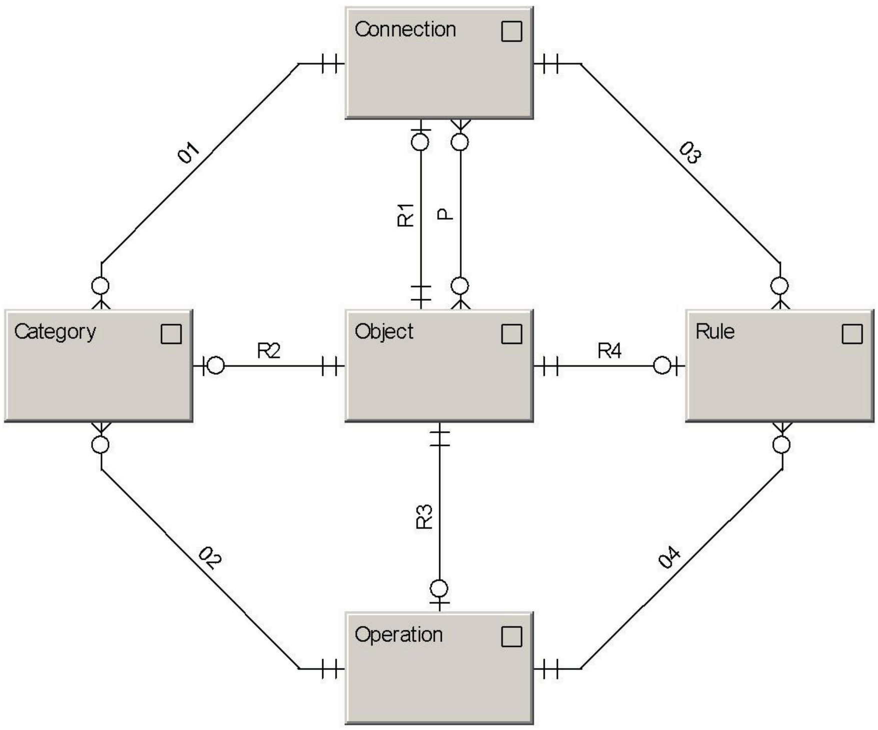

Figure 5.1: Diamond of Focus

Knowledge does not concern only individual objects, individual connections or individual operations. Knowledge concerns also our classification of individual items into categories (#Category), and our stating rules (#Rule) about it. Knowledge is that an object belongs to a category, or that a certain rule is valid for a certain category of objects. When we speak about a category, we speak about it as about any arbitrary object. But when we want to use a category, we have to know by means of which connection it was created (defined) and by means of which operation we can evaluate what belongs to the category and what does not belong there. Rules, similarly to categories, are determined by a specifying connection and an evaluating operation. The specifying connection describes, in the language of objects, what is the test carried out for the given model by a respective evaluating operation about. Further, when we speak about a rule, we do so as in case of categories: as about any arbitrary object. However, if we want to use a rule, then we use specifying connections and evaluating operations.

Anything that we want to mention is an object, i.e. an element of the container (#Object). Container (#Object) forms the center of the „Diamond Diagram“, cf. fig. 5.1.

Any ideational construct that can be used for arrangement of some elements, where an element can be anything on which we can focus, will be called a container. Elements can be inserted into the container as well as removed from it. Containers used in the Diamond will be defined by specification of elements that belong to them. Every element stored in DoAF that represents an objects of the examined reality (an entity) on which we are able to focus will be called a d-object. We will say that a d-object represents an object of the examined reality for the purposes of modeling.

Definition 7 (Object) *Container (#Object) is defined in such a way that it contains all such d-objects which can be mentioned in DoAF, i.e. which can be assigned certain properties.*

If the object on which we focus turns to be a connection or an operation or a category or a rule, and we will want to use this connection or operation or category or rule, then we connect it to (#Connection) or (#Operation) or (#Category) or (#Rule), respectively, by an edge R1 or R3 or R2 or R4. Then we say that this object from the container (#Object) represents this connection or operation or category or rule–cf. fig. 5.1.

Following definitions give precise meaning to DoAF vertices.

Comment 1 (Constraint for Objects) *An object may represent only one of the four mentioned possibilities: a connection, an operation, a category, or a rule.*

Definition 8 (Connection) *Container (#Connection) is defined in such a way that every of its elements is a sequence of the length n (n-tuple) of d-objects, where n is some finite natural number. Sequences in (#Connection) container differ in length, usually. Every element of the container (#Connection) is called a connection or a sequence.*

Definition 9 (Operation) *Container (#Operation) is defined in such a way that every of its elements is an algorithmically computable transformation of one state of DoAF to, generally, another state of DoAF. The elements of the container (#Operation) are called operations. The operation is this transformation, i.e. a pattern, not an instance of this transformation executed in a given time-point.*

Comment 2 (State of DoAF) *By the state of DoAF we mean one particular filling of DoAF (as a container) by elements-instances, that may dwell in DoAF.*

Definition 10 (Category) *Container (#Category) is defined in such a way that every of its elements has the following properties:*

1. *it is a container for d-objects,*
2. *it is one-to-one mapped to the pair* $\langle Cn, Op \rangle$, *where* $Cn \in (\#Connection)$, $Op \in (\#Operation)$, *and*
3. *it holds about the operation Op that by means of the connection Cn it can recognize whether a given object is or is not in this container.*

The elements of the container (#Category) are called categories. The connection Cn is called a defining connection of this category. The operation Op is called a defining operation or an evaluator of this category.

Definition 11 (Rule) *Container (#Rule) is defined in such a way that every of its elements has the following properties:*

1. *it is a d-object,*

2. *it is one-to-one mapped to the pair* $\langle Cn, Op\rangle$, *where* $Cn \in (\#Connection)$, $Op \in (\#Operation)$, *and*

3. *operation Op, by means of the connection Cn, carries out a test whether the rule is valid, i.e. the operation returns the value True if the test is successful, and value False in the opposite case.*

The elements of the container (#Rule) are called rules. The connection Cn is called a specifying connection of this rule. The operation Op is called a specifying (or testing) operation of this rule.

The edges 01 and 02 in figure 5.1 link a category with, respectively, a defining connection and an evaluating operation of this category. The edges 03 and 04 link a rule with a specifying connection and an testing operation of this rule. The diamond diagram contains, moreover, the so-called P-edges (projection edges) that link individual objects with a connection (sequence) that describes the given relationship between these objects.

The DoAF has to be provided with a set of operations in order to permit its utilization and modeling of world by means of it. These operations are divided, according to their semantics, into families of operations. We need a family of operations for creating, deleting, updating, making accessible and recognizing d-objects (CREATE, DELETE, WRITE, READ, OBTAIN, LABEL), a family of operations for „traveling" around the circumference of the diamond (GET_CONNECTION, GET_OPERATION, GET_CATEGORY, GET_RULE), and a family of operations for work with P-edges (projection of a connection onto its element/item, adding and removing an element/item to/from a connection, testing if an element belongs to a connection, etc.).

The deep description of those operations falls into technicalities and we will omit it in this conceptual level. However, there could be an approach to building service systems upon the DoAF directly. In such implementations of service systems the mentioned operations are utilized. ⋇ [STA]$_{94}$:This approach was proven in the first generations of the so called Universal Information Recorder system. In the following sections I will propose another way to build a service system which is not directly rooted in DoAF but takes advantage of this first diamond.

In the following text, we will discuss in greater detail those operations that make the DoAF a universal modeling tool. This topic falls in technicalities again and again those operations could be directly implemented when a direct DoAF-based implementation of service system will be chosen, only. But a deep understanding of those operations is crucial for comprehension of the following diamonds in next sections. That is why we will go through more detail description of them.

Edges R1, R2, R3 and R4 are jointly called R-edges. R-edges are used for execution of transition between mentioning (operation MENTION) and using (operation USE) and vice versa. If we stand on some particular object of the container (#Object), which we have mentioned somehow, then by the operation USE we cross, along the respective R-edge, to the represented connection (#Connection) or operation (#Operation) or category (#Category) or rule (#Rule), which we can then use directly. The particular operation is specified as follows (we will use the known Prolog clause denotation: „+“ stands for an input parameter, and „-“ for an output parameter):

Specification 2 (Operation use_obj(+Idobj,-Idnode)) *The operation* ***use_obj*** *is used for acquiring a d-object–an instance of a DoAF vertex (-Idnode) that is represented by the given object (+Idobj), i.e. for transition from the center of the diamond to the respective vertex. The operation fails if the given object does not represent anything.*

Operation USE may fail on objects belonging to (#Object) that are a „mere“ object and do not represent any category, operation, rule or connection. An example of such an object can be a concrete document, e.g..

If we are on one of the vertices of the Diamond Diagram, then the operation MENTION takes us, along the respective R-edge, to the representing object which we can then mention directly. Formally:

Specification 3 (Operation mention(+Idnode,-Idobj)) *The operation* ***mention*** *is used for for acquiring a representative object (-Idobj) of a given d-object that is an instance of one of the diamond vertices (+Idnode), i.e. for transition from the given vertex to the center of the diamond. If Idnode is an instance of the container (#Object), then Idobj will be unified with Idnode.*

Operation MENTION applied to an object belonging to (#Object) therefore gives the same object without change. Thus, this operation cannot fail.

Operation USE applied to a connection belonging to (#Connection) permits to use this particular sequence (to find out what is connected to what, or possibly, to have the possibility to complete a record about a particular sequence). Formally:

Specification 4 (Operation use_con(+Idcon,-Idobj_list)) *The operation* ***use_con*** *returns a list of objects (-Idobj_list) of which the given sequence (+Idcon) is established.*

Similarly, the operation USE applied to a category belonging to (#Category) permits to use this category (to find out about its current content, or whether a given object belongs to it, or possibly, to add further elements to this category). Formally:

Specification 5 (Operation use_cat(+Idcat,+Arglist)) *Using a category, i.e. the operation* ***use_cat****, means to run an operation, linked with the category (+Idcat) by the relation 02. The arguments of this operation are the category used (+Idcat), a sequence, linked with this category by relation 01, and arguments listed in the list (+Arglist).*

Operation USE applied to an operation belonging to (#Operation) simply executes this operation. Formally:

Specification 6 (Operation use_ope(+Idope,+Arglist)) *The operation* ***use_ope*** *runs a given operation (+Idope) and gives it as its arguments the content of the list (+Arglist). Arglist may contain both instantiated as well as free variables. What happens in case of free variables in Arglist is determined by how the used operation was defined within the respective operation* ***create****.*

This approach gives us a full flexibility concerning what and how we can manipulate with objects in the diamond.

Finally, operation USE applied to a rule belonging to (#Rule) runs execution of a test that checks the validity of the given rule. Formally:

Specification 7 (Operation use_rul(+Idrul,+Arglist)) *Using a rule, i.e. the operation* ***use_rule****, means to run an operation, linked with the rule (+Idrul) by the relation 04. The arguments of this operation are the rule used (+Idrul), a sequence, linked with this rule by relation 03, and arguments listed in the list (+Arglist).*

Thanks to the universal construction of categories we can work with different types of categories. For example, we can work with categories specified by the list of their elements (the specifying connection/sequence of a category then contains directly the elements of the modeled category), or with categories that use their relationships to other categories for the evaluation of their own content, or fuzzy categories, or categories that are evaluated by means of neural networks, e.g.. Similar freedom can be applied in rules creation (#Rule).

Thanks to representative edges R1, R2, R3 and R4, we can mention any of the entities in fig. 5.1–i.e. to relate it with other entities, categorize it and fill its attributes. We can create connections, e.g. between categories (more accurately, between representative objects of these categories), or between categories and rules, we can categorize the categories, etc. This is the major difference when compared to the majority of modeling tools where the modeled world is strictly separated from the world by means of which we model. There is only one world for DoAF. Further, in DoAF it is possible to integrate models of various extent of abstraction and diverse purposes. Thus, we can create one model, and another above it which is used for tuning of the first model; and still another model above these two that learns from their interaction.

The Diamond of Attention Focusing can be used as follows: when we support execution of processes, we focus on the center of the Diamond. We record and evaluate information about objects on which we focus, and we do so by means of the diamond vertices. When we want to improve the model for support of process execution, we focus just on categories, connections, rules and operations, i.e. on the diamond vertices. Thus we may imagine that alternation of utilizing knowledge and developing knowledge is represented by regular „throbbing“ of the diamond, as if it were a heart of a living creature.

This is the first insight into the amazing world of ideas and concepts, i.e., to the cyberspace. Its benefits are:

- uniform way of representing and manipulating of objects without reference to their nature,
- massive utilization of self-reference,
- union of the modeled world and the modeling world into one world only,
- avoiding chaos by systematic using of MENTION and USE operations,
- abundant level of universality to cover and/or express any situation that can be met within a service systems life-cycle.

5.1.2 Modeling Point of View

Let's try to understand what an agent can see through the prism of the Diamond of Attention Focussing (DoAF).

Imagine a state in which our DoAF is filled by

- some connections, i.e. sequences of objects, mostly again (and recursively) representing other connections,
- a set of categories, i.e. containers enabling to sort and classify objects,
- some operations, i.e. in a way visualized transformations of one state of the model into another state,
- a set of rules, i.e. conditions of well played game.

Let our agent spots an object in this situation. To visualize the whole picture the agent must to connect this objects into known objects by an appropriate type of connection.

TODO

In this way the DoAF could be a model of such agent's memory enabling to store pictures of the world formed by both, the time-space and the cyberspace.

※ [STA]$_{95}$:TODO: Hawking's invariant representations - their development versus connections/sequences in Diamond of focus; memory-prediction model - its hierarchy of neo-cortex regions and the prediction mechanism and the logical implication; pattern matching and self-reference; fixing point anchoring the universal model

5.2 Diamond of Cognitive Elements

The origin of the Diamond of Cognitive elements is in re-thinking of widely used type approach, i.e. a type system utilization and maintenance, to making order in our thinking of the space we live in and we think of. The issues of counting on one universal type system are:

- the type system have to be unique one for a given domain to help us to make an order, i.e. to support an understandable hierarchy creation
- if we want the domain covers all the space, i.e. the whole real world (three spatial dimensions and one time dimension) together with the cyberspace (the space of information, knowledge, their representations, ideas, images, conceptions, visions) the problem will be which type system is the top one
- all attempts made to order all things into one hierarchy end in a crash; very unnatural consequences of such hierarchies arise.
- in each such case the new cognition brings problems with the till now accepted type hierarchy, very often; it press us to rewrite the hierarchy according to the new knowledge

Those issues could be simply and naturally overcame: *by rejection of the only one universal type system*. The problem is what helps us instead of type system with recognition „what is what“, i.e. what helps us to recognize the nature of a pointed object.

Speaking on „net“, for example, we could have in mind a computer network or a social network or a net of railways, a fisherman net, etc. Usually we say „it depends on context“ what of those meanings we have in mind. Thus context could be our solution. Value of each fact depends on a context in which the fact is evaluated.

One and the same fact can be certain in a different measure in particular contexts. An example is: A message written in tabloid is taken into account very differently from the situation when the same message is written in serious press.

Another point of view is: what attention we pay to the particular fact? For example 'I am seeing a car damaged by fallen tree' or 'I am seeing *my* car damaged by fallen tree'. Again the attention payed to a particular fact depends on the context in which we evaluate this fact.

Going through the space we are under a mass firing of perception elements. Our environment flows us with messages. Not all of them are on the same level of importance, no all of them are on the same level of certainty. Working with facts we obligatory have to cope with attention and with fuzzyness.

Those considerations led me and my collaborators [1] to the second pattern of service system construction formulation in the beginning of year 2007. The following subsection contains a system architecture view of this pattern. In the next subsection we will go through an explanation from the brain functioning point of view, again.

TODO doplnit dle poznamek

ve SRAD je to to R

principally it is about catching the semantics

about real work with semantics outside of our heads

5.2.1 System Architecture Point of View

Facts can occur in variety of forms: from very complex ones to very simple ones. It is natural that a complex fact is built over more simpler ones. Thus some elementary facts could exist. A very elementary fact is that an item belongs to a category. Knowing what property the category C materializes or expresses we recognize from the fact that item I belongs to category C that I has the property expressed by C.

But the fact itself, without relation to a specific situation or state-of-affairs, is not very interesting. This specific situation we use to denote „a context". The measure to which we believe a message depends on the context. As was mentioned above, a message written in tabloid is taken into account very differently from the situation when the same message is written in serious press, e.g.. The same can be applied to any elementary fact. In this case this means simply: to what measure of believe (or measure of certainty) an item I belongs to a category C with respect to the context K. Thus the elementary fact is not the simple membership in category but „*a measure of certainty in which an item belongs to a category*". This elementary fact we will denote as CI-connection.

The second point is a level of importance or relevance of a fact. A message (bearing fact) that Peter Hurt had accident will take our attention very

[1]namely Filip Prochazka

differently in two situations: in a situation we do not know any Peter Hurt or in a situation we have a relative named Peter Hurt. The fact itself, so as an elementary fact itself, cannot be used in thinking or service producing process. In each particular situation the use of a fact or an elementary fact depends on the context. To express what happens during a process of our perception, when facts are taken into account, we will use a term „manifestation“. The facts are manifested in contexts.

Note, that what was said about us (human beings) with respect to perception of facts and manifestation of facts holds for service systems, too. The reason is: a service system is a combination of people, technology, knowledge, and information. Not depending on what portion of such system is human one and what portion of it is artificial one, and not depending on how this proportion is changed during the service system evolution, the behavior of such a system must be driven by context dependent attention payed by the system to particular complex or elementary facts.

Having a context K the elementary fact that 'an item I belongs to category C with a measure of certainty m_{crt}' with respect to a context K is manifested in this context K on an attention level a_{lev}. The measure of certainty and the attention level are both subjective and can be referred to with respect to a particular agent, only. The whole manifestation construction we will call *elementary belief* of an agent. Figure 5.2 expresses the situation.

Speaking on elementary beliefs of agents we will omit the reference „of an agent“ or „of agents“, meaning there is a set of agents, not explicitly referred to, each agent of which possesses this elementary belief. A service system, being formed by agents, is full of elementary beliefs. These elementary beliefs are elements from which all necessary knowledge of the system and particular value propositions in various contexts are constructed.

To describe precisely what a manifestation of an elementary fact in a given context, i.e. an elementary belief, is we use a conceptual model; figure 5.3 expresses the situation.

The center of the Diamond on figure 5.3 is an item to which we can focus our attention and we can speak on, think on, simply we can *mention* it. But not only we can do it. Anything else which is able to perform some mental operations (such as mapping, e.g.), which possess a kind of autonomous behavior, which posses a kind of collaborative behavior, and which is able to focus its attention, can do it, too. Let's we call anybody or anything being able to manifest the above described capabilities to be an **agent**. At least we, human beings, are agents. But we can imagine there are other agents than we.

Agents have their mental spaces. Speaking on agent's mental space we understand the part of cyberspace being at disposal to the agent in its/his/her memory. The word cyberspace we use to express a space of electronically or in our brains (or other way) recorded signals or messages, structures, rules, but at the same time our ideas, thoughts, and other similar things.

The agent's mental space can be extended by addition of new items so as it can be reduced by forgetting items.

Definition 12 (Item) *Container (#Item) is defined in such a way that it contains all points of the mental space of an agent; each such point can be assigned by a property; each such point can be focused on by an agent.*

Elements of (#Item) container are called items.

The previous Diamond of Focus has the (#Object) container to which anything which can be uniquely identified could be entered.

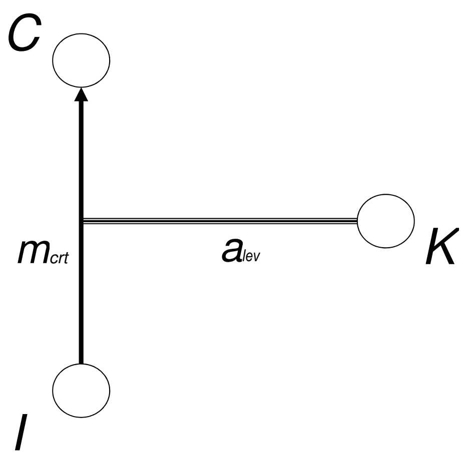

Figure 5.2: Elementary Belief: The „T“

Comment 3 (Item to object relationship) *The container (#Item) has the same content as the container (#Object) from the definition 7. This means it contains d-objects.*

Why do we such kangaroo jumping? What is the benefit from a new name for the same thing? The answer is: for didactic purpose. Intuitively we understand objects, we recognize them, and we manipulate them virtually in our minds or really in the real world. That is why we defined „objects" in the definition 7 in the section on Diamond of focus. But are there, in our real world, any such objects at all? The answer is not so simple as it looks for the first insight. I try to make an explanation of the nature of objects of our real world, as we perceive them, based on objectification process performed by any perceiving agent in the next section. That is why we need to use another term than „object" here.

The top vertex of the diamond on figure 5.3 contains categories. They were defined in the previous section in definition 10. This definition is based on an image of a container. A container contains elements, the elements can

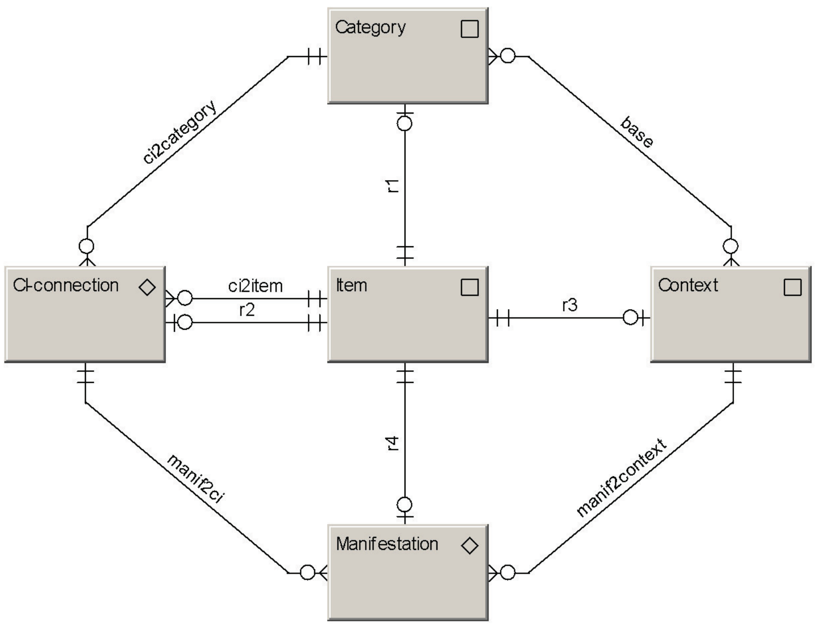

Figure 5.3: Diamond of Cognitive Elements

be added to a container or removed from a container. But are there any containers in our brains? Definitely not! Only synapses, i.e., very simple connections, we have at our disposal. What could be an image of category in terms of simple (binary) connections?

Categories are used to express that items have or posses features or characteristics. We can say: category invariantly expresses a particular feature of all items belonging to this category. Let's turn to Jeff Hawkins' invariant representations ⁕ [STA]$_{96}$:citace On Intelligence. Those invariant representations enable to recognize real world objects and to think of them. Invariant representations simply are categories. The fact that an item is from category can be expressed by a binary connection between the item and the respective invariant representation which materialize respective feature or characteristics.

Let's compare this considerations with George Lakoff's concept of category, namely prototype category. ⁕ [STA]$_{97}$:TODO, odkaz a kratky vyklad viz disertace, vzit angl Lakoffa a citovat s upravami dle me disertace k nasledujicim pojmum

basic level of categorization in the hierarchical taxonomy of categories

„human sized" categories

agents of service systems are human sized

categories depends on a way the agents interact with its environment

prototype categorization properties

graduated prototype effect

metonymy

radial categories

The understanding of category and categorization is crucial to service systems behavior and structure mentioning as well as using. An agent to be able to capture semantics of messages and their contexts within relevant contexts has to perform categorization on their own! And it has to do it in alignment with Hawkins's hypothesis on invariant representations. Thus we reformulate the definition of category in a dual sense to the previous one. The previous definition of category in section on Diamond of focus will be equivalent with the following two definitions.

Definition 13 (Category) *Container (#Category) is defined in such a way that it contains each marked item of the agent's mental space. Marked items are used as invariant representations by agents.*

Each element of (#Category) container is called category*.*

Definition 14 (CI-connection) *Container (#CI-connection) is defined in such a way that it contains each particular connection between a measure of certainty, an item, and a category, which meaning is:*

(measure of certainty) with which given (#Item) belongs to given (#Category) /0,1:0,M.

Elements of (#CI-connection) container are called CI-connections *or* elementary facts*. Denoting* I *the item,* C *the category, and m_{crt} the measure of certainty, the CI-connection is triple (* I, C, *m_{crt}).*

The domain of m_{crt} is the interval $\langle -1, 1\rangle$ of real numbers.

If m_{crt} = 1 than it is sure that I $\in$ C .

If m_{crt} = -1 than it is sure that I $\neg\in$ C.

If m_{crt} = 0 than nothing can be said about the fact that I $\in$ C.

If 0 < m_{crt} < 1 than I $\in$ C *with a measure m_{crt}.*

If -1 < m_{crt} < 0 than I $\neg\in$ C *with a measure $\mid m_{crt} \mid$.*

The bottom vertex of the diamond on figure 5.3 contains manifestations. Intuitively, manifestation is a kind of operation. To explicate what manifestation does, we need to understand a framework of this operation. According to the introductory considerations in this section the framework for manifestation will be a context–the remaining vertex of the diamond. The context brings a message to us making us acquainted with the situation and circumstances. The context is the tool making us capable to evaluate CI-connections or more complicated facts.

According to Diamond of Focus we have operations MENTION and USE that enable us to switch between the mention form and the use form of an object. The container (#Operation) in the Diamond of Focus contains possibly infinitely many of operations. A typical operation works on objects, i.e. it **mentions** those objects by **using** connections and/or other operations and/or categories and/or rules.

In the Diamond of Cognitive Elements we have only one operation, the manifestation. Comparing it to the previous situation within the Diamond of Focus, the difference is that we **use** (the **use form** of) a CI-connection by **using** (again the **use form** of) a context to manifest the CI-connection in the framework of that context.

My opinion is that this particular operation is this, what differs a proactive behavior of living beings from the reactive behavior of machines or other non-living entities. This is a hypothesis that at least helps us to understand capabilities of service systems in comparison to traditional computational systems or information systems.

On the other hand, what can be manifested in a framework of a given context is not the CI-connection, i.e. the elementary fact, only, but a more complex structures (facts) could be manifested. Intuitively, such complex structures or facts are again contexts. And similarly as in case of elementary facts manifestation, we would like to manifest these contexts in a framework of another context. Using the Hawkins' hypothesis on hierarchy of neo-cortex regions we simply take the manifested context as an item (a representation of this context for the purpose of mentioning it) and we will again manifest the CI-connection (expressing that the mention form of the manifested context belongs to a category) in a framework of this another context.

Having in mind the considerations of Jeff Hawkins in his book „On Intelligence" ⋇ [STA]$_{98}$:citace On Intelligence we can say: the context is a sequence of patterns. As there is a hierarchy of such patterns, i.e. each pattern is again a sequence of patterns, the context is a sequence of sequences of sequences ... But what are the leafs of this tree of sequences?

When **using** a context to bring us a message to make us acquainted with the situation and circumstances we **mention** names of patterns forming this context. Focusing our attention to a particular pattern (which is a sequence again, and thus it is another context) to understand to it, we **use** this pattern=sequence= context to bring to us a message to make us acquainted with the situation and circumstances, and again, we **mention** names of another patterns forming this sub-context. This recursive process continues in each particular branch of the sequence tree till the elementary fact, i.e. the CI-connection is reached, or till the next step is not necessary for understanding the starting context.

But this process is a little bit complicated. The mechanism of this recursive procedure is just the manifestation operation. The whole process is done not only on elementary facts, where measure of certainty is incorporated, but on elementary beliefs, where attention level is incorporated, too. Contexts in our heads, so as contexts dwelling in agents minds or in service systems, will differ if only they are distinct in levels of attention. Obviously they produce different results when used as bases for decisions.

The only way how we can visualize and evaluate facts, the elementary ones (CI-connections) or the complex ones (sequences/contexts) is exhibition of them by using the manifestation operation. Without manifestation there is no cognitive action. Manifestation exhibits a matter of facts about structures and/or rules driving a knowledge development.

The manifestation operation means that a focused elementary fact is exemplified within a given context or in the framework of given context. Moreover it assigns a level of attention to the object resulting from this manifestation in the framework of a given context. Manifestation answers the fundamental question: „What is it within the given context?“ And it answers by the returned object (the result of manifestation operation) and by a level of attention resulting from a computation made by the manifestation operation over attention spread through the context.

To describe the whole understanding of a context in its fascinating recursive nature the R-edges in the figure 5.3 have to be employed and the manifestation operation have to be understood.

Thus we have a problem from a pure deductive theory point of view: to define the manifestation operation we need to know what context is, and, to define the context we need to know what manifestation is.

Let's look the situation pragmatically. The intuitive image of context we have after the previous explanation is enough to understand first the preliminary definition of manifestation.

Definition 15 (Manifestation–The Preliminary Definition) *Container (#Manifestation) is defined in such a way that it contains each particular connection between a level of attention, a manifested CI-connection and a framework context, which meaning is:*

(level of attention) with which given (#CI-connection) is manifested in given framework (#Context) /0,1:0,M.

Elements of (#Manifestation) container are called manifestations *or* manifestation operations*.*

Denoting K *the framework context,* a_{lev} *the level of attention, and leaving the denotation from definition 14, the* manifestation *is a triple*

((I, C, m_{crt}), K, a_{lev}).

The domain of a_{lev} *is the interval* $\langle 0, \infty)$ *of real numbers.*

The inner triple (I, C, m_{crt}) *is an* elementary fact*, the outer triple, i.e. the manifestation itself, is called* elementary belief*.*

The level of attention expresses an amount of mental energy that an agent have to pay to the particular manifestation operation. There is infinite number of possibilities how this attention can be spread on the manifested object. It depends on the strategy of the given service system or its operational tactics which of those possibilities will be chosen. And this is close connected to the end picture or visualization that is produced by the manifestation operation execution.

The manifestation or elementary belief is this entity, which was depicted in fig. 5.2. Due to its shape we call it familiarly „the T". In chapter 8 on Service Systems and education we will see that it is close to the concept of „T-shaped professional" strongly emphasized in SSME.

A second step in the process of understanding manifestations and contexts will be done by the following definition.

Definition 16 (Context) *Container (#Context) is defined in such a way that it contains each marked finite sequence of manifestations. The term „marked" means that there exist an agent which marks this sequence. Each such marked sequence is used as by agents an invariant representation (in Hawkins' hypothesis sense).*

Elements of (#Context) container are called contexts*.*

Denoting particular manifestations $\mathrm{T}_1, \mathrm{T}_2, ..., \mathrm{T}_n$*, the context is the sequence*

$(\mathrm{T}_1, \mathrm{T}_2, ..., \mathrm{T}_n)$.

By definition the container (#Context) contains one special element $\perp$*, which is called* empty context*.*

Thus the context is nothing more than a special case of the connection/sequence from the Diamond of Focus. An additional distinction is that this sequence is marked. This means the pragmatical reasons drive the decision whether a sequence will be considered as context or not.

Note, that the T-s are usually interconnected in a context: those interconnections result from identities of *items* and/or *categories* and/or *contexts*. In this way contexts can form complicated networks not far from those networks formed by synapses in our brains. An example of such context network is on fig.

⋇ [STA]$_{99}$:slozeny kontext od Oskyho vlozit sem

Knowing better what a context is we can finish the manifestation definition, now. This is a third step in manifestation and context understanding.

Definition 17 (Manifestation–The Final Definition) *Container (#Manifestation) is defined in such a way that it contains each particular operation, satisfying the definition 9, such that it is a mapping*

(#CI-Connection) $\times$ *(#Context)* $\rightarrow$ *((#Context),* $\langle 0, \infty)$*)*

with the following properties:

- *Let* $K_1 = (T_1, T_2, ..., T_n)$ *be a particular context from (#Context),* (I, C, m_{crt}) *be an elementary fact from (#CI-connection). Then the resulting context after manifestation operation will be a context*

  $K_2 = (T_1, T_2, ..., T_n, T)$*,*

  where

  $T = ((I, C, m_{crt}), K, a_{lev})$ *is the elementary belief.*

- *The operation is 1-1 mapped to a HIT-attribute*

  (level of attention) with which given (#CI-connection) is manifested in given framework (#Context) /0,1:0,M,

  in such a way that a_{lev} *is the attention level on which the CI-connection* (I, C, m_{crt}) *is manifested in the framework context* K_1.

The (#Manifestation) container plays the role of (#Operation) container of the Diamond of Focus. Instead of many possible types of operations in case of Diamond of focus which instances dwell in the container (#Operation) we have only one type of operation, now. The instances of this operation differ in attention distribution, only. But all of them result in extension of the context which is the framework of the manifestation. The manifestation operation perform a kind of growing process. A given context grows by adding a new elementary fact. The visualization of this growing step depends on attention distribution. The way, how the attention is spread out on contexts in agents mental space, and the dynamics of its distribution, they both form the miracle of thinking. To find algorithms of attention distribution and to develop event-driven programs distributing attention is a great challenge in technology advancements of Service Systems.

My opinion is this will be the most productive area of Service Systems research and development. Now, we leave more details as technicalities interesting when technology portion of Service System is implemented but not necessary to understand the whole philosophy.

Note, that we still don't know how to start the cognition process and it is not clear enough how to cope with complex structures. Can we manifest something more complex then elementary facts?

Let's turn to the start point of a cognition of an agent. Having some context we can perform a manifestation operation; having at least one manifestation, we have at least one context. But how to create the first context or the first manifestation? And what of these both have to be the first?

Thus the fourth step to understanding manifestations and contexts is a definition of something like a fixing point:

Definition 18 (Elementary true) *Let* I *be an item,* C *be a category,* a_{max} *be a maximal attention level an agent could pay. Then the elementary belief*

((I, C, 1), ⊥, a_{max})

is called an elementary true.

Elementary trues are points fixing agent's mental space in the huge striking cyberspace. At least one such elementary true must be delivered into agent's memory to start the cognition process. All other things perceived by the agent will be cognized using the manifestation; the cognition is materialized in growing context or growing contexts.

In other words *elementary trues* are axioms in agents' lives. From these axioms, which anchor „the mind“ as fixing points, the cognitive process of an agent starts.

Not to be so „mathematically serious“ all the time, try to imagine you create an agent with completely separated contexts in its mental space. A kind of schizophrenia is modeled by this. I believe the agents (Service Systems) with particular strategies of their contexts grow will help to study psychotic disorders in future.

Having one elementary true we have a context. A very simple one, but we have! Having at least one simple context, we can make a first manifestation of a new fact in this context. To continue we need to employ the MENTION-USE mechanism based on R-edges and explicated in previous section 5.1 when Diamond of Attention Focusing was introduced.

In the previous paragraphs of this section we have explained what means *item*, *category*, *CI-connection*, *context*, and *manifestation* from the figure 5.3. At the same time we have explained meaning of edges *ci2item*, *ci2category*, *manif2ci*, and *manif2context* on this figure. They are simply projections of associative entity sets #CI-connection and #Manifestation to their components. To complete the explication of the Diamond of Cognitive Elements we have to describe the so-called R-edges *r1*, *r2*, *r3*, and *r4*, and the last edge named *base*.

However, the meaning of R-edges we know from the Diamond of Focus. Again, they are the representative edges, which are used for execution of transition between mentioning a particular object–here an *item*–(operation MENTION) and using a particular object (an *item*) (operation USE) and vice versa. If we stand on some particular object of the container (#Item), which we have mentioned somehow, then by the operation USE we cross, along the respective R-edge, to the represented *category* from (#Category) or *CI-connection* from (#CI-connection) or *context* from (#Context) or *manifestation* from (#Manifestation), which we can then use directly.

Operation USE may fail on *items* from the container (#Item) that are a „mere“ *items* and do not represent any *category*, *CI-connection*, *context* or *manifestation*. (An example of such an *item* can be a concrete port in a given computer network, e.g. Generally, such an example is anything which cannot be used to think! But it could be thought about!)

If we are on one of the vertices of the diamond diagram from fig. 5.3, i.e. we stand on a particular instance of (#Category) or (#CI-connection) or (#Context) or (#Manifestation), then the operation MENTION takes us, along the respective R-edge, to the representing *item* which we can then mention directly.

Operation MENTION when applied to an object belonging to (#Item) gives us the same object without change. Thus, this operation cannot fail.

The philosophy of MENTION-USE is completely transferred to Diamond of Cognitive Elements from the Diamond of Attention Focussing. The detailed description of those operations within the Diamond of Cognitive Elements context falls into technical specifications of concrete implementation. We will omit those details here as not necessary technicalities, because our aim is conceptual understanding of cognitive process modeling principles.

Thanks to representative edges *r1*, *r2*, *r3* and *r4*, we can mention any of the entities dwelling in Diamond of Cognitive Elements containers. Particularly this means an *item* in an *elementary fact* can be again a *context*. Thus we can **use** now the **mention form** of a context K_1 by **using** (the **use form** of) a CI-connection and we **use** (the **use form** of) this CI-connection by **using** (again the **use form** of) the other context K_2 to manifest this particular CI-connection (containing the categorization of the K_1 context) in the framework of that other context K_2. So, our intuition expressed above that complex structures or facts are again contexts is true and, similarly as in case of elementary facts manifestation, we can manifest these contexts in a framework of another context. And this looks like a confirmation of the Hawkins' hypothesis on hierarchy of neo-cortex regions. We simply take the manifested context as an item (a representation of this context for the purpose of mentioning it) and we manifest the CI-connection (expressing that the mention form of the manifested context belongs to a category) in a framework of this another context. By the CI-connection we express what the manifested context really is in the framework context. This is the fifth step in manifestation and context understanding process.

On the other hand those entities dwelling in the container (#Item) which turned to be a *category* or a *CI-connection* or a *context* or a *manifestation*, we can use to proceed in the cognitive process.

Again, here in fig. 5.3 we have the game with self-reference at our disposal. This is crucial to understand *contexts* and *manifestations*.

Using a sequence

$USE \rightarrow MENTION \rightarrow USE \rightarrow MENTION \rightarrow USE \rightarrow ...$,

in combination with manifestation of new elementary facts, to the appropriate depth in each branch of the context tree (cf. the text between definitions 14 and 15) we can imagine that the *context* can **„grow“** as a kind of **„semantic plant“** in a **„semantic garden“**. Such a *context* is this, what Jeff Hawkins (see ⋇ [STA]$_{100}$:citace On Intelligence) calls a memory region. At the same time it could be the whole memory of an agent or of a service system. My opinion is the semantics is nothing more that such a semantic plants garden. And thus we can cope with semantics in artificial agents.

The complexity of space, in which an agent (or a service system) lives, is hidden in *contexts*. Thus the context evaluation, which is performed during a manifestation operation, is a crucial topic. In alignment with Jeff Hawkins memory-prediction model (see ⋇ [STA]$_{101}$:citace On Intelligence) a simple invariant representations have to be used to context evaluating computations.

The section on Diamond of Cognitive Elements started with a list of issues of counting on one universal type system. We said that such one universal type system have to be rejected to overcome those issues. But a question arose: what could compensate missing type system? The answer is the last edge on fig. 5.3 named *base*.

Definition 19 (Context's Base) Base *of a given context is a set of categories connected to the given context by a relationship which meaning is:*

(#Category)-s forming a base of given (#Context) /0,M:0,M.

The term 'forming a base' means that this way constructed set of categories possess the property of type system for the given context and is used as a type system of this context.

Thus each item occurring anywhere in the context belongs to one and only one of categories forming a base of the context.

Remember that type system is a set of pairwise disjoint nonempty sets such that each item in focus belongs to just one of those sets.

It is true that: Each *context* has one and only one base. A *category* can be a member of bases of various *contexts*.

The base of context serves to answer an often question when an item is pointed: „What is it?“ The just finishing explication helps us to understand that the answer depends on context in which the question was asked.

—

Summarizing what was done by the second diamond introduction and comparing it with the first diamond–Diamond of Focus we can say:

- the first diamond's objects were reduced to simple points in an agent (or a service system) mental space,
- the first diamond's general connections were split into elementary facts, i.e. CI-connections, and sequences, i.e. complex facts or contexts,
- the first diamond operations were reduced to only one type of operation–manifestation–a particular instance of which implements a chosen strategy and/or tactics of attention spreading,
- the first diamond concept of category was turned to a marked item serving as the invariant representation for memory-prediction systems of agents and/or service systems, and
- the first diamond rules were spread between CI-connections, contexts, and manifestations.

I leave the last point as an exercise to an attentive reader to demonstrate this assertion is true.

5.2.2 Functioning Point of View

Memory-prediction model (see ⋇ [STA]$_{102}$:citace On Intelligence) helped us to understand what intelligence is and how the cognitive process works. To understand –from the cognitive point of view–how a service system works the Diamond of Cognitive Elements can be exploited.

The way from perceived items to their invariant representations is a means to catch semantics. Thus, when we need an agent or a service system can work with semantics we have to procure a model together with its execution engine being able to run through the way to invariant representations. Such a model could be the Diamond of Cognitive Elements (DoCE).

Let us imagine a state in which our DoCE is filled by

- some contexts, i.e. sequences of elementary beliefs, mostly again (and recursively) representing (manifestations of) other contexts,
- a set of categories, i.e. marked items being invariant representations itself,
- a set of CI-connections giving in a way contents of those categories together with particular measures of believe to which particular items belong to those categories, and
- some manifestations, i.e. in a way externalized visualizations of previously perceived items endowed with particular measures of attention.

Let our agent perceive an item in this situation. To understand what is the meaning of this item, i.e. what semantics he/she/it sees in it, the agent must turn this item into elementary belief, i.e., to manifest it in the context from which he/she/it perceives this item. The agent tries to recognize the item within embedded facts[2] categorizing item in the given context. A pattern-matching technique can be applied to find out whether the perceived item is known to agent or not.

If the item is found in a memory of contexts, i.e. the item matches some item in agent's memory stored context, it is recognized to be known. In other words, the agent knows its semantics.

If the item is not found, the same pattern-matching procedure is performed with those items of the given context that are sequences. Remember

[2]facts embedded in the elementary beliefs forming the context

that mentioning the item means working with the center–the node (#Item)– of DoCE and using the item in this particular situation means working with the vertex (#Context) of DoCE.

The process ends when either a pattern somewhere in the hierarchy of embedded sequences is recognized or the elementary fact– a CI-connection belonging to the vertex (#CI-connection)–is reached. Thus we can say, the recursive Prolog-like unification help us to obtain on leaves of the evaluation tree either known facts or elementary facts. An elementary fact is known when it is in the set of CI-connections stored in agents memory or it is not-known in other case.

The basic step of a learning process of an agent could be simply storing a new elementary fact to the agent's memory. In this way the DoCE could be a model of such agent's memory.

But what happens after the described pattern recognition process? Intuitively we see it is not all what have to be explained. There must be something more to empower us to say „yes, we have a consistent hypothesis of how the cognitive process runs". This remaining point is manifestation. Anything we can understand, so as anything an agent can understand, must be in a way visualized for the subject trying to understand. This visualization is performed by the manifestation operation–cf. the vertex (#Manifestation) of DoCE. Manifestation of some fact could be seen as a way of enrooting of perceived fact into a chosen context with selected level of attention.

In our brief excursion into transparent intensional logic (cf. ⋇ [STA]$_{103}$:ref to TIL section) we learned to distinguish between intensions and extensions. Each object in our space (the real three-dimensional time-space together with the cyberspace) we can focus on has its intensional part and its extensional part. The intensional part tell us what could be, what could exist, or what are the possibilities of the object variations in possible states of world. The extensional part tell us what is, what just exists, or what is the current variation of the object.

Manifestation is the operation giving us the extensional part of an object to the intensional one of the same object dependently of a state of world expressed by a context. But it gives not a self-existent visualization of the object in a given context. It gives a subjective visualization endowed with a level of attention payed by the recognizing agent.

Reading the book On Intelligence (see ⋇ [STA]$_{104}$:citace On Intelligence) and rethinking the concept of hierarchy of regions in neo-cortex described there together with prediction operation leading to invariant representations and to utilization of them in perceiving process, we can visualize the cognitive process using the model of DoCE, and moreover building an agent system based on this model we can simulate this cognitive process in a way.

⋇ [STA]$_{105}$:TODO: more explanation to cognition and service systems

⋇ [STA]$_{106}$:TODO: the following comments transform into a text

Plausible reasoning: induction and analogy

Use the G. Polya's book Mathematics And Plausible Reasoning (1954)

Demonstrative reasoning vs. plausible reasoning

Demonstrative reasoning: math and logic

Plausible reasoning: everything except math and logic and except stupid reasoning.

Objectification= how we construct objects in our minds using elementary beliefs and engaging the first Diamond.

5.3 Diamond of Agent Team Organization

The first two diamonds gave one possible answer to a question: „What the World is about?“ or „How we, human beings, can understand what is, and what happens, around us?“. More general we can see it that it could be answers to questions: „What the environment of an agent or a Service System is about?“ and „How the agent or Service System can comprehend semantics“. A Service System to be successful needs to understand its environment. This capability has to be present either in a form of „people heads“ or in a form of special technology supported tools. Namely the Diamond of Cognitive Elements could be a challenge of artificial intelligence problem solving model.

But to grasp the semantics is only one part of artificial predictive behavior. At least one other part must be explained. It is an ability to find autonomously advice in a situation; or to find „what is a right reaction to a happened events?“. My opinion is that this is about *organization* (I mean here organization as a process). When a man wants to do something he must organize himself, first. Without such a self-organization the man's behavior will look like a syndrome of serious disease. We all do this self-organization before each action, but we do it almost automatically. We do not think on it, thus, we do not know we do it, often. The same we do as individuals we do in teams when trying to reach some goal together. It is easier to think about and to describe the team organization than a self-organization. That is why we will follow knowledge and competence elements of project management, the project management processes, rules and techniques. And as the life (of a human being so as of a Service System) is not a question of one isolated project, we will speak from beginning of program (of projects) and/or portfolio (of projects and programs) management.

From the Diamond Path Framework point of view expressed by the S-R-A-D scheme it is now about the „A“–advice. The association chain is: Advice—advice itself—organize as I explained in the previous paragraph. ⋇ [STA]$_{107}$:TODO

This Diamond we discovered[3] at the end of year 2007 as a reaction to a pure practice problems we were faced to. This were problems with a team's attention on a long lasting program of projects–technologic, application, and organization ones. ⋇ [STA]$_{108}$:TODO

[3]Filip Prochazka discovered it and I made the first systematic definition of it

5.3.1 System Architecture Point of View

The inner organization of service system could be driven by the so called Diamond of organization. Figure 5.4 shows this Diamond.

TODO text TODO opravit obrazek DIAM3 – nazvy hran dle textu

Our approach will be a project management one. The reasons could be find in chapters 1 were introductory comments were made and 3 about Service Systems.

⋇ [STA]$_{109}$:TODO hlubsi vysvetleni a dat vice do souvislosti

Let's have a look at fig. 5.4 intuitively. This gives us an idea how to organize a team of Agents to achieve a Goals, and to do it sustainably.

Agents, human or artificial, in both cases autonomous and co-operative,

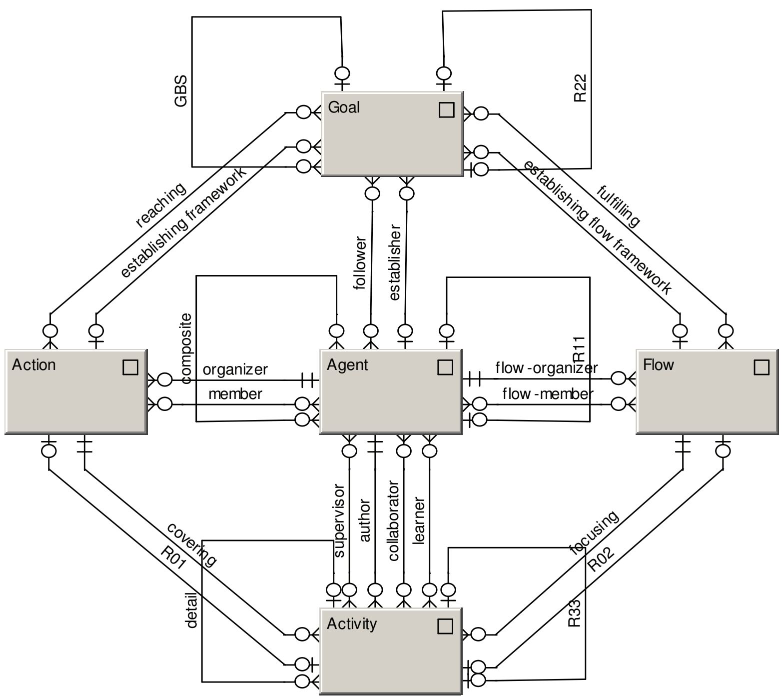

Figure 5.4: Diamond of Organization

act to perform some *Activities* in a role of *author* or *collaborator* or *supervisor* or *learner following* given *Goals* or *establishing* these *Goals*.

Actions cover some *Activities* into manageable tasks as building blocks of attention focusing to actions in a short or middle range time periods and filling up those *Goals*. Typical examples of actions are projects. An *Action* can *establish* new *Goals* if this is a more complex task. Portfolio o Program of projects belongs to this category. So, *Actions* are projects or programs of projects or portfolios of projects and programs, or they can be another composite activities.

Actions are organized or led by Agent-Project Manager (or Coordinator or Organizer) and they are worked out by Agents-Project Members.

Flows are new organizing structures not yet widely recognized in current project management methodologies. *Flows focus attention* to particular *Activities* from a long range time period point-of-view to reach continuity and sustainability in selected kinds of *Activities*. This selection is domain and/or strategic aims dependent. The relationships of *Flows* to *Goals* and the relationships to *Agents* are symmetric to analogous relationships of *Actions*.

We can say: ideas are materialized using *Actions* but they are cultivated to be sustainable using *Flows*.

Each *Activity* can be *decomposed* into detail *activities*. Each *Goal* can be breakdown into well known hierarchy–the *WBS*. A group of *Agents* can be organized into a *composite of Agents* and can be dealt with as with an *Agent*.

The vertical axis of this Diamond expresses what and why has to be done, while the horizontal axis expresses how all this is organized.

The whole game performed by *Agents* need to be organized in such a way that crossing boundaries of some context doesn't destroy the whole portfolio or program or project management build. In other words we need features and facilities to be able to recognize that two items being seen as two different entities in two contexts (i.e. often referred to in two separate models) are really only one and the same object. For this purpose we use in our model special links–*self-referential links*. Each such link expresses identity of connected objects and crosses boundaries of particular contexts, namely context of a given *Agent*, context of a given *Activity*, context of a given *Goal*. Further, an Action to an Activity link–denoted *R1* on fig. 5.4–, and a Flow to an Activity link–denoted *R2* on fig. 5.4–express the switch of context from the one, in which something is an *action* or *flow* to the one in which this something is recognized as *activity*, only.

We know this self-reference from previous two Diamonds, where the so called R-edges were introduced. In both, the Diamond of Attention Focussing and Diamond of Cognitive Elements, these R-edges serve to switch between **mention** and **use** forms/modes of particular objects. Switching between mention and use modes is in fact switching between contexts as we mention an object in another context then we use it. Thus the *self-referential links* here are in a way generalization of R-edges from the previous two Diamonds.

This self-reference is one of the most powerful concepts in project management at all. Partially, this concept is contained in the well known PDCA-cycle[4] and in lot of recommendations within ICB[5]. My experience is that expressing this concept more explicitly and more generally brings lot of simplifications in the management process. The management process is then build from unified items and it is easy to follow it, easy to maintain it and easy to develop it. Hence the management process starts to be a sustainable one.

Let's start with precise definitions. As the aim of this Diamond is to organize a team of agents, the agent is in the center position.

Definition 20 (Agent) *Container (#Agent) is defined in such a way that it contains each individual or object having a competency to execute some activities autonomously and co-operatively. This means it is possible to identify detectable behavior of this individual.*

The autonomy means ability to create plans and strategies and to manage execution of these (by itself prepared) plans and/or strategies. The autonomy of an agent means its ability to contribute to its own persistence.

Co-operativity means the individual is able to take part in a group of agents (similar individuals) which creates mutually acceptable plans and these plans jointly executes. The co-operativity of a group of agents means its ability to contribute jointly to their persistence as a whole.

Elements of (#Agent) container are called agents.

Moreover, agent can possess the property of compose-ability: agent can be composed of other agents which are its parts; agent can be a part of another agent which is the whole to which the first agent is a part. Thus agents can be so called holons (whole-parts) in a sense of Arthur Koestler approach.

[4]Demming's Plan-Do-Check-Act cycle

[5]IPMA Competence Baseline–a standard for Project Management from a competences point of view

A natural question is: what an agent can do? Thus the second important definition is the definition of activity.

Definition 21 (Activity) *Container (#Activity) is defined in such a way that it contains each (intangible) phenomenon of space-time or cyberspace such that*

- *execution of this phenomenon cause a change of some structures of this space, and*
- *the originator of this execution is an agent.*

Elements of (#Activity) container are called activities.

Thus a phenomenon to be an activity it must fulfill the following conditions:

1. *there exists an agent which causes or executes this phenomenon,*
2. *there exist structures in time-space or cyberspace recognizable and distinguishable from anything other,*
3. *when the phenomenon is executed the structures of the previous point are recognizably changed.*

The object from the (#Activity) container can be on one hand executed, i.e. it can be directly ***used****, and on the other hand this object can be planed and/or designed, i.e. it can be* ***mentioned****.*

Activities in this Diamond correspond to operations in the Diamond of Attention Focussing. Indeed, an activity is composed of one or more operations in a given order. The activity execution means ordered execution of the activity forming operations.

Agents execute activities to accomplish some goals or objectives or to deliver something.

Definition 22 (Goal) *Container (#Goal) is defined in such a way that it contains each such formulation of desired or required state of affairs that enables to recognize whether the required state is reached or not.*

Elements of (#Goal) container are called goals *or they can be called* objectives *or* deliverables.

If it is possible to measure values of parameters of desired state of affairs and determine a time-point of achievement of this state (i.e. the formulation is specific one) the formulation can be called objective *or* deliverable*. In other cases this formulation is simply* goal.

Goals are often specified so vaguely, that they are more about feeling than about reasoning. In such cases instead of goals we can speak of needs*. The decision to speak of needs is pragmatical only: the less definite formulations belong to needs; formulations expressing orientations belong to goals.*

The specific formulation can be called deliverable *if it expresses packaging of products or services into one separate delivery.*

A special type of objective *exists for projects: it is called* triple constraint*. Triple constraint has three types of parameters:*

- time–*which is a time of achievement of desired state of affairs,*
- costs–*costs necessary to achieve the desired state of affairs, and*
- parameters of accomplishment–*parameters describing the achieved state in another way then by necessary costs and necessary time (i.e. other parameters specifying a structure in time-space and/or cyberspace).*

A special type of objective *exists for Service Systems: it is called* Value Proposition*. The following claims specify in what circumstances the objective which formulation covers Clients benefits is called value proposition:*

- *the formulation includes* time parameter *specifying time dimension of achieving the desired state of affairs,*
- *the formulation includes* parameters of accomplishment *in the sense of triple constraint,*
- *the formulation includes* parameters of benefit *specifying clearly understandable benefits for the client.*

Note that this item, which proved to be a goal in one context could be an objective in another context or a deliverable in some other context. Metaphorically said this depends on a distance from which we watch the situation.

Till now we spoke about things as they are no matter if some organization is introduced. The first organizational element that helps to accomplish effectively and efficiently desired goals is the action or composite activity or in other project.

Definition 23 (Action) *Container (#Action) is defined in such a way that it contains each intangible wrapper of activities such that it is possible to decide univocally which activities are in the wrapper and which are not. The activities in the wrapper possess a certain structure of partial ordering and/or time ordering which has to be monitored in both cases, when they are and when they are not executed.*

Elements of (#Action) container are called actions *or they can be called* projects *or* composite activities. .

We speak of pure projects *if the wrapped activity sequence is*

- *unique,*
- *not repeatable,*
- *it has a clearly distinguishable start and end,*
- *it heads to one goal or to intersection of some goals, or it heads to one objective or to intersection of some objectives.*

In other cases we speak of composite activities *or simply* actions.

By the execution *of the project or a composite activity or an action we mean the execution of all its wrapped activities while the partial ordering structure and/or time ordering is respected.*

Comment 4 (Project/Composite Activity/Action polymorphism) *(#Action) is a base category and can be used to contain, and thus to describe, projects, processes (i.e. repeatable groups of activities with their ordering), sets of projects organized into programs or into portfolios as well as to describe only temporarily grouped activities.*

Current project management methodologies and standards end here and elaborate over till now defined concepts. Our experience was that attention payed by team members to particular elements of the whole program needs to be managed in a way, as in other case the whole process of program management turns to inefficiency. A question arose: how to employ the attention payed by particular team members to particular elements of the whole program systematically, comprehensibly and simply enough? The answer is the so called „flow“.

Definition 24 (Flow) *Container (#Flow) is defined in such a way that it contains each named long-lasting attention focussing to a particular topic resulting in a continuous attention flow which influences or can influence a space and/or time complexity of the agent's computational process, executed in a framework of some element of (#Action) container, by which an element of (#Goal) container is achieved.*

Each element of (#Flow) container is a category[6] *serving as intangible wrapper.*

Elements of (#Flow) container are called flows *or* attention flows*.*

Each particular flow can be seen as a thread connecting the future with the past through the present to enable to track attention-connected aspects of situation and environment in time dimension.

Attention flow focuses attention paid to activities of selected kinds within long period of time to reach continuity, improvement, and sustainability in performing these activities. The selection of activities being focused is domain and/or goals and needs dependent.

Excuse me, my dear reader, but a complete understanding of flows needs to read the following explications of the Diamonds connections.

Organizationally, for every attention flow there exists corresponding flow-project, which defines objectives to be achieved in respective flow for the forthcoming period.

Technically, the attention flow measurement and evaluation is based on summing up all credits assigned to activities by attention function[7] over whole program or portfolio of projects.

Speaking on attention flow it is natural, that such flows have some limits. Attention paid to activities so as long-lasting attention flow are in a sense finite. We can assume in each situation we have at our disposal appropriate *attention pool* from which the attention is distributed over activities.

—

Let's turn our attention to connections explaining the whole game of „advice-advice itself-organize“ in relational perspective.

As the first thing of an agent's life is to do something we begin with focusing the (#Agent)–(#Activity) relationship.

[6]see definitions 10 and 13

[7]Attention function will be explained immediately in the following part of this section; it assigns credits to each activity-agent pair to express how much attention was/is/will be paid by this agent to this activity

An agent when doing an activity pay some attention to the work on this activity. When planning a work on activity to an agent, again, a necessary amount of attention must be planned. Thus the relationships between agents and activities need to be appraised by the amount of paid attention:

((#Agent), (#Activity)) $\rightarrow \langle 0, \infty)$,

where the interval of non-negative real numbers contains amounts of assigned attention.

Let's understand that this attention is the same one as the attention measured in elementary beliefs in the previous section 5.2 about the Diamond of Cognitive Elements.

Let $att_{agent}(x, t)$ be the function which assigns to each item

$x \in (\#Item)$

a non-negative real number expressing the amount of attention paid by this agent to this item x in a given time point t. This function evidently depends on all contexts, on all categories, and on all items the agent has in its memory–see fig. 5.2. But it depends not only on the content of memory of that one agent. The agents are co-operative and they communicate. Thus they share parts of contents of their memories. In fact, the function $att_{agent}(x, t)$ depends on memory contents of all agents being involved in a way in this game. Let *EnvAct* be a set of all items being necessary for decisions taking place during the activity execution, i.e.

EnvAct = $\{x \in (\#Item) |$ x *is used within the activity execution* $\}$.

Then the attention paid by the agent to the activity execution in the time interval R is

$\int_{t \in R} (\sum_{x \in EnvAct} att_{agent}(x, t)) dt$

which, I am sorry, is completely for nothing!

Nobody knows and nobody will know how to compute such value, which is the value of the particular assignment of the type

((#Agent), (#Activity)) $\rightarrow \langle 0, \infty)$.

Hence, the only possibility is to introduce a type of credit function

((#Agent), (#Activity)) $\rightarrow$ Credit,

which instances are called *attention credit functions* or *credit functions* or *attention functions* and

where *Credit* is a set of pragmatically chosen credit values. An example of this could be credits used within university studies to evaluate the load of particular courses.

To elaborate with the Credit set is a matter of concrete implementation so as to choose the appropriate credit function.

There are four possible relations between (#Agent) and (#Activity) sorts. We start chronologically according to this relationships life-cycle.

An agent when meets a new kind of activity is first in the position of a learner to this activity.

Definition 25 (Agent-Activity Learner Relationship) *The relationship* learner *is a container for connections connecting a credit to pairs of particular instance of (#Activity) and particular instance of (#Agent) with the meaning:*

(Credit) assigned to given (#Agent) participating on given (#Activity) in the role of learner /0,1:0,M.

Trivially defined over this connection is:

(#Agent)-s participating on given (#Activity) in the role of learner /0,M:0,M
(mentioned here for the sake of explicitly expressing the cardinality).
The learning process can be realized by

- *reading the mention form of the activity's operations*
- *monitoring of execution of the use form of the activity's operations*
- *monitoring of inputs and outputs of the activity's operations use forms*

The learner learns by watching other agents performing the given activity how to cope with this activity, what are the principles governing this activity and what is the position of the activity in the context of other activities.

Being acquainted with the kind of activity the former learner could start to co-operate with other agents performing this kind of activity in the role of collaborator.

Definition 26 (Agent-Activity Collaborator Relationship) *The relationship* collaborator *is a container for connections connecting a credit to pairs of particular instance of (#Activity) and particular instance of (#Agent) with the meaning:*

(Credit) assigned to given (#Agent) participating on given (#Activity) in the role of collaborator /0,1:0,M.

Trivially defined over this connection is:

(#Agent)-s participating on a given (#Activity) in the role of collaborator /0,M:0,M

(mentioned here for the sake of explicitly expressing the cardinality).

The collaboration means that the agent executes use form of operations of given activity in co-operation with the agent playing the role of author for this given activity. The agent-collaborator is subordinate to the agent-author within this activity execution.

When the agent-collaborator knows to cope with the kind of activity, is well trained in doing it, and is experienced enough in the kind of activity circumstances, it can start to play the role of author.

Definition 27 (Agent-Activity Author Relationship) *The relationship* author *is a container for connections connecting a credit to pairs of particular instance of (#Activity) and particular instance of (#Agent) with the meaning:*

(Credit) assigned to given (#Agent) participating on given (#Activity) in the role of author /0,1:0,M.

Trivially defined over this connection is:

(#Agent) participating on a given (#Activity) in the role of author /1,1:0,M (mentioned here for the sake of explicitly expressing the cardinality).

The authorship means that the agent executes use form of operations of given activity either alone or in co-operation with other agents playing the role of collaborators for this given activity. The agent-author is responsible for the activity result and it is superordinate to the agents-collaborators within this activity execution.

Note, that each activity must have just one agent-author.

The last role of an agent with respect to an activity is the supervisor role. An agent, who used to be author in lot of activities, not of the kind of the pointed one, only, with enough experience and with broad and deep knowledge of near contexts and possible connections, could turn to be a supervisor of this activity.

Definition 28 (Agent-Activity Supervisor Relationship) *The relationship* supervisor *is a container for connections connecting a credit to pairs of particular instance of (#Activity) and particular instance of (#Agent) with the meaning:*

(Credit) assigned to given (#Agent) participating on given (#Activity) in the role of supervisor /0,1:0,M.

Trivially defined over this connection is:

(#Agent)-s participating on a given (#Activity) in the role of supervisor /0,M:0,M

(mentioned here for the sake of explicitly expressing the cardinality).

The supervising means that the agent evaluates to which measure and quality the result of the given activity align with results and states of elaboration of other activities in a respective environment of this given activity. The alignment evaluation is done in both dimensions: the structural one and the behavioral one. The environment depends on particular contexts being involved in the situation.

The agent-supervisor does not execute any use form of operations of given activity. The agent-supervisor reads mention forms of results delivered by agent-author and possibly can execute use forms of available verification and validation activities. A verification means testing. A validation covers prospecting activities, seeking of anomaly behavior and non-plausible results or reactions to impulses from the environment.

Note, that there could be more supervisors for a given activity. This happens when the activity adds to more different contexts and one supervisor is not able to cover all these contexts.

There are two possible relationships between an agent and a goal (or an objective or a deliverable). The first is that an agent can follow a goal (or reach an objective or it can deliver a deliverable). The second is that an agent can establish a goal (or an objective or a deliverable). This is enough; Lee Iacocca from Chrysler Corporation said: Lead or follow, otherwise take off!

Definition 29 (Agent-Goal Follower Relationship) *The relationship* follower *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Goal) with the meaning:*

(#Agent)-s following or reaching or delivering given (#Goal) /0,M:0,M.

The goals are followed.

The objectives are reached.

The Deliverables are delivered.

Definition 30 (Agent-Goal Establisher Relationship) *The relationship* establisher *is a container for connections connecting particular instance of (#Agent) to particular instances of (#Goal) with the meaning:*

(#Agent)establishing given (#Goal) /0,1:0,M.

The agent from this definition is an originator of the addressed goal or objective or planned deliverable. When speaking of deliverable this agent is not the one who will deliver it, as this role is linked to agent follower in the definition 29. Here this agent could be called a planner of the deliverable.

Note that so as an activity can be executed by an agent only, the goal or the objective or the deliverable can be established again only by an agent. People very often speak about goals or objectives or deliverables established by a project or program of projects or in our case by a flow. But this is a metonymy in the Lakoff sense. A project or program or a flow could be an inspiration for goal/objective/deliverable establishing, or it could be a framework for such establishment. But the goal or the objective or the deliverable can be established by something what has the property of autonomy, only. The cardinality says that there could be goals to which an establisher is not known.

When relating agents to actions/composite activities/projects or to flows two typical relationships are involved: an agent could be an organizer in this connection or it could be a member here.

Definition 31 (Agent-Action Organizer Relationship) *The relationship* organizer *is a container for connections connecting particular instance of (#Agent) to particular instances of (#Action) with the meaning:*

(#Agent) organizing given (#Action) /1,1:0,M.

To organize a project or a composite activity or an action means to perform use-forms of controlling operations forming these processes of the Service System described by this Diamond that serve to control and to organize.

Particularly these operations are

- *operations with disposable attention pool*[8]*, i.e. resources allocation and re-allocation,*
- *accountability distribution, i.e. organizing of authors, collaborators, supervisors and learners to a synergic execution of particular activities in alignment with defined plans,*
- *traces recording, i.e. recording of all history of the project/composite activity/action not only in performance dimension but in attention dimension, too; the attention is traced just by the attention function.*

[8]see the text after the definition of Flow–def. 24

Definition 32 (Agent-Action Member Relationship) *The relationship* member *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Action) with the meaning:*

(#Agent)-s participating in given (#Action) in the role of members /0,M:0,M.

To be a member means that the agent participates or is planned to participate at least in one activity covered by the action as an author or collaborator or supervisor or learner.

The role of member of a team of agents is not definable over Agent – Activity and Activity – Action relationships because there exist such states of world in which the agent can be assigned to an action as a member but one or both of assignments done by Activity – Action and by Activity – Agent relationships can be unknown.

The same relationships as for the pair Agent – Action exist, will exist for a pair Agent – Flow.

Definition 33 (Agent-Flow Flow-Organizer Relationship) *The relationship* flow organizer *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Flow) with the meaning:*

(#Agent) organizing given (#Flow) /1,1:0,M.

To organize a flow means to perform use-forms of controlling operations forming these processes of the Service System described by this Diamond that serve to control and to organize for a long lasting sustainability.

Particularly these operations are the same as in definition 31 with small differences:

- *operations with disposable attention pool*[9]*, i.e. resources allocation and re-allocation, focussed to working methods,*
- *accountability distribution, i.e. organizing of authors, collaborators, supervisors and learners to a synergic execution of particular activities in alignment with defined plans concerning with working methods,*
- *traces recording, i.e. recording of all history of the flow not only in performance dimension but in attention dimension, too; the attention is traced just by the attention function.*

[9]see the text after the definition of Flow–def. 24

The main difference comparing it with def 31 is that not the work itself is focused but the working methods are.

When a concrete action has to be done a specific project is designed. The flows do not substitute projects.

Definition 34 (Agent-Flow Flow-Member Relationship) *The relationship* flow member *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Flow) with the meaning:*

(#Agent)-s participating in given (#Flow) in the role of flow-members /0,M:0,M.

To be a flow-member means that the agent participates or is planned to participate at sessions where co-operative decisions are taken on the following activities executing, namely on attention paid to these activities, on projects/composite activities/actions performance, again namely on attention paid to these projects/ composite activities/actions, and on attention distribution to goals/objectives/deliverables monitored and being filled by the given flow.

The role of flow-member of a team of agents is not definable over Agent – Activity and Activity – Flow relationships because there exist such states of world in which the agent can be assigned to a flow as a member but one or both of assignments done by Activity – Flow and by Activity – Agent relationships could be unknown.

—–

The figure 5.4 –Diamond of Organization– tell us about whole-part compositions as well. These compositions deal with agents, goals and with activities.

Definition 35 (Agent-Agent Composite Relationship) *The relationship* composite *is a container for connections connecting particular instance of (#Agent) to another particular instances of (#Agent) with the meaning:*

(#Agent) which was created by a composition in which given (#Agent) participates /0,1:0,M.

The definition says that the agent cannot be a direct part of more than one „whole-agent“. It doesn't contradict the situation when an agent participates in more teams of agents. On the contrary, this composition of agents expresses much closer relationship of agents than a team of agents. Teams of agents are grouped by actions or by flows, and they are dynamic ones, depending on situation. A composition of agents into a new more complex agent is something stable in a lot of situations.

The composition or decomposition (depending on a taken point of view) of goals/objectives/deliverables is nothing other than the WBS = Work Breakdown Structure well known from project management discipline. We call it goal breakdown structure (GBS), here, to be consistent with other definitions.

Definition 36 (Goal-Goal GBS Relationship) *The relationship* GBS *is a container for connections connecting particular instance of (#Goal) to another particular instances of (#Goal) with the meaning:*

(#Goal) which is directly superordinate to given (#Goal) within a goal breakdown structure /0,1:0,M.

Goals can be superordinate to another goals and they are superordinate to all objectives and to all deliverables.

Objectives can be superordinate to another objectives and they are superordinate to all deliverables.

Deliverables can be superordinate to another deliverables into any finite level

A GBS is a tree which branches can end at various levels.

Definition 37 (Activity-Activity Detail Relationship) *The relationship* detail *is a container for connections connecting particular instance of (#Activity) to another particular instances of (#Activity) with the meaning:*

(#Activity) to which contributes given (#Activity) as a part of its direct decomposition /0,1:0,M.

Now we understand the center of the Diamond of Organization and the compose-ability of agents (composites), goals/objectives/deliverables (GBS's), and activities (details).

—

Let's turn our attention to the Diamond circumference. Following definitions help us to understand deeply the actions and flows relationships to activities and to goals.

The first of this definitions gives an amendment to the entity sort (#Action) definition, only.

Definition 38 (Action-Activity Covering Relationship) *The relationship* covering *is a container for connections connecting particular instance of (#Action) to particular instances of (#Activity) with the meaning:*

(#Action) to which given (#Activity) is wrapped /1,1:0,M.

Note, that the cardinality tells „there are no unwrapped activities!“.

The second definition makes the same for the entity sort (#Flow):

Definition 39 (Flow-Activity Focusing Relationship) *The relationship* focusing *is a container for connections connecting particular instance of (#Flow) to particular instances of (#Activity) with the meaning:*

(#Flow) focusing given (#Activity) /1,1:0,M.

Again, think of the cardinality!

Two fundamental benefits come from this focusing relationship: The first is that summing up all attention paid by all agents to all activities focused by the given flow, and doing this for all flows, we can compare attentions assigned to particular flows and we can harmonize them or optimize them.

The second is that working methods established in the given flow are adopted in those projects to which focused activities are wrapped.

Thus, the flow and the focusing relationship appears a good attention distribution tool in Service System modeling and execution.

The upper half of the described diamond requires four definitions.

Definition 40 (Action-Goal Reaching Relationship) *The relationship* reaching *is a container for connections connecting particular instances of (#Action) to particular instances of (#Goal) with the meaning:*

(#Goal)-s reached by using activities wrapped in given (#Action) /0,M:0,M.

This describe a way by which an agent follows goals/objectives/deliverables as described in definition 29. Note, that the *follower* relationship is not definable over the *reaching* relationship and any of *organizer* and *member* relationships. There exist such states of affairs in which some of these relationships could be unknown while the *follower* relationship is known.

Definition 41 (Action-Goal Establishing Framework Relationship) *The relationship* establishing framework *is a container for connections connecting particular instance of (#Action) to particular instances of (#Goal) with the meaning:*

(#Action) in framework of which given (#Goal) was established /0,1:0,M.

This relationship will be used to remember such a situation in which a project or a composed activity or an action arrived to a state in which an agent, not necessary connected to this project/composite activity/action by a relation from def. 32 or def. 31 (but often being connected by this way), decides to establish the respective goal/objective/deliverable. This relationship helps to understand better the context of the respective goal/objective/deliverable establishment.

As the flows have to be goal-oriented, too, to make the whole game transparent, last two definitions could finish the Diamond of Organization description.

Definition 42 (Flow-Goal Fulfilling Relationship) *The relationship* fulfilling *is a container for connections connecting particular instances of (#Flow) to particular instances of (#Goal) with the meaning:*

(#Goal)-s fulfilled by given (#Flow) /0,M:0,M.

Definition 43 (Flow-Goal Establishing Flow Framework Relationship) *The relationship* establishing flow framework *is a container for connections connecting particular instance of (#Flow) to particular instances of (#Goal) with the meaning:*

(#Flow) in framework of which given (#Goal) was established /0,1:0,M.

Again, as in definition 41 this relationship will be used to remember such a situation in which a flow arrived to a state in which an agent, not necessary connected to this flow by a relation from def. 34 or def. 33 (but often being connected by this way), decides to establish the respective goal. And so as in the case of definition 41 this relationship helps to understand better the context of the respective goal establishment.

—

Lot of things could be done using in this way described model. Lot of complicated projects can be planned and realized to describe a Service System behavior using such defined model. But the model is still a flat one. Attentive reader knows it is like to describe a plane when watching the whole space! The missing point are my favorite R-edges which fire us „from the plane to the space“.

As I told in the introduction to this section here these R-edges play a more generalized role than in the first two diamonds–the Diamond of Attention Focussing and the Diamond of Cognitive Elements. In contrary to these two starting diamonds, the R-edges are not a centric ones but two of them are on the diamond circumference and the last three are recursive, see fig. 5.4.

First, definitions of R-edges will be set and then I try to explain why organizing entity sorts (i.e. the sorts (#Action) and (#Flow)) do not have recursive R-edges.

Definition 44 (Representation Relationships) *The relationship* R01 *is a container for connections connecting particular instance of (#Action) to particular instance of (#Activity) with the meaning:*

(#Activity) which in other context represents given (#Action) /0,1:0,1.

The relationship R02 *is a container for connections connecting particular instance of (#Flow) to particular instance of (#Activity) with the meaning:*

(#Activity) which in other context represents the whole given (#Flow) /0,1:0,1.

Connections from these containers are called representation edges.

Note, that a long-life attention focused to a particular topic being recorded by a particular flow can be viewed in other context as a poor activity, a small episode in a much bigger game.

Definition 45 (Recursive Representation Relationships) *The relationship* R11 *is a container for connections connecting particular instance of (#Agent) to another particular instance of (#Agent) with the meaning:*

(#Agent) which in other context represents the given (#Agent) /0,1:0,1.

The relationship R22 *is a container for connections connecting particular instance of (#Goal) to another particular instance of (#Goal) with the meaning:*

(#Goal) which in other context represents the given (#Goal) /0,1:0,1.

The relationship R33 *is a container for connections connecting particular instance of (#Activity) to another particular instance of (#Activity) with the meaning:*

(#Activity) which in other context represents the given (#Activity) /0,1:0,1.

Connections from these containers are called recursive representation edges.

The recursive representation edge is a convenience aiming to force capabilities of autonomous Service System behavior. So as the representation edges in both, the Diamond of Attention Focussing and the Diamond of Cognitive Elements, bring universality into modeling and cognition of state of affairs, the R11 recursive representation edge brings the possibility to substitute a given agent in one context by its deputies in other contexts.

The research of this topic is in the beginning now. The possibilities which are open by this recursive representation edge are not completely explored, and they will become aware step by step so as they will be refilled step by step.

The R22 recursive representation edge brings the possibility to substitute a given goal or objective or deliverable in one context by its deputies in other contexts, generally not saving the original belonging to the goal or to the objective or to the deliverable. Thus, a goal in one context can appear as an objective in other context and as a deliverable in another context. And the other combinations are possible.

Switching attention from a project to a program of which this project is a part we often need such identity relationship helping us to recognize that two phenomenons in the space and/or cyberspace are in fact one and the same item. A project objective could be a program deliverable, e.g.

The R33 recursive representation edge is the last crucially important vehicle to an open and universal deal between agents, teams of agents, and the environment. Again it is a kind of identity relationship helping us to recognize that two phenomenons in the space and/or cyberspace (two activities in different contexts) are in fact one and the same item.

A natural application of this recursive representation edge is within a project-program-portfolio management. An activity in a context of portfolio can be a program or a project in other context, an activity in a context of program can be a project in another context.

—

And this is the end. The Diamond of Organization is completely described. I hope that it is not so difficult to imagine for example the PMI's PM BoK[10] or the IPMA's ICB[11] to be stored in a memory of an agent or a Service System, if the memory is organized according to Diamond of Organization amended by the Diamond of Cognitive Elements and the Diamond of Attention Focussing. These three diamond can comprise conceptual model of such memory.

Remember, that till now we covered the „See“, the „Recognize“, and

[10]PMI = Project Management Institute, PM BoK = Project Management Body of Knowledge

[11]IPMA = International Project Management Association, ICB = IPMA Competences Basline

the „Advice“ (or „Organize“) from the above mentioned S-R-A-D scheme. The last element–the „Do“–will be discussed in the next section 5.4 on the Diamond of Predictive Behavior.

One question remains as attentive reader knows: Why there are no recursive R-edges (so as no composite edges) around the organizing entities (#Action) and (#Flow)? The answer is easy. They are not necessary. If, e.g., we want to mention a given project in another context, we can take it as an activity using R1 edge, cross any context boundaries using R33 edge and to deal with the result again as with a project using once more the R1 edge (in opposite direction, only). Similarly it could be with flows.

When some composition of projects or actions or flows is needed, I am sure, you will find that the same reasoning can be applied to composite edges around these two entity sorts.

5.3.2 Project Management Point of View

In case of the previous two diamonds we used a brain-like functioning description as example of these diamonds utilization within a Service System modeling and execution. In case of the organization diamond a better way will be to show how it is utilized in a real situation of project and program management. This could help you to use this diamond autonomously.

We start with the following question: „How to make the development of the Management Process sustainable?“ [12]

To be clear about sustainability, we consider a process P of project/program/portfolio management as sustainable whenever it comprises sub-processes or sub-projects dealing with

- introspection,
- evaluation and
- revision

of actual methodology of process P in all recognized contexts[13] to achieve more effective and efficient execution of project/program/portfolio steps.

To achieve this, the project/program/portfolio teams together with appropriate technology (project management supporting SW tools) have to form a composite possessing properties of Service System. Moreover, this Service System have to be organized following the way described by the Diamond of Organization.

The point here is that services/service systems are in its essence very similar to the projects/programmes. Firstly, they both are being undertaken for benefit realization. Thus, they both are target/goal oriented. They both are unique processes, or time oriented processes indeed, with the commitments and milestones specified. They both involved people and technology as their crucial components. Thus, they both are resources consuming. They both are based on cooperation and synergy. Their provision or execution can be either reflexive or based on the partnership.

[12] most of this subsection is derived from the article 'How to Make Development of Management Process Sustainable' published in Proceedings of IPMA World Congress, 2009, Helsinky, by Michael Oskera and me

[13] which align the process P with current and predicted situation and adapt it to evolving environment and needs

Therefore, our approach to the development of management process is based on following assumptions

1. that development of management process itself is project, and that
2. this project is service provision and
3. the service of management process provision is project.

The commonly discussed reasons of management process failure are

- not clearly defined target segment of market including well described customer and their needs,
- confused reasons of building a management policy and organization, and finally
- poor management and execution.

Therefore it is convenient to see the process of building up a management process and the company organization as quite complex project or programme, which need to be executed properly.

To be able to compare and unambiguously share and understand the basic project documentation, the set of prerequisites had to be specified for every project prior its launch. These prerequisites follows project management standards, like PMI or IPMA standards, and thus they are not surprising. Specification of each project has to include:

- project benefits,
- project goals,
- project objectives,
- project deliverables, i.e.
- product oriented work breakdown structure (WBS),
- project tasks bound to the WBS,
- task schedules,
- project team including project manager, and
- task assignments.

Product state/Task categories. The need of to be able to easily recognize in which stage of development is the given product of the any project led us to define several categories which may be applied on both products and tasks. This classification is derived from the life cycle of typical product being produced in MM by any kind of project. It is iterative process comprising following phases:

- research,
- specification,
- implementation/realization,
- production,
- business, and
- education.

Of course, any phase can be skipped because it is trivial or irrelevant to respective product, but in general it holds that to get any product to any of these stages there exist task or tasks, which are undertaken to do so. Therefore, we talk about both product and task classification.

This classification is materialized as particular categories in the (#Flow) container of the Diamond of Organization. Each such category represents its own flow. Thus we speak of *research flow*, *specification flow*, *implementation flow*, *production flow*, *business flow*, and *education flow*.

Attention as a measure for tasks planning and evaluation. Classifications of projects and products/activities introduced above would be hardly useful without ability to quantify what and to which extent is happening there. One of the possible and also usual measures is the time. Although our measure is also based on time, we rather talk about attention measure. The attention is concentrated on the problem or not, whereas time can be simply wasted.

The importance of attention measuring using an attention credit function from the Diamond of Organization originates in issues in motivation and rewarding of team members (human agents) on a project/program or portfolio.

On one hand it is important to support synergy in a team, sharing of accountability, common focusing to success but on the other hand we need a healthy self-confidence of particular team members (both, human or artificial agents).

Attention function is the means the purpose of which is a solution of above set situation.

As was said above *attention function* measures amount of attention paid by an *agent* to an *activity* relatively to the whole attention paid by all agents to all activities of the given *project/composite activity/action*.

As was said in definitions 27, 26, 28 25, an agent can work on an activity in one of the following roles:

- author (A) of the solution,
- collaborator (C),
- supervisor (S), and
- learner (L).

Author is responsible for the solution and works on this solution; collaborator helps the author if necessary (there can be work assignments assigned to author, only) and works on the solution.

Supervisor has to be an agent (team member) with higher level of overall understanding and knowledge; supervisor is responsible not for the task itself but for the alignment of the given task with the whole built configuration; he/she/it reviews the solution in a wider context.

Learner learns by watching the working members.

By learning one extends its ability to be an author of particular tasks or to be a supervisor of a particular built.

A day work of an agent in a particular role is appraised by a defined number of credits; denoting c_A number of authors credits per a day work and respectively $c_{(}C)$, $c_{(}S)$, $c_{(}L)$ for the other roles, the rule is

$c_{(}S) > c_{(}A) > c_{(}C) > c_{(}L)$.

An example could be to set

$c_{(}L) = 0$

and

$c_{(}A) = 3/2 \times c_{(}C)$.

Number of credits inscribed to supervisor per reviewing a task could be some ratio of all credits assigned for this task. Exact numbers must be tuned by experience. Sometimes could be useful to use the ratio 1/5.

Let's assume that each list deliverable of any used WBS will be created by using one activity, for the convenience.

Assigning activities to agents according to definitions 27, 26, 28, and 25 each task with assigned resources is labeled by estimated credits and estimated time length. Thus, each assigned task has its own credit value so as each resource (team member) is assigned a number of credits per this task. This is done publicly as a part of planning of the future work or evaluation of really reached state within a project of setting and evaluating goals which accompanies the projects under discussion. Hence, the program of projects under discussion or the portfolio of projects and programs under discussion is enriched by a special project enabling the management of all these projects and their management as the whole is sustainable and effective.

When product breakdown (the WBS), which turns to be a GBS = Goal Breakdown Structure in the diamond, is changed thanks the progress in our understanding of the whole solution, the credit assignments and time estimations are revised. When the task is completed this assignment is revised to reflect the reality. As the planned values are remembered, we have possibility to compare ideas with the reality.

The sum, denoted $A_(P)$, of all actually assigned credits over the whole product breakdown represents the overall attention dedicated to the project by the whole team.

The sum, denoted $A_(M)$, of all actually assigned credits to one team member over all by her/him solved tasks represents the overall attention dedicated by this team member to the project. The portions of those credits assigned in roles author, collaborator, supervisor are denoted $A_(aM)$, $A_(cM)$, $A_(sM)$, respectively.

A particular team member „deliverable" to the project is $A_(M)/A_(P)$. Her/his deliverable in a role author, collaborator, or supervisor is

$A_(aM)/A_(P)$,

$A_(cM)/A_(P)$,

$A_(sM)/A_(P)$,

respectively.

A floating part of team member revenue is calculated based on these deliverables.

One can imagine that we obtain lot of very interesting information about the project life thanks to the attention function. A deep insight into a project and its inner environment is at our disposal.

This valuation of elementary tasks in combination of task, product, and project categorizations gave us possibility to define characteristics, which can be easily and uniformly measured and evaluated over whole program or its arbitrary parts. The measurements are performed by summing up all credits assigned to the tasks, which satisfy filtering criteria. Measurement can be confined by any combination of items from following categories:

- time interval,
- team member(s),
- project(s),
- kind(s) of project,
- kind(s) of activities,
- character(s) of activities (e.g. meeting, work travel),
- status of activities (e.g. complete, incomplete, postponed),
- objective(s), and
- goal(s).

Thus, it is possible for example to measure effort paid by certain team members to the research tasks within last two weeks on two selected projects. Also the contribution of individual team members to the program can be measured and serve as a lead for incentives pay-offs. Very interesting are also overall graph views of who and to what extent works on which project (see figure projects and team members TODO). Also views of who works with whom (see figure project cooperation TODO) can be distilled from the attention function usage. The ellipses stand for people, whereas rectangles stand for projects. In addition, these graphs can be drilled down into more detail information.

⋇ [STA]$_{110}$:Figure from PE: projects and team members

⋇ [STA]$_{111}$:TODO sem dat obr Projects and team members

⋇ [STA]$_{112}$:TODO sem dat obr Project cooperation

Attention function solves the above cited issues. How it does it we will see in the next subsection ??

Conditions of use the Attention Function in a real company's team are:

- Tasks have to be breakdown into tasks elements which are in a way homogenous, i.e. mutually comparable in some parameter.
- Program management has to now deeply
  - specification of what to be done and how to do it,
  - the whole team and its capabilities
  - each single team member and her/his capabilities
- Execution of any particular task takes short time, at most two weeks, better one week,
- The management has to have balls!
- Network organizational structure; it is a natural consequent of Attention function. There is no separation of the pool of authors/leaders from the pool of collaborators.

Therefore, utilization of attention function can bring very interesting information about the „life of projects". A deep insight into a programme and its inner environment is at disposal.

Long-term focus of attention - Flows. The combination of attention measure and task classification mentioned above offer yet another new perspective - perspective of long-term focus of attention based on the typical product creation life cycle. We called them attention flows accordingly to the Diamond of Organization, where every kind of product/task corresponds to one flow. Thus, we get R-Flow for research activities, S-Flow of specification activities, etc. Attention flow focuses attention paid to tasks of selected kinds within long period of time to reach continuity, improvement, and sustainability in performing tasks of that kind. This selection is domain and/or strategic aims dependent. Organizationally, for every attention flow there exists corresponding flow-project, which defines objectives to be achieved in respective flow for the forthcoming period. Technically, the attention flow measurement and evaluation is based on summing up all credits assigned to tasks by attention function over the whole program or portfolio confined by flow-corresponding kind of task.

The above described topics of program and project management doesn't bring benefits each by its own, only. There is substantial benefit arisen from the mutual synergy effect of those topics when utilized together.

From the procedural and organizational perspective, the addressed issues here are dealing with setting up the organizational structure of the management team and with the setting up the rhythm of PPPM process.

Strategic Management and Management In Large Extent. Obviously, to succeed in one particular project or projects does not mean to succeed in achieving the whole business goal. Therefore, the success of the whole programme depends on ability to manage whole programme in the way that goals of all projects are achieved. Consequently, any programme or portfolio has to be simultaneously considered as a project, i.e. it has to be also triple-constrained having its goals, schedule, and overall budget specified.

There are two basic categories of inter-project dependencies:

1. logical dependencies (for instance, task A in project $P_(1)$, cannot start prior task B in project $P_(2)$ is finished), and

2. resource dependencies (resource X can work just on one task at the moment).

Constraint known as critical chain (Goldratt, 1997) of the project/programme is the longest sequence of depending tasks.

Thus, to manage the programme it is necessary

1. to manage the project essence of the programme by means of strategic management and

2. to manage comprised projects in double-level fashion:

   (a) to manage each single project (management in small extent), and

   (b) to manage the whole set of projects according to constraint corresponding to critical chain (management in large extent).

To do so, we added three organizational projects into the programme or portfolio under discussion:

- Strategy Project (SP),
- Project of Management In Large Extent (MILE), and
- Project of Forming In Large Extent (FILE).

Goal of SP is to continuously

1. perform strategy analysis,
2. choose strategic direction, and
3. elaborate strategy achievement to the extent that it can be realized by other projects within programme.

In fact, team of SP is top management of the respective company.

Goal of MILE project is to continuously monitor and evaluate all projects within their mutual interactions and interconnections induced by dependencies and changes. Attention function is very beneficial here as it can easily reveal overloaded resources in the context of assignment to tasks and respective objectives and goals.

Team of MILE-like projects can be called by term steering committee, as it is commonly understood. The key point here is that steering committee has ultimately to be part of the top management.

The third topic is forming the team. If the triple constraint of the whole programme is floating, which yields also to the floating WBS and other plans, the FILE project was introduced. A portfolio can be seen as a programme with very floating goals and one shared goal–financial success. Thus, the FILE project has its place in portfolio too.

Goals of the FILE project are

- to prepare,
- to inform, and
- to unify

the whole programme/portfolio team in what the next steps of further development will be. We call these activities as forming of the team. Forming is one of possible ways of Mention/Use principle or PDCA cycle application. From the periodicity perspective, forming of the team is performed at three stages:

(a) forming seminars (FS),

(b) forming weeks (FW), and

(c) forming days (FD).

The timing corresponds to the overall rhythm of the organization (see figure rytmus PPPM),

⋇ [STA]$_{113}$:TODO sem dat fig rytmus PPPM

where the largest recognized time period is stage. It is of strategic importance and its duration could be approximately half of year. Before beginning of each stage, FS takes place. It is usually two-days workshop aimed to communicate and elaborate the strategy to the team for next upcoming stage. Usually, goals of projects are discussed and specified.

Each stage is divided into 3 iterations. Iteration is the period which objectives to respective goals are planned for. Iteration is transition from strategic to tactical level. Its duration is approximately two months.

Each iteration is split up into 1 special and 4 regular phases. The special phase is called phase 0. It is a span where FW takes place. As the name suggest its duration is usually one week. Within this period project plans for the respective iteration are elaborated, i.e. for every project, goals are revised, specific objectives are discussed and specified, WBS is elaborated, and tasks and assignments for the next phase are share out. There is also space to get and evaluate feedback on PPPM methodology and agree on any improvements (Mention phase), which will be validated in the following regular phases of the iteration (Use phase) and revised in the special phase of the next iteration (Mention phase).

This is the ultimate self-referential aspect of whole methodology as all, what is mentioned here in this section 5.3.2, can be revised and potentially changed. Note, that the previous sentence is written in section 5.3.2 as well.

Projects are executed within 3 regular phases. At the beginning of each phase the FD takes place. FD is aimed to harmonize plans of all running projects for the current phase and to share what exactly should be done in that phase. The harmonization is heavily based on the repeating online measurements of attention function over project plans, which are simultaneously tuned.

Mills. The most important thing on strategic management, management in large extent, and forming activities, is to perform them on regular basis and in harmonized way. However, the regularity here is rather approximate than absolute, because the rhythm has to reflect and absorb irregularities, which may interfere it both expectedly (e.g. Christmas, public holidays) and unexpectedly (e.g. illnesses, vacations).

We use metaphor of mills of here. The MENTION and USE switching is tightly connected to a well known competition between theory and practice. No one of them could proceed alone. Trying to proceed alone is like jumping on one leg. Thus I call this principle „two legs principle". Trying to make a long journey, e.g. trekking in high mountains, the crucial focus is on regularity of our steps. In other case we fall down and never reach our goal. The mills mill regularly. That is why we speak of mills.

If the flour-mill is to be beneficial, it has to be in operation regularly or constantly, because its products are always necessary no matter, what is happening in e.g. politics, business. It is the same with PPPM. This process and its parts have to be executed regularly otherwise the chaos will reign the course of events. The only difference is that flour-mill always grinds cereals, whereas PPPM mill always grind outer and strategically significant events, projects and their interconnections.

A three mills approach can be established in general:

- mill of individual project,
- mill of the whole programme, and
- mill of the program-organization co-operation.

This general approach can be modified. Establishing mills and keep them working is crucial to succeed and is one of the hardest task within the PPPM process implementation service provision. We see the running mills as the key to the sustainability of the PPPM.

Network organizational structure. The last, rather organizational, result of Diamond of Organization application is the application of so-called network organizational structure (NOS).

The application of NOS is a natural consequent of the attention function utilization. There is no need to separate the pool of authors /leaders from the pool of co-workers to be able to value people accordingly. Attention function does it in distributive way at the most bottom level, i.e. at the task level. Thus it is common that a team member A is sometimes assigned to a task as a chief of a team member B, and thereafter the team member A is assigned to a task as a co-worker of the team member B. Therefore, it is possible to find the most efficient role assignment to every task. Thus, everyone can learn from everyone by doing an activity together in appropriate and fitting roles. Usually, workgroups comprise from 2 up to 4 team members.

I also strongly recommend to apply NOS at the project level. It means that every team member is responsible at least to one project. This leads to step-by-step improvement of managerial skills in the whole team and raise the level of engagement. Thus, the forming activities are performed within the whole team where everybody is interested and paying attention.

The NOS is closely tight to FILE. The benefit from this approach is enriched when the whole team is involved, i.e. each team member takes active participation in the discussions and negotiation.

⋇ [STA]$_{114}$:vyuzit reversed triangle inequality for information capacity jako podporu tohot zpusobu prace–nedochazi k ztratam informace

⋇ [STA]$_{115}$:TODO few words to make a conclusion

5.4 Diamond of Predictive Behavior

What is the way of to act in space–the real time-space and the cyberspace? What means „to require something“? What is the neighborhood of the requirement when we want to fulfil it or solve it? Such questions and other similar questions look like an abstract philosophical verbosity. But can we imagine an autonomous subject doing something aimed will be completely comprehended without answering the questions: What is requirement? What is its neighborhood when a solution is sought for it?

TODO

The Diamond of Predictive Behavior was discovered in the second half of year 2007. I tried to draw a model of application production, i.e. I tried to model the „Do“ from our S-R-A-D scheme, in this time. My idea was to support a production ability of our Mycroft Mind team. In other words it was an attempt to describe in a generic form how to do what was aimed. And this is the predictive behavior what is the cornerstone of this.

The Diamond of Predictive Behavior serves as a conceptual reference model of service system, which may provide more elaborated background for service execution and service innovation processes.

5.4.1 System Architecture Point of View

Again, as in previous three diamonds, we start with an intuitive description. Here it is much more important than in previous cases, as this diamond is more complex. The first three were „plain diamonds“ consisting of five containers. This one is a 3D diamond consisting of 7 containers. Three of them we know from previous diamonds, namely the containers (#Goal) and (#Agent) which were in the third diamond, and the container (#Context) which we know from the second diamond.

The fourth known element is Service: we haven't it yet as a container but we know the content–service was defined in section 3.1 by definition 1.

These 7 containers are connected by 36 edges. It is convenient to see the picture on fig. 5.5 this way that the axis Agent–Requirement–Model is perpendicular to the plane Goal - Context - Service - Use-case. Thus we can see three intersecting plane (sub)diamonds containing five containers each:

(sub)diamond (Goal - Service - Context - Use-case),

(sub)diamond (Goal - Service - Agent - Model), and

(sub)diamond (Context - Use-case - Agent - Model) with one common point–the Requirement.

Our explication starts again from the center where the basic element is.

What is the basic element of predictive behavior? What could be done with it, and in what environment? ⋇ [STA]$_{116}$:TODO navaznost na Jeff Hawkins Taking into account that *autonomy can be defined as the property of a system to contribute to its own persistence* ⋇ [STA]$_{117}$:cite David Vernon Cognition in Artificial Systems, 2006 I argue the answer is nothing more and nothing less than *requirement*. Requirement is a record of desire or of will or of wants. Requirements and their meeting–this is the mechanism of a system to realize contribution to its own persistence. And Service Systems are autonomous systems.

A requirement could be

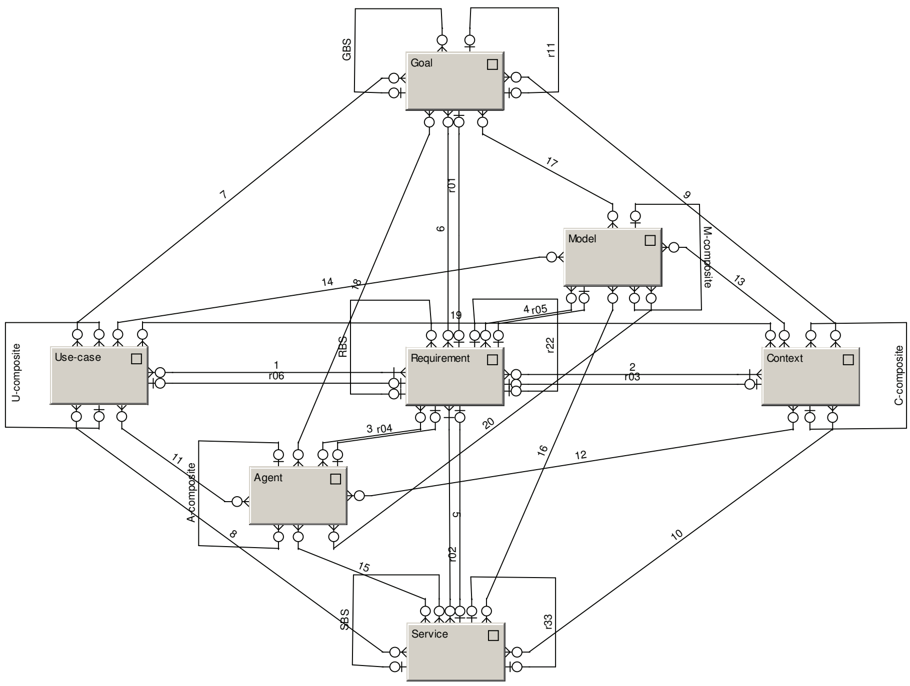

Figure 5.5: Diamond of Predictive Behavior

- something that is required in advance,
- a required activity,
- anything indispensable.

Requirements heads to fulfil some *goals* (or needs or to reach some objectives) and can be for the customer benefit solved by some *services*. These three (requirements–goals–services) form a not very tangible, really intangible, axis of will and satisfaction which is anchored in a way in a more tangible plane set by Agents, by Models, by Contexts, and by Use-cases (or behavioral patterns).

Requirements are addressed to partners or, generally, *agents* and this have to be remembered in a particular *model*. Be aware that a system cannot be autonomous if it has no memory. And this what is stored in a kind of memory are models.

Requirements are evoked by some behavioral patterns or *use-cases* or requirements could initiate such behavioral patterns or use-cases.

In any case, requirements occur within a domain or an application field, which forms their *context*.

Agents are bound to specific *contexts* and can be characterized by typical behavioral patterns, i.e. *use-cases*.

Agents can be co-operators or interested parties or competitors in a process of *goals* or needs fulfillment or objectives reaching.

In analogous way *agents* are connected to *services* within a process of solution creation.

Models remember not only *requirements*, but also *use-cases*, *contexts*, *services* and their descriptions, and *goals* and their descriptions, too.

Contexts are linked to *goals* achievable within them and to *services* which can be created in these *contexts*.

Behavioral patterns or *use-cases* are linked to *goals* achievable by means of these use-cases and to *services* which can be created by these use-cases.

All nodes of this Diamond can have its hierarchical breakdown; the nodes on intangible axis have recursive representation links crossing boundaries of particular contexts; each vertex can be in other context seen as a requirement trivial construction.

The Diamond of Predictive Behavior serves as a reference model for Service Systems. It is more complicated than previous three diamonds. For example, the edges here are not simple conceptual model edges as in previous diamonds; in turn all the numbered edges (numbered 1 - 20) are multi-edges as they represent two or more edges in a sense of conceptual model edges. Thus, we will go through precise definitions step by step using five phases of the diamond build.

Let's start with the diamond skeleton (see fig. 5.6). Its central element and thus the central element of all (sub)diamonds is the (#Requirement) container.

Definition 46 (Requirement) *Container (#Requirement) is defined in such a way that it contains each record documenting a single need of what a particular product or service should be or should do. It is a statement that identifies wanted and/or necessary attributes or capabilities or characteristics or quality of a system in order to it will have value and utility to a user.*

Two typical subcategories are distinguished: (#Functional Requirement) and (#Non-functional Requirement).

Object from the category (#Functional Requirement) is every description

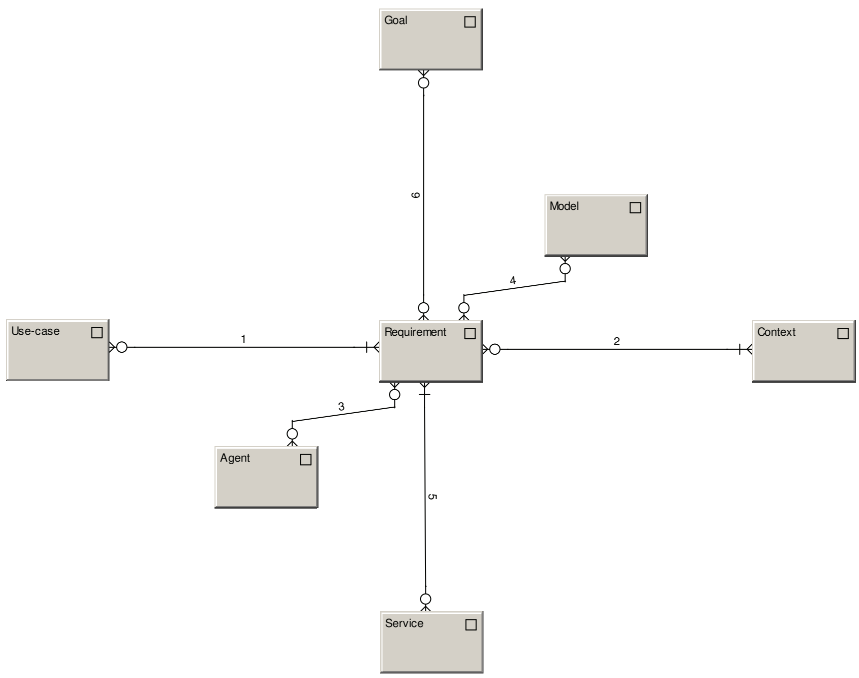

Figure 5.6: Diamond of Predictive Behavior - phase 1 of the concept

of the functions that the service consists of. They are sometimes known as capabilities.

Object from the category (#Non-functional Requirement) is every constrain of the service. Non-functional requirements specify criteria that can be used to judge the operation of the whole service or solution, rather than specific functions or behaviors. Non-functional requirements are sometimes known as quality requirements.

From the goal orientation of agents point of view the requirements are points of the mental space of an agent; each such point is assigned by a „desire mode“ of attention.

Elements of (#Requirement) container are called requirements*.*

An element of (#Functional Requirement) could be for example, formatting a text, computing a shortest path to a destination, or modulating a signal.

(Sub)container (#Non-functional Requirement) can contain, e.g. response time less than 5 sec., middle time period between two failures greater than 100 days, etc.

Note, that the *requirement* in Diamond of Predictive Behavior is analogous to *item* in Diamond of cognitive elements.

Container (#Goal) is defined in section 5.3 in definition 22. To remind it: object from the category (#Goal) is every formulation of a required state-of-affairs more or less definite. By the term „definite“ we mean that it is possible clearly recognize whether the required state is achieved, there is a possibility to measure parameter values of the required state and it is possible to determine a time point of the desired state achievement. The less definite formulations belong to needs. Formulations expressing orientations belong to goals. Really definite formulations belong to objectives. Deliverables usually do not play a role within this diamond. But it is not excluded.

NOTE: Need/Goal/Objective stands for Category when aligning Diamond of Predictive Behavior to the Diamond of Cognitive Elements. Need/Goal/Objective works in a sense as category giving an explication to focused connections.

Container (#Service) contains particular *services*. A *service* is defined in definition 1 in section 3.1. In other words we can say:

Object from the category (#Service) is a complex answer to a need/goal/objective expressed in terms of accomplishment of appropriate requirements arising from a registered need/goal/objective. Abstracting from the Clients benefits we can speak of *solution*. So, service is a solution bringing benefits to its Client.

Note, that according to Wikipedia, solution is a product, service, or combination of both which is said to solve a business or consumer's problem. This is a common understanding of this word. But we adopt the first one, the Service Science approach.

Service can be connected to the above mentioned need/goal/objective through appropriate model or through appropriate use-cases or through agents/partners or through contexts/domains/application fields, or, simply, through a set of requirements.

Service can be composed of other services.

Object from the category (#Service) is our activity for a customer that does not result in ownership. It can be an application made accessible for a customer. Only access data are delivered and the Service is operated on Provider own overhead. Or it can be consultancy work depleted for a customer using Provider's know-how.

In a special case a service can be reduced to a product. Then the product itself brings a direct benefit to a Client and thus seems to be „a service provider".

Object from the category (#Product) is a thing operated on clients/customers own overhead.

Note, that (#Service) stands for (#Manifestation) when aligning Diamond of Predictive Behavior to the Diamond of Cognitive Elements. The *service* in a way manifests what is in a value co-creation process the most interesting.

Container (#Context) is defined in section 5.2 in the definition 16. Pragmatically, in the game in which the Fourth Diamond is involved no all contexts playing a role in the Second Diamond are appropriate. Usually we use such contexts which could be called application fields or domains. Application field or domain is a narrow context of other elements in this Diamond. But no other context is excluded; it is not typical only.

Object from the category *Domain/Application field* is every domain or application field of human proceedings that can be supported by a Service System. Domains are chosen pragmatically according to their business potential. Domain or application field can be recognized according to existence of a conceptual data model describing this application field.

The remaining container in fig. 5.6 is a new one.

Definition 47 (Use-case) *Container (#Use-case) is defined in such a way that it contains every fixation and combination of functional requirements that add value to relevant business.*

Use-cases, stated simply, allow description of sequences of events and/or activities that, taken together, lead to a situation that a system does something useful.

To this end use-case describes a sequence of activities, those activities defined by definition 21 in section 5.3, composed into such structure that, when executed, brings a usefulness to a Client.

Elements of (#Use-case) container are called use-cases*.*

Each use-case is a scenario according to which a process can run.

Such scenarios help us to understand the game of Service System performance. It is an explanation by a type-example. Be aware, it is not a full specification of all possible aspects of Service System behavior.

Note, that (#Use-case) stands for (#CI-connection) when aligning Diamond of Predictive Behavior to the Diamond of Cognitive Elements. Use case connects some individuals=requirements into a goal reaching sequence, i.e. to something similar to a category. Let us aware that use cases are naturally categorized by their goals.

Container (#Agent) is defined in section 5.3 in definition 20. To remind it, object from the category (#Agent) is every agent acting on the market, which is autonomous and co-operative in a sense of holonic approach and to which attention can be reasonably payed. Agents are usually business partners of various kinds: clients, providers, suppliers, stakeholders, etc.

Agents are the actors in the Service Systems games. An agent itself can be seen as a Service System very often.

We are interested in agents/partners who are clients/customers or providers/suppliers or co-operators, or who are other way involved in the „life process“.

Remember that an agent could be not only people or organization of but an artificial item, too. And it doesn't matter whether such item exist now. As I told in chapter 3 there is no reason to separate human agents from the artificial ones. The most often case is agents are combinations of people and artificial items.

Container (#Model) is a new one notwithstanding that the content of it were spoken here many times. It is a nice example of self-reference as we think and cognize through models.

Definition 48 (Model) *Container (#Model) is defined in such a way that it contains every formalized (with prescribed proprieties) record or set of records or description enabling to remember anything important from the point of view of any element or any part of the described Diamond of Predictive Behavior.*

The term „enabling to remember" means the model provides containers for items and for connections (as well as for operations and integrity constraints) abstracted from a given part of real time-space and/or cyberspace.

Strictly said: the Diamond of Attention Focussing is a model. Everything which can be modeled by Diamond of Attention Focussing is a model. Nothing else is a model.

Elements of (#Model) container are called models*.*

Each model is a structural pattern according to which the process of remembering is running.

Usually the formalized record is done in terms of business potential of the Service System technologies utilization.

The definition of model is crucial. Let's understand that owing to the First Diamond universality the Second Diamond and the Third Diamond are models, too. Moreover, this Fourth Diamond is again a model.

A Service System to be autonomous has to have the ability to remember. The container (#Model) serves as memory of the Service System described by the Fourth Diamond.

When a new service is designed or a new Service System is developed a prospecting document carries basic facts and beliefs (see section 5.2).

Prospecting Document (which is a special case of model) should contain together with a formulation of needs/*goals*/objectives an application field description (*context*), a *requirements* specification and typical *use-cases* presentation and analysis. All this is anchored in a conceptual *model* of the situation. Then it contains connections to possible or given partners (*agents*), usually. Concrete *service* or solution specification is optional. Thus a prospecting document is a model with the structure of this Fourth Diamond. This is why this diamond can play a role of reference model of Service Systems.

The first picture tells us: *requirements* must be headed to *goals* and the aim is they are satisfied by *services*. The whole game is played in a domain or application field, which is the *context* helping us to understand matters of facts. To be able to imagine what happens *use-cases* bring a visualization. The game is played with *agents* which are actors in it. As time is important and lot of the new is based on the past, i.e. as history is unavoidable, a kind of memory have to be employed; this is done by *models*.

Relationships The last step of the first phase is to precise connections of defined containers.

As I told in introduction to the Fourth Diamond the lines at fig. 5.6 represent multi-edges as each of them covers several edges in a sense of conceptual modeling. The same will be true for all other not specially named lines at figures 5.7, 5.8 5.9 and 5.10. The cardinality of particular connection is ascribed to its semantic specification; the cardinalities at figures, i.e. cardinalities of multi-edges denote the most general cardinality of by the multi-edge wrapped connections.

The first two multi-edges form the vertical axis of the Fourth Diamond.

Definition 49 (Requirement-Goal Multi-edge) *The multi-edge* Requirement-Goal *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Goal) with the following meanings:*
(#Requirement)-s covered by given (#Goal) / 0,M:0,M.
(#Requirement)-s arising from given (#Goal) / 0,M:0,M.
(#Goal)-s arising from given (#Requirement) / 0,M:0,M.

Definition 50 (Requirement-Service Multi-edge) *The multi-edge* Requirement-Service *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Service) with the following meanings:*
(#Requirement)-s fulfilled by given (#Service) / 1,M:0,M.
(#Service)-s established to fulfill given (#Requirement) / 0,M:1,M.

Let's be aware of the relationships cardinality: it is important that a service must fulfill at least one requirement and there is no reason to establish a service which does not fulfill at least one requirement.

Note that the second relationship is not a rotation of the first; a service established to fulfill given requirement R_1 can fulfill besides requirements R_2 and R_3 but no one from these both is a reason to establish a new service.

Definition 51 (Requirement-Context Multi-edge) *The multi-edge* Requirement-Context *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Context) with the following meanings:*

(#Requirement)-s reviewed in given (#Context) / 0,M:1,M.

(#Context)-s which could be used to review given (#Requirement) / 1,M:0,M.

In chapter 3 I argued that contexts form the important part of Service Systems. In section 5.2 we get to know the power of contexts in cognition processes. Thus, it is not surprising that to do anything with a requirement without context is really „out of context“! This is the meaning of the cardinalities in the above definition.

Again, no one of these two relationships is a rotation of the other. E.g., a requirement is reviewed in two contexts only, but there could be three other contexts which are reasonably candidates to review it; it just haven't been performed yet.

Definition 52 (Requirement-Use-case Multi-edge) *The multi-edge* Requirement-Use-case *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Use-case) with the following meanings:*

functional (#Requirement)-s which are fixed and/or combined in given (#Use-case) / 1,M:0,M

(#Use-case)-s created to fix given functional (#Requirement) / 0,M:1,M

Maybe here I can explain a difference between a use-case and a process described as a composite activity in the Third Diamond. There is a fine difference in a pragmatic approach. A process is a sequence of in a way ordered or organized activities. There is nothing about wants or desires. But use-case while looking as a process is about wants, will, desires. A use-case without actor who wants or desires to follow this scenario is a nonsense.

That is why use-case fixes requirements while process is composed of activities. And this is at the same time an explanation of the cardinalities in the last definition. I leave to a patient reader to reason that, again, the two relationships are not mutual rotations.

Definition 53 (Requirement-Agent Multi-edge) *The multi-edge* Requirement-Agent *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Agent) with the following meanings:*

(#Requirement)-s expressed by given (#Agent) / 0,M:1,M
(#Agent)-s accepting given (#Requirement) / 0,M:0,M
(#Agent)-s antagonistic to given (#Requirement) / 0,M:0,M
(#Agent)-s neutral to given (#Requirement) / 0,M:0,M

To develop a Service System the first relationship is the most important. But to tune the Service System behavior within competitive environment the next three relationships are beneficial.

The cardinality of the first relationship expresses that each will or desire or item of wants has to have its author; there must be an agent who wants it.

Definition 54 (Requirement-Model Multi-edge) *The multi-edge* Requirement-Model *is a container for connections connecting particular instances of (#Requirement) to particular instances of (#Model) with the following meanings:*

(#Requirement)-s reflected and/or conceptually specified by given (#Model) / 0,M:0,M.

(#Model)-s created to reflect and/or conceptually specify given (#Requirement) / 0,M:0,M.

(#Requirement)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.

Note, that in the relationship

(#Requirement)-s reflected and/or conceptually specified by given (#Model) / 0,M:0,M.

the requirements are *used*, while in the relationship

(#Requirement)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.

the requirements are *mentioned*. Thus, these relationships are very different each from the other.

The first two relationships are again not mutual rotations. The first one is about requirements forming details of the model. The second one is about a requirement being satisfied by using the model.

—

Multi-edges of the second phase of the Fourth Diamond step by step building. In the second phase we complete the picture by a first portion of circumference multi-edges and the first diagonal multi-edge (connecting opposite vertices).

Definition 55 (Use-case-Goal Multi-edge) *The multi-edge* Use-case-Goal *is a container for connections connecting particular instances of (#Use-case) to particular instances of (#Goal) with the following meanings:*

(#Use-case)-s describing by example given (#Goal) in a procedural manner / 0,M:0,M.

(#Goal)-s inspired by given (#Use-case) / 0,M:0,M.

(#Goal)-s used in given (#Use-case) / 0,M:0,M.

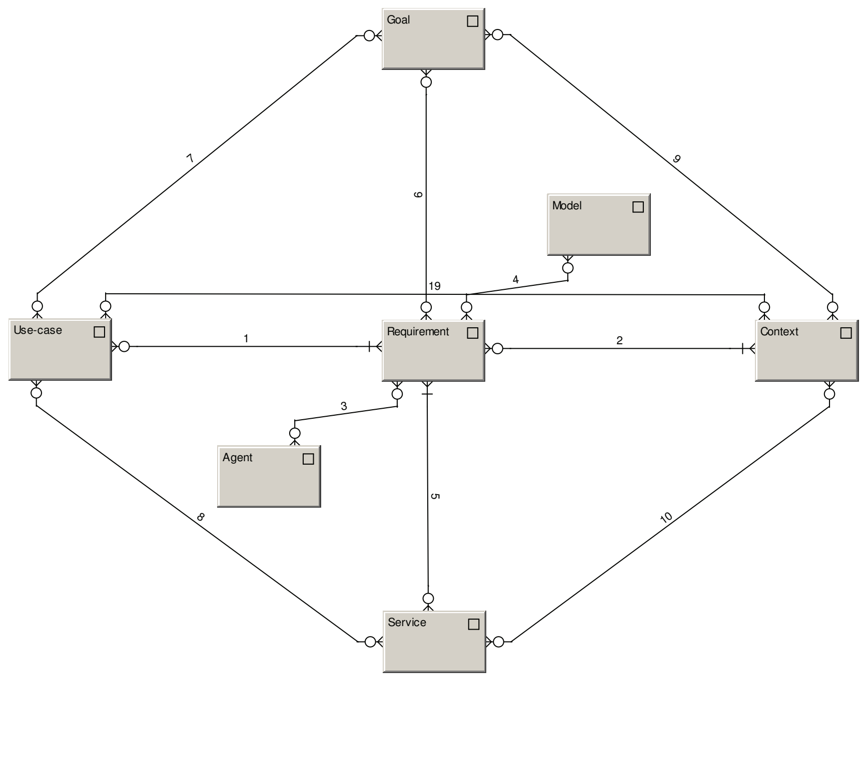

Figure 5.7: Diamond of Predictive Behavior - phase 2 of the concept

The first of these relationships helps better understanding what a given goal is about. As a goal could be a value proposition it is a way to explain by example the matter of fact described by this proposition. Often we understand better the situation when some scenarios of possible developments are given.

A scenario itself can inspire new goals.

Sometimes a step by step reaching of particular goals can visualize a scenario.

Definition 56 (Use-case-Service Multi-edge) *The multi-edge* Use-case-Service *is a container for connections connecting particular instances of (#Use-case) to particular instances of (#Service) with the following meanings:*
(#Service)-s that support given (#Use-case) / 0,M:0,M.
(#Use-case)-s describing given (#Service) / 0,M:0,M.
(#Use-case)-s which are inspirations for given (#Service) / 0,M:0,M.

When a specification of how to reach given goal is turned to implementation we need to describe how the solution bringing service will be used. Again the use-cases are appropriate tools for to do it comprehensibly. The first two relationships serve to this purpose.

The last relationship is an important vehicle to the future Service System innovation.

Definition 57 (Context-Goal Multi-edge) *The multi-edge* Context-Goal *is a container for connections connecting particular instances of (#Context) to particular instances of (#Goal) with the following meanings:*
(#Context)-s in which given (#Goal) occurs / 1,M:0,M.
(#Goal)-s typical for given (#Context) / 0,M:0,M.

Each goal has to appear in some context. It is excluded anomaly a goal without context; such goal couldn't be understand and will be completely for nothing.

The second relationship is necessary to understand a context well. Often we have no possibility to list all elementary beliefs (see definition 16) of context in section 5.2) composing the context or it wouldn't be understandable for our partners–other agents. Be aware that lot of contexts in our life we know or understand only through these typical goals.

Definition 58 (Context-Service Multi-edge) *The multi-edge* Context-Service *is a container for connections connecting particular instances of (#Context) to particular instances of (#Service) with the following meanings:*
(#Service)-s applicable and useful in given (#Context) / 0,M:1,M.
(#Context)-s in which given (#Service) can be executed / 0,M:0,M.

Always we need to know in which context a service can be executed, i.e. the service is completely prepared for execution and the execution depends on Clients decision only. Be aware that during the Service System life cycle there are situations in which there is no context in which a given service can be executed–simply it is not prepared yet.

The first relationship supports again the service innovation. A service is applicable and could be useful in a context but it is not prepared yet to this. At least in one context a service must be applicable and useful as in other case such service has no sense.

If the service can be executed in context K then it will be applicable and useful in that context K.

The last one is the diagonal multi-edge:

Definition 59 (Context-Use-case Multi-edge) *The multi-edge* Context-Use-case *is a container for connections connecting particular instances of (#Context) to particular instances of (#Use-case) with the following meanings:*
(#Use-case)-s created in given (#Context) / 0,M:1,1.
(#Context)-s in which given (#Use-case) can be applicable and/or interpreted / 1,M:0,M.
(#Use-case)-s typical for given (#Context) / 0,M:0,M.

Each use-case is created in just one context. But this use case is possibly applicable in other contexts. At least in one context it is applicable–in this one in which it is created.

Again, be aware that lot of contexts in our life we understand mostly through their typical use-cases.

The third phase amends the rest of multi-edges and the diamond shape of our model starts to be visible. As agents are known from the previous Diamond of Organization, let's start with agents's multi-edges.

Definition 60 (Agent-Use-case Multi-edge) *The multi-edge* Agent-Use-case *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Use-case) with the following meanings:*

(#Agent)-s playing a role in given (#Use-case) / 0,M:0,M.

(#Use-case)-s by example describing processes of given (#Agent) / 0,M:0,M.

(#Use-case)-s created by given (#Agent) / 0,M:0,M.

First, agents can be actors in the scenario described by given use-case. Second, processes performed by an agent could be better understood when

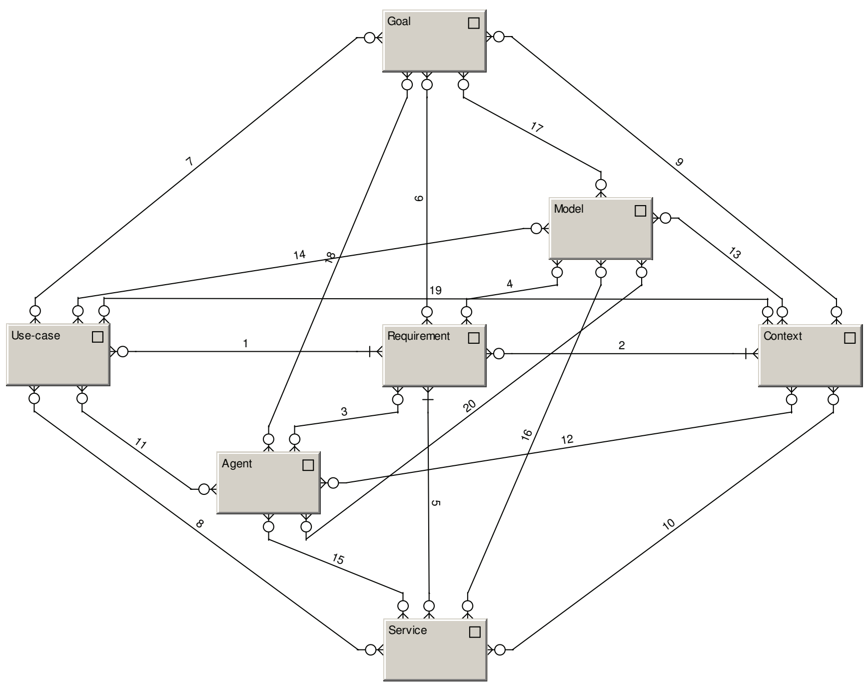

Figure 5.8: Diamond of Predictive Behavior - phase 3 of the concept

described by use-cases.

The last relationship gives a possibility to subscribe each scenario by its author. This is important from the continual improvement of Service Systems point of view.

Definition 61 (Agent-Goal Multi-edge) *The multi-edge* Agent-Goal *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Goal) with the following meanings:*
(#Agent)-s following given (#Goal) / 0,M:0,M.
(#Agent)establishing given (#Goal) / 0,1:0,M.
(#Agent)-s interested in filling up given (#Goal) / 0,M:0,M.
(#Agent)-s necessary for filling up given (#Goal) / 0,M:0,M.
(#Agent)-s antagonistic to given (#Goal) / 0,M:0,M.
(#Agent)-s neutral to given (#Goal) / 0,M:0,M.

The first two relationships are known from Diamond of Organization, see section 5.3.

The agents which are interested or necessary or antagonistic or neutral to a given goal have to be recorded for the sake of the environment of the provided services deeper understanding.

Definition 62 (Agent-Service Multi-edge) *The multi-edge* Agent-Service *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Service) with the following meanings:*
(#Agent)-s which are clients of given (#Service) / 0,M:0,M.
(#Agent)-s which are providers of given (#Service) / 0,M:0,M.
(#Agent)-s which are involved as co-operators for producing given (#Service) / 0,M:0,M.
(#Agent)-s requesting given (#Service) / 0,M:0,M.
(#Service)-s applicable and useful for given (#Agent) / 0,M:0,M.

Agent–client and agent–provider are basic roles of agents within a Service System. Agents–co-operators for a given service is a widespread relationship helping to record various kinds of contributions to a service provision.

Agent requesting a service is a potential client and it is important to know him. Agent for whom a service could be useful is interesting from a marketing point of view.

Definition 63 (Agent-Context Multi-edge) *The multi-edge* Agent-Context *is a container for connections connecting particular instances of (#Agent) to particular instances of (#Context) with the following meanings:*

(#Agent)-s operating in or being important for prospection of given (#Context) / 0,M:0,M.

(#Agent)-s typically occurring in given (#Context) / 0,M:0,M.

(#Context)-s helping to understand behavior and/or nature of given (#Agent) / 1,M:0,M.

The second relationship helps to understand better a given context.

The other four multi-edges will be those coming from the container (#Model).

Definition 64 (Model-Use-case Multi-edge) *The multi-edge* Model-Use-case *is a container for connections connecting particular instances of (#Model) to particular instances of (#Use-case) with the following meanings:*

(#Use-case)-s conceptually analyzed in given (#Model) / 0,M:0,M.

(#Use-case)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.

(#Model)-s playing a role in given (#Use-case) / 0,M:0,M.

Definition 65 (Model-Goal Multi-edge) *The multi-edge* Model-Goal *is a container for connections connecting particular instances of (#Model) to particular instances of (#Goal) with the following meanings:*

(#Model)-s describing given (#Goal) / 0,M:0,M.

(#Goal)-s arising on the base of deep understanding to given (#Model) / 0,M:0,M.

(#Goal)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.

Definition 66 (Model-Service Multi-edge) *The multi-edge* Model-Service *is a container for connections connecting particular instances of (#Model) to particular instances of (#Service) with the following meanings:*

(#Service)-s described by given (#Model) / 0,M:0,M.

(#Model)-s contributing to understandability of given (#Service) / 0,M:0,M.

(#Service)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.

Definition 67 (Model-Context Multi-edge) *The multi-edge* Model-Context *is a container for connections connecting particular instances of (#Model) to particular instances of (#Context) with the following meanings:*

(#Context)-s in which given (#Model) can be interpreted / 1,M:0,M.

(#Context)-s prospected and/or specified by given (#Model) / 0,M:0,M.

(#Context)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.
(#Model)-s describing some important features of given (#Context) / 0,M:0,M.

The last multi-edge connecting (#Model) container with (#Agent) container is the second diagonal multi-edge at fig. 5.10.

Definition 68 (Model-Agent Multi-edge) *The multi-edge* Model-Agent *is a container for connections connecting particular instances of (#Model) to particular instances of (#Agent) with the following meanings:*
(#Agent)-s contributing to given (#Model) / 0,M:0,M.
(#Agent)-s encapsulated into entity sorts of given (#Model) / 0,M:0,M.
(#Model)-s describing some important features of given (#Agent) / 0,M:0,M.

A question arises: why there is no multi-edge between (#Goal) and (#Service) containers?

The container (#Goal) belongs to a planning half-space of the Fourth Diamond and the container (#Service) belongs to a realization half-space. These two half-spaces are naturally connected by (#Requirement). The remaining four containers, namely (#Use-case), (#Context), (#Agent) and (#Model), belong to both half-spaces. Goals and services could be interconnected through any of those five categories. A pragmatical and methodological reasons lead to the non-existence of direct connection between goals and services[14].

„Alibi relationship“. Now we are familiar with 20 multi-edges of the Fourth Diamond. From the chapter 4 on Cyberspace and its Content we know that it is not easy to declare a conceptual model to be complete. A future research very probably brings new relationships. That's why we introduce our „alibi relationship“. We will simply assume that in each of the described 20 multi-edges there exists appropriate relationship with the semantics according to the pattern:

(#X) which is in not specified relationship to given (#Y).

Future research can introduce new specified relationships which „will be derived“ from this one not-specified.

—

[14]when the R-edges will be explained we will see it is not necessary at all

Composition relationships. There exist 7 composition relationships, each one to each container. Note, that they are not multi-edges but they are usual conceptual model edges. These relationships express the whole-part relationship. The first three on the vertical axis of the diamond are explained by the following definition:

Definition 69 (Breakdown Structures) *The edge* GBS *is a container for Goal Breakdown Structure connections with the following meaning:*

(#Goal)-s to which given (#Goal) is directly decomposed / 0,M:0,1.

The edge RBS *is a container for Requirement Breakdown Structure connections with the following meaning:*

(#Requirements)-s to which given (#Requirement) is directly decomposed

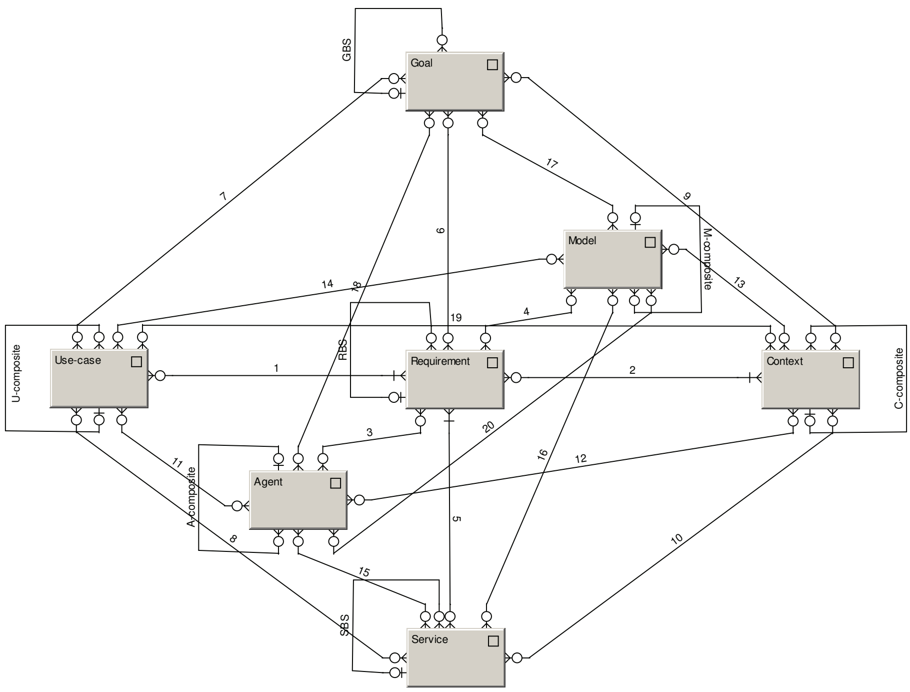

Figure 5.9: Diamond of Predictive Behavior - phase 4 of the concept

/ 0,M:0,1.

The edge SBS *is a container for Service Breakdown Structure connections with the following meaning:*

(#Service)-s to which given (#Service) is directly decomposed / 0,M:0,1.

The Relationship GBS we know from the Third Diamond, see definition 36. Here it is the same relationship. Analogically we have RBS and SBS relationships. Namely SBS, the Service Breakdown Structure, will appear a very important for building Service Systems from other Service Systems.

The remaining four composition relationships are:

Definition 70 (Composition Structures) *The edge* U-composite *is a container for Use-case Composition Structure connections with the following meaning:*

(#Use-case) of which given (#Use-case) is a direct part / 0,1:0,M.

The edge C-composite *is a container for Context Composition Structure connections with the following meaning:*

(#Context) of which given (#Context) is a direct part / 0,1:0,M.

The edge A-composite *is a container for Agent Composition Structure connections with the following meaning:*

(#Agent) of which given (#Agent) is a direct part / 0,1:0,M.

The edge M-composite *is a container for Model Composition Structure connections with the following meaning:*

(#Model) of which given (#Model) is a direct part / 0,1:0,M.

When a deep understanding of the relationship

(#Context) of which given (#Context) is a direct part / 0,1:0,M

is needed, we can use the Second Diamond. The Agent Composition Structure is known from the Third Diamond. The Model Composition Structure was spoken of in chapter 4 about cyberspace content.

—

The whole picture of the Diamond of Predictive Behavior is at the figure 5.10. We obtain it from the fourth phase picture by adding R-edges–the identity edges crossing boundaries of different contexts. Again, as in previous diamonds, only this vehicle give us the space of higher objects, i.e. the space of reasoning and cognition.

Recursive R-edges are similar to those one of the Diamond of Organization.

Definition 71 (Recursive R-Relationships) *The relationship* r11 *is a container for connections connecting particular instance of (#Goal) to another particular instance of (#Goal) with the meaning:*

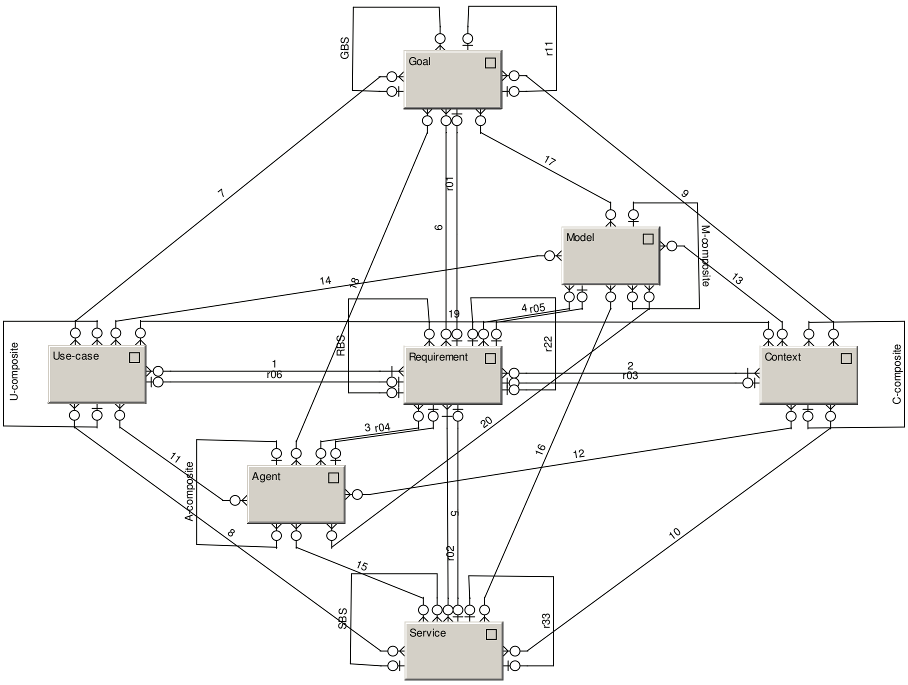

Figure 5.10: Diamond of Predictive Behavior - the whole picture

(#Goal) which in other context represents the given (#Goal) /0,1:0,1.

The relationship r22 *is a container for connections connecting particular instance of (#Requirement) to another particular instance of (#Requirement) with the meaning:*

(#Requirement) which in other context represents the given (#Requirement) /0,1:0,1.

The relationship r33 *is a container for connections connecting particular instance of (#Service) to another particular instance of (#Service) with the meaning:*

(#Service) which in other context represents the given (#Service) /0,1:0,1.

Connections from these containers are called recursive r-edges.

The r11 edge is the same as the edge R22 in Diamond of Organization, see fig. 5.4.

The thinking process goes often from particular phenomena to an abstract conclusion. That is why we can for a non trivial time period recognize two separate requirements in two different contexts and then after a new information occurrence we recognize those two requirements are one and the same. Each of those two requirements is bounded in its own family of connections. What an agent can do in such situation? To forget one of those requirements and to re-arrange its family of connections to the reminding one? No, it is a very complicated and risky operation. The most simple, and I believe the same as our brain does, is the operation of joining these two requirements by the identity edge r22 crossing boundaries of respective contexts.

The same way it goes with services. Again very often there are developed two distinct services in different contexts, some time they live separately and after some new information they are connected by the identity edge r33 as they were recognized to be one and the same service.

Classical R-edges. I call it classical, as they are placed in the Fourth Diamond in the same manner as R-edges in the first two (technical) diamonds [15].

Definition 72 (R-Relationships) *The relationship* r01 *is a container for connections connecting particular instance of (#Goal) to particular instance of (#Requirement) with the meaning:*

[15] which were the first discovered diamonds

(#Goal) which in other context plays the role of given (#Requirement) /0,1:0,1.

The relationship r02 *is a container for connections connecting particular instance of (#Service) to particular instance of (#Requirement) with the meaning:*

(#Service) which in other context plays the role of given (#Requirement) /0,1:0,1.

The relationship r03 *is a container for connections connecting particular instance of (#Context) to particular instance of (#Requirement) with the meaning:*

(#Context) which in other context plays the role of given (#Requirement) /0,1:0,1.

The relationship r04 *is a container for connections connecting particular instance of (#Agent) to particular instance of (#Requirement) with the meaning:*

(#Agent) which in other context plays the role of given (#Requirement) /0,1:0,1.

The relationship r05 *is a container for connections connecting particular instance of (#Model) to particular instance of (#Requirement) with the meaning:*

(#Model) which in other context plays the role of given (#Requirement) /0,1:0,1.

The relationship r06 *is a container for connections connecting particular instance of (#Use-case) to particular instance of (#Requirement) with the meaning:*

(#Use-case) which in other context plays the role of given (#Requirement) /0,1:0,1.

Connections from these containers are called (classical) r-edges.

A *requirement* expresses a wants or needs, thus, in other words a will of persistence, a will of sustainability. *Requirement* is the central point of autonomy, as autonomy means an ability of the system to contribute to its own persistence, as was stated above. Everything important for a Service System sustainability is expressed by the diamond vertices here. Thus each of these vertices could turn to a *requirement.* End this is the meaning of classical r-edges.

This is very similar to equi-lying edges in first two diamonds, Diamond of Attention Focussing and Diamond of Cognitive Elements. Let's take a *connection* in the First Diamond, e.g. At the vertex (#Connection) we find it as it is, prepared to be directly used. When we speak of *connection*, i.e. we mention it as an *object*, we make a transition in the First Diamond along the R1 edge into the center of the diamond. The same it is with, e.g. a *goal*, in case of the Fourth Diamond. Again, in the vertex (#Goal) it is a *goal* as it is, completely prepared to be used. When we make a transition from the vertex (#Goal) to the diamond center (#Requirement) we can mention this *goal* in one selected manner: „I desire the *goal* is fulfilled“ or „I plan to fulfill the *goal*“ not so emotionally speaking.

Similarly it goes with other vertices of the Fourth Diamond, and similarly it could be compared to the Second Diamond.

⋇ [STA]$_{118}$:TODO

5.4.2 Executive Point of View

Taking into account the holonic nature of Service Systems, i.e. the fact that Service System can be viewed as a part of another bigger Service System and at the same time this Service System can be viewed as a whole composed of smaller Service Systems, and taking into account the fact that we, human beings, are also Service Systems, we can conclude: when modeling Service systems, we need to be able to record

- connections
- contexts
- certainty
- attention
- composites of the previous mentioned

and to be able to revise such models, accordingly to world changes.

Provided that Service Systems are primarily business systems, we are interested in such models of Service Systems, in which solutions, i.e. *services*, are sought to a set of *requirements*, where the sets of requirements are heading to some *goals*. The whole „game“ is played in a domain (a *context*) with a set of *agents* (natural or artificial ones) involved in, can be described by some *use cases*, and anything of the above mentioned can be remembered again in a form of *models*. This led us to deployment of a generic conceptual model for Service Systems, namely the Diamond of Predictive Behavior, when facing to services execution and Service Systems operation.

This model (shown on fig. 5.10) conforms with previously presented Service System definition (see section 3.2) and provides uniform top-level view that can be applied to any Service System. I believe that thinking about Service Systems by means of this model can help us perceive particular Service Systems uniformly in terms of one framework, which may be rewarding, for example, in composition of complex Service Systems from several others. Complete description of this model was done in previous subsection 5.4.1.

The features of a Service System that are accentuated by this model, could be summarized as follows: Service system has to be autonomous, co-operative, compose-able, and adaptive. To be useful, its behavior has to be predictive and oriented in various contexts. This means it has to be able to identify and deliver services achieving given goals in conformity with particular requirements and domain or context specifics. It should „understand“ use-cases describing typical or important situations. It should „communicate“ and „co-operate“ with agents in the game involved. And finally, it should „remember“ relevant structural connections and behavioral operations in a form of such models which are context-sensitive, attention aware and certainty-sensitive.

Now let us combine this model with previously introduced context-sensitivity. We can assign contexts for example to individual Service Systems and then naturally compose more sophisticated Service Systems by combining information from relevant contexts.

Another application of contexts may be modeling of intersection of several Service Systems. Two Service Systems can intersect in their *targets*, service *providers* and service *clients*. Then we are able to look at the intersecting agent, which is always the same for both Service Systems, in context of any of the systems and thus see the important features of this agent that are relevant just for that particular system. Finally, using contexts, we can naturally work with an agent playing the role of service client in one context and the role of service provider in another.

As was already mentioned, the presented model describes a uniform framework for thinking about Service Systems. I am convinced that such a tool helps its users to focus attention on important aspects common for most of Service Systems.

Since service systems often involve provider and client cooperation in order to create value, and mutual cooperation is more likely to be successful, when solution is described in terms comprehensible for both, provider and client, conceptual models appear to be more important for service system design compared with product design. Conceptual models capturing semantics assist in consistent understanding of service domain to provider and client. Another benefit of modeling with semantics lies in easier management of service system changes during system evolution in time.

Goals modeling for Service Systems. There are established goal-oriented requirements engineering techniques (⋇ [STA]$_{119}$:cite Lamsweerde, 2001) and goals are being used for identification of individual services in the goal-service modeling method patented by IBM (⋇ [STA]$_{120}$:cite Ang et al., 2008). These approaches conform to the proposed conceptual framework, as it allows describing goals refinement as well as requirements and use-cases derived from individual goals.

Customers are not buying goods or services but value propositions (⋇ [STA]$_{121}$:cite Gummesson, 2007). According to (⋇ [STA]$_{122}$:cite Maglio and Spohrer, 2008), the key to understanding the nature of sharing arrangements among entities within service systems lies in the distribution of competences and the value propositions that connect entities. These statements illustrate the importance of value proposition and I believe that further elaboration of modeling the semantics of value proposition using tools such conceptual modeling and reference conceptual model of Service System behavior in combination with goal-oriented methods may prove fruitful.

Goal-refinement graph provides useful connection between (verbose) value proposition and individual services. The graph can be transformed into so called Service Breakdown Structure (see the top and the bottom vertices of Diamond of Predictive Behavior and fig. 6.6), inspired by work breakdown structure defined in project management (⋇ [STA]$_{123}$:cite PMI, 2008). Explicit connection of services and goals with partial value propositions adds structure to information provided by value proposition. The service breakdown structure enables for instance traceability of value proposition parts that are affected by unavailability of a particular service.

⋇ [STA]$_{124}$:TODO: lepe a jinak rozebrat Diam4 z pohledu managementu a marketingu

How the Fourth Diamond helps to strategic, tactic and operational management.

What is its role in business development process.

How it can be used in systematic marketing.

What are the benefits of this model and this approach.

How the „Do“ in S-R-O-D scheme is performed.

⋇ [STA]$_{125}$:vyuzit reversed triangle inequality for information capacity v Diams Path a specialne v Diam4, goal formulation, value proposition formulation, requirements assessment, use-cases revision, model preparation and revision

5.5 Diamond of Diamonds–the Fifth Diamond

The four diamonds, Diamond of Attention Focussing, Diamond of Cognitive Elements, Diamond of Organization, and Diamond of Predictive Behavior were conceptual models. Conceptual models help us, and help to any agent, to remember what is important for an autonomous behavior and for sustainability.

The first of them helps to perceive time-space and cyberspace–to see *objects* there, their *connections*, similarities which head to categorizations into *categories*, what happens with these objects, connections and categories, i.e.*operations*, and repeating behavior or structures, i.e. some *rules* of the world. All this is remembered as a content of the First Diamond containers. Thus the Diamond of Attention Focussing is about **seeing** or watching the world. This is the first step in the continuous cognitive process of any autonomous agent. The central point of seeing is *Object*.

The Second Diamond is a model of remembering elements of cognitive process itself. To be persistent and to show autonomous behavior an agent has to recognize *items* in „its mind“ (whatever it is), i.e. in its mental space,

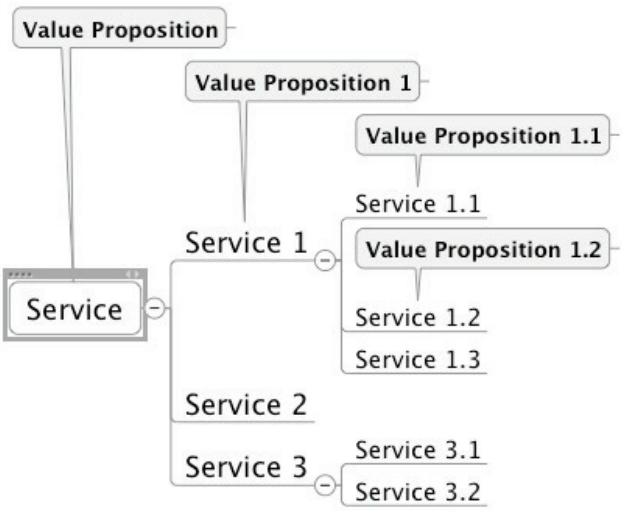

Figure 5.11: Service breakdown structure

simply in cyberspace, and hooks in a way similar items to marked ones by which it provides a process of categorization. *Categories* in itself are remembered together with connections of particular items into these categories, i.e. with *CI-connections*. To be able to evaluate situations, *contexts* have to be remembered. And each particular judgement on an item's belonging to a category together with the importance of this fact, i.e. with attention paid to it, which is in fact a *manifestation* of this judgement, has to be remembered, too. As the mind is recursive, and it enables to think of any composed structure as of an item, the memory enabling to remember cognitive elements has to have this recursiveness embedded. The Diamond of Cognitive elements possesses these features and thus it is about **recognizing** of what was seen. The central point of recognizing is *Item*.

Seeing and recognizing support the MENTION mode of behavior of an agent but help nothing to the USE mode. The Diamond of Organization represents a memory in which organization of sustainable processes is remembered. Non-autonomous components can be built into complexes while autonomous agents can be organized into complex structures–teams of agents and agents of a higher order. Organizing is this what turns agents to USE mode. Organizing covers *agents*, how they deal with *activities*, what they *goals* are, and what are the long-lasting *flows* of attention to behave goals oriented and sustainably. As we assume here the cognition as an enactive process dependent not only on *mentioning* but on *using*, too, the **organizing** is the third step in the continuous cognitive process. The central point of organizing is *Agent*.

To do something with no idea what happens when this something will be done is not usable at all. Action consequences have to be predictable. For this purpose an agent needs to remember elements and aspects of actual and planned actions in both worlds, the real time space and the cyberspace. It needs to remember *requirements*, the *goals* to which particular sets of requirements are headed, *services* or solutions which satisfy the requirements and fulfill planned goals, and to do it in environment formed by agents, contexts, models and use-cases. It is necessary to remember these *agents*, *contexts*, *models*, and *use-cases* in particular connections to the given requirements, goals and services which gives the whole picture of agents predictive behavior. The doing means to be in USE mode. Again, as the cognition is an enactive process, we see the **doing** to be a fourth step of continuous cognitive process. The central point of doing is *Requirement*.

This four conceptual models form a scheme that could be called **See–Recognize–Organize–Do scheme of enactive cognition process**. I will refer to it as to S-R-O-D scheme.

The Fifth Diamond, that will be described now, is not a model in a sense the previous Diamonds are. This is in a way „a gyrocompass of our (and any other agent's, too) mental ship going through the cyberspace".

A team of agents forming a Service System should understand environment, situation, and themselves. That is why it needs to know

- where to go
- how to reach its goals and objectives
- what deliverables must be produced
- how to organize themselves to produce those deliverables
- how to perform „the play"

The Fifth Diamond, or Diamond of Diamonds, is a conception model that leads us, and can lead any autonomous and co-operative agent, through the life-cycle. This not about structures and containers for remembering but it is about attention concentration dynamics which helps „to live and to cognize". My opinion is, that this could be effectively done by harmonizing the attention paid to particular topics of the S-R-O-D scheme.

Let's come to this Fifth Diamond, to its full picture step-by-step, with a step-by-step explanation.

Reminding the definition 2 of Service System in chapter 3 and concerning attention to the time axis we can conclude:

Statement 6 (Agent's live out) *An agent, whatever it is, lives out the* Present *on the base of the* Past *and with expectation of the* Future.

All decisions and all activities notwithstanding whether they are performed in the time-space or in the cyberspace are undertaken in the Present. This could be activities of the mode USE as well as activities of the mode MENTION. Thus it could be about doing something or about thinking something or speaking something. Speaking something includes recording a message of something. This is the meaning of „live out" in the statement.

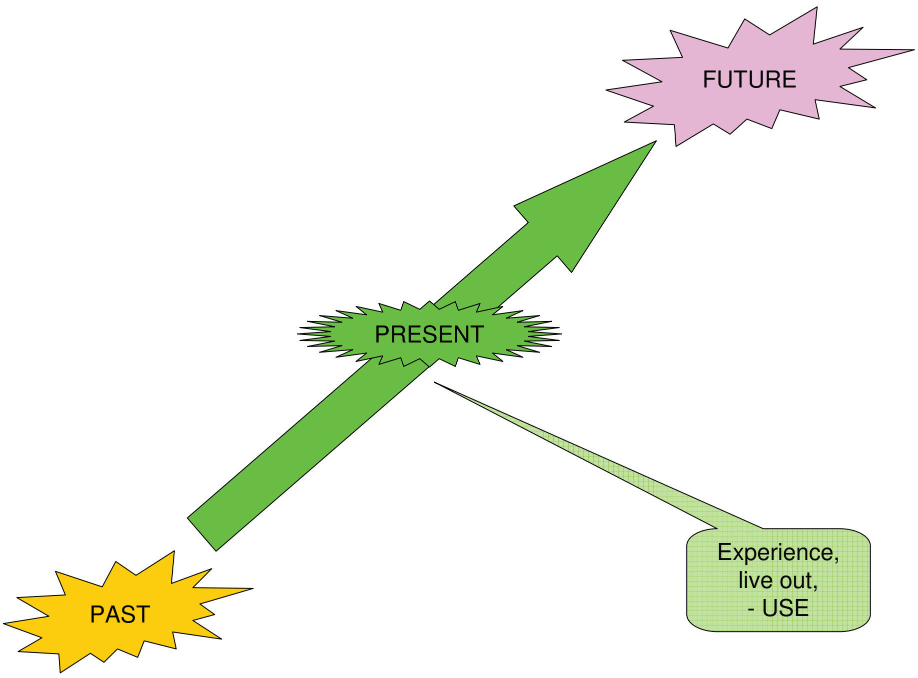

Figure 5.12: Diamond of Diamonds - phase 1: life in time

As there is a history of such undertakings and a Service System is provided with a memory, i.e. it remembers the history, all current decisions and activities are performed on the base of this history. If the history is much more about the time-space we will speak about experiences, if the history is much more about cyberspace, we will speak about plans. Such plans can be implicit ones, i.e. not recorded as formal plans, or explicit ones. Implicit plans can be simple ones–ideas or complex ones–ideal models of any type. Explicit plans are recorded models described according to an appropriate formalism. Experiences and plans drive our present decisions and activities. This is what I mean by „based on the Past“ in the statement.

Service Systems and agents are endowed by goal orientation. This means their behavior is predictive one. It aims to something in the future. Present activities and decisions are made in expectation of a desired or wanted state-of-affairs. Such expectations were prepared by another activities in the MENTION mode. All thinking on the Future is in the MENTION mode. The continuum of activities and decisions from the Past through the Present to the Future is the central point to which an intelligent agent focuses its attention.

Note, that it is not possible to USE something in the Future, now, so as it is not possible to USE something from the Past, now. The USE mode is designated to the Present. The MENTION mode is designated to the Past and to the Future. Even if we say that we speak or just think of something in Present, it is a metonymy. In fact, exactly saying, the Present is a time point only; all around this point is in the Past or in the Future. But, so as we avoid the integral calculus expression in attention assignment to a pair (agent, activity) in section 5.3 by introducing the attention credit function, we avoid differential calculus contemplations here by introducing „a reasonable environment of the Present time point“ and we will deal with it as it is all the Present. In this sense the MENTION mode can refer to present, too.

The last important remark is: the USE mode doesn't imply that activities in this mode have to concern with time-space world, only. These activities can be as well concerned with the cyberspace.

To conclude the first phase of Diamond of Diamonds explication we can say: The axis Past-Present-Future represents the agent's or Service System's life in time. It is the axis of

- experience creating,

- activities and decisions in USE mode,
- simply of live out in each continuously changing present time point.

This is what the fig. 5.12 tells us.

The figure 5.13 shows how the Past-Present-Future axis is an axis of orbit on which the four diamonds circulate. The orbit represents

- visualization,
- the MENTION mode,
- simply think about.

And there the S-R-O-D scheme is materialized in this orbit. The usual order of making visualizations of a Service System situation is given by this scheme. When an activity take place after the step-by-step visualization

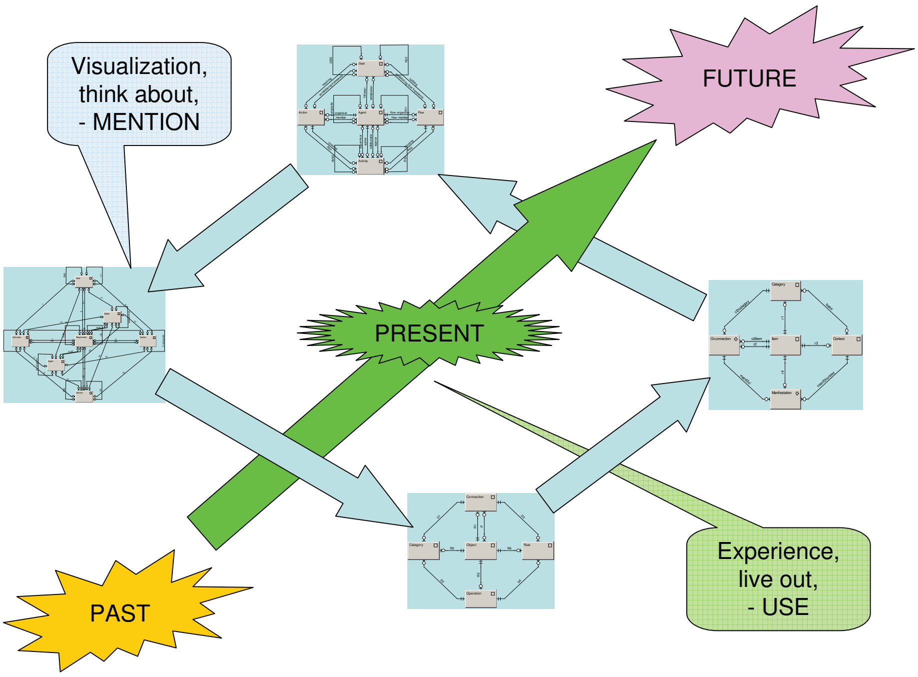

Figure 5.13: Diamond of Diamonds - phase 2: conceptual thinking of the situation

by means the four diamonds–conceptual models, the overall situation of the Service System will change. The realized activity cause it. And the new situation can be visualized in the same manner.

This is nothing else than the MENTION-USE switching which was spoken of in chapter 4 on Cyberspace and its Content and in section 5.1 where the First Diamond was described.

In other words it is an expression of enactive perception and enactive cognition. It can be seen well that without acting there is no cognition at all. And the MENTION-USE switching principle is the crucial phenomenon of continual cognitive process of a Service System.

Let's suppose the attention spread on the particular diamond is proportional to „the distance“ of the diamond from the observer, this means from you–the reader. It is in alignment with the idea that Service Systems are sources of attention. The more Service Systems dealing with an object, the more attention is paid to the object. Thus, diamond(s) on a shorter distance from the front is endowed by more attention than diamond(s) on a longer distance. And the diamonds circulate on the orbit. The message of fig. 5.14 is that the attention is periodically changed on the four diamonds.

I would like to remind the mills in project-program-portfolio management spoken of in sect. 5.3 in the project/program/portfolio management point of view. I explained there that the regularly repeating of managerial activities

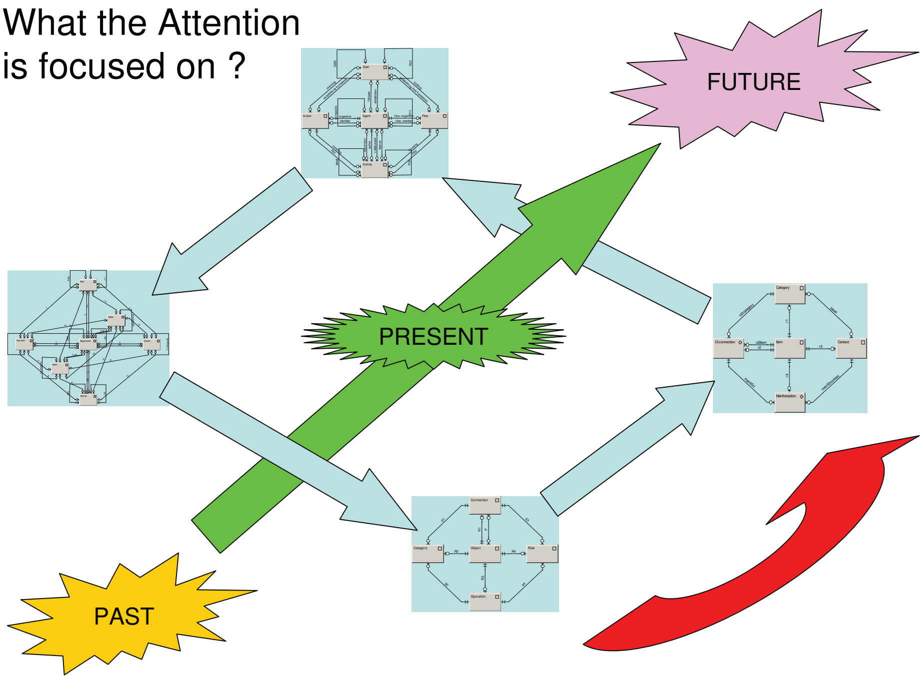

Figure 5.14: Diamond of Diamonds - phase 3: attention focussing to the situation

is necessary to keep the management process sustainable one. Here we can see it was not a single opinion based on single experience this mills approach. On the contrary this periodic changing of attention is a principle of agent's, and thus Service System's, too, autonomy.

Statement 7 (Agent's autonomy mechanism) *An agent, whatever it is, to be autonomous has to periodically focus this portion of its attention which it spreads in cyberspace to the diamonds circulating on the orbit around the Past-Present-Future axis in the order of the scheme of topics See - Recognize - Organize - Do.*

The Diamond of Predictive Behavior has two half-spaces: upper half-space is a planning one, lower half-space is a realization one. When attention is focused to this diamond, things to be done are planned and then executed in the time-space or in the cyberspace and together with this recorded in appropriate containers of this conceptual diamond.

I understand you can have objections to this statement. E.g., what proves the order of topics?, or must it be so regular? My answer lies in the following: the irregularity of the attention switching so as the order of topics could be modeled by changing the whole amount of attention focussed to the nearest diamond on the orbit. Limitary it could be a zero amount of attention: such diamond is escaped in the current cycle.

Since there was nothing said of the amount of focused attention in the statement 7, the model works. The statement 7 bring directions how to build an artificial agent and make it autonomous. In the time-space the orbits and circulating is natural. I see no reason why the same cannot be natural in the cyberspace. This is the explanation of the statement.

—stanoteTODO What about our brains? Are there any jumps?

What if periodically one particular of the S-R-O-D topics is excluded by a zero amount of attention?

TODO text

The topic oriented attention distribution is one possibility. The another possibility is to distribute attention according to a time point of view. There could be, e.g., a distribution driven by a calendar or by clocks. But the basic attention distribution is along the three time regions, namely the Past, the Present and the Future. Phase 4 of the step-by-step Diamond of Diamonds model building on fig. 5.15 shows the attention focused to the Present. This usually means activity performing or decision taking. This attention focusing corresponds with USE mode.

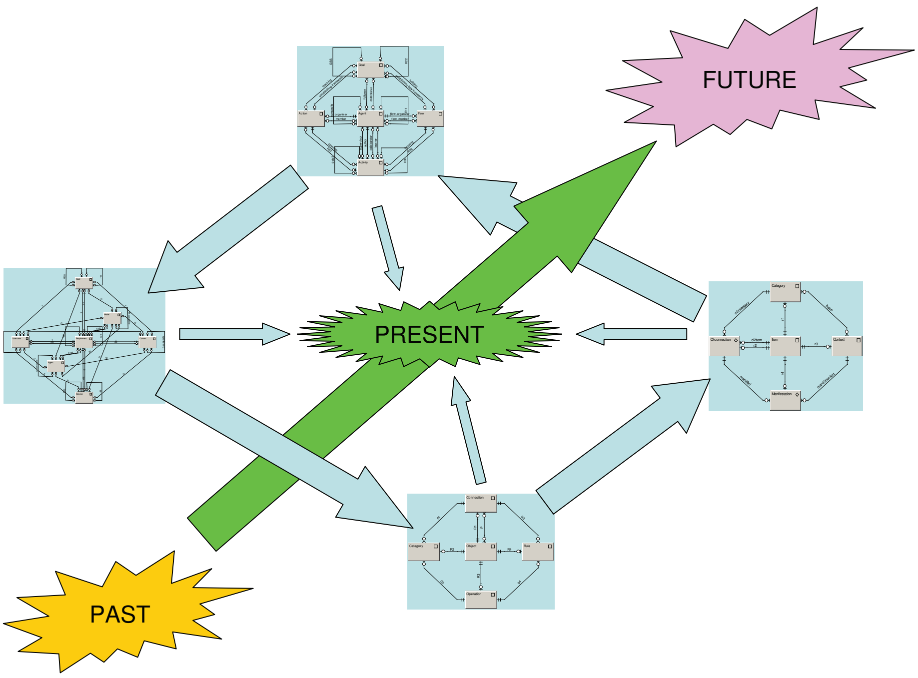

Figure 5.15: Diamond of Diamonds - phase 4: attention focussing to the Present

The next five phase on fig. 5.16 extends the attention distribution from the Present to both, the Present and the Future. Often this is the situation of planning and immediately executing of planned activities. It could be a kind of planning and execution cycle which could be named short-time planning. An operative management is full of such short-time planned and immediately executed activities.

At this situation no history from the Past is taking into account and this means poor time context is employed. A quick MENTION-USE switching is performed heading to short-time goals, probably better to say objectives.

One possible interpretation could be that a tactical management is involved, too, if our understanding of the Present is a wider interval around

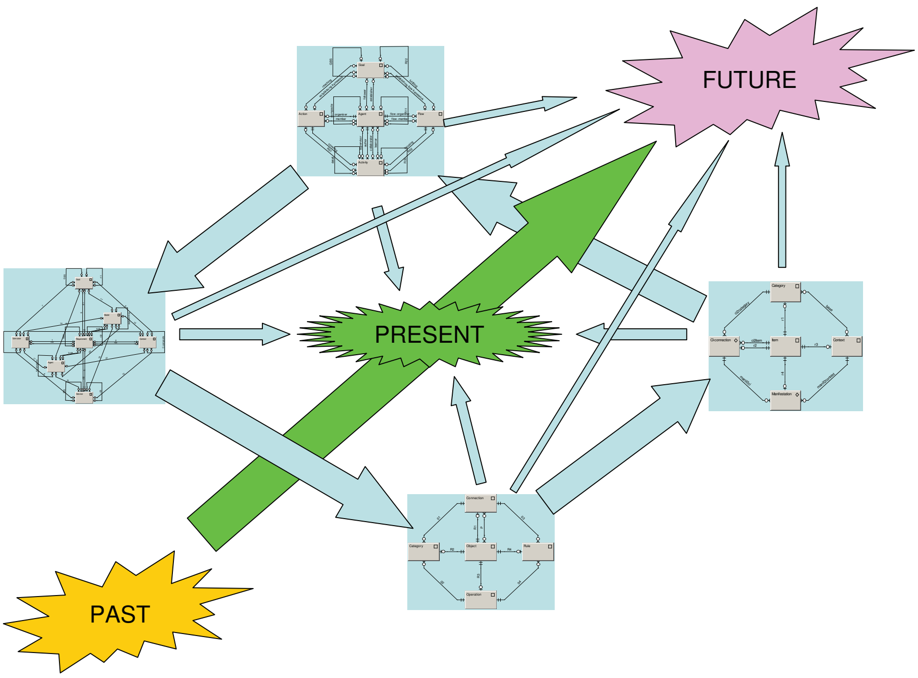

Figure 5.16: Diamond of Diamonds - phase 5: attention focussing to the Present and to the Future

the present time point.

Figure 5.17 brings the whole picture. The attention could be distributed wherever in the time continuum from the Past through the Present to the Future. According to an application domain, or generally a given context, concrete situation and selected strategy the attention can be dynamically redistributed. The dynamics of attention distribution is the phenomenon driving the Service System behavior.

The full model enables all kinds of management: from an operative one, through a tactic to a strategic one. The strategic management uses switching attention between the Future and the Past.

Now the research and state of art in the attention distribution is at the very beginning. I believe that a future research of this field brings lot of very

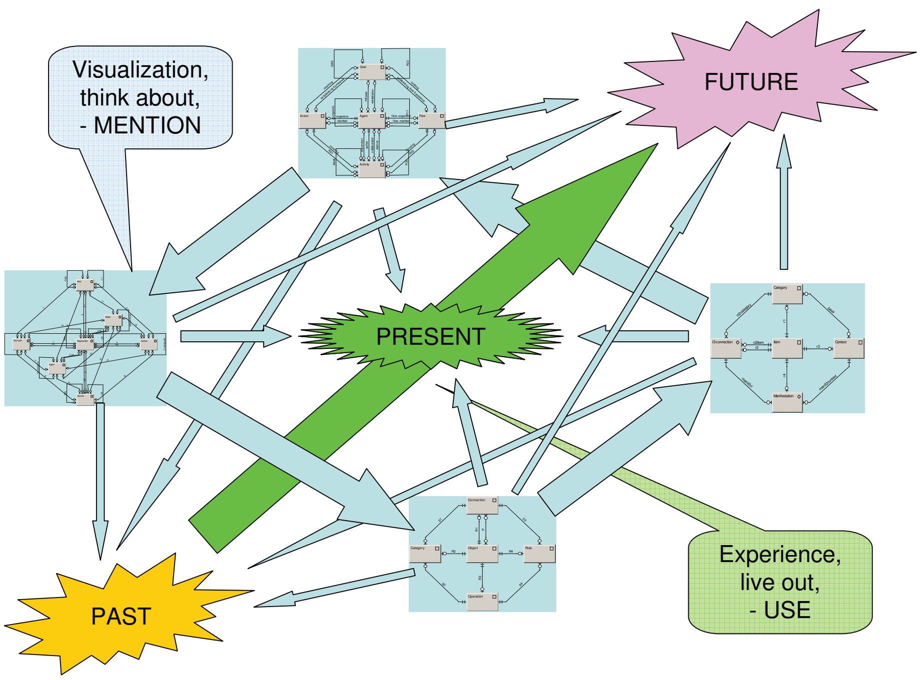

Figure 5.17: Diamond of Diamonds - full picture: attention focussing to the Present, to the Past, and to the Future

interesting and applicable results in the field of Service Systems and artificial intelligence.

A good metaphor to the Fifth Diamond could be „cyberspace-craft“ flying in the cyberspace from the Past to the Future revolving around the time axis which causes periodic attention focusing to particular diamonds. The amount of attention being at disposal is dynamically changed and depends on events from the surrounding cyberspace and/or time-space.

Service Systems are composites of people, technology, other systems, provided information and knowledge and covered by value proposition defining their purpose (cf. def. 2). They „see“ the time-space and cyberspace which could be modeled by the First Diamond (cf. sect. 5.1). They also „recognize“ what watched in the time-space and cyberspace, they recognize meanings of facts and build semantics in appropriate contexts from elementary beliefs, which could be modeled by the Second Diamond (cf. section 5.2). Service Systems organize themselves to be able better response to their environment to be autonomous in a sense of contributing to their own sustainability (cf. sect. 5.3). Service Systems perform actions autonomously to achieve predicted and desired results; they plan for it, they execute plans and they „remember“ plans and reality, „compare“ it to revise plans or develop new ones, simply they perform the contribution to their sustainability. This last could be modeled by the Fourth Diamond.

So, the Service Systems are hybrid systems, hybrids of human beings in a way organized with artificial systems all together composing in a way living organisms, however, these organisms are decentralized. Anyway we can speak of their lives. The Diamond of Diamonds– conception model together with the four diamonds described in previous sections–conceptual models form a possible model of Service Systems life.

Each concrete filling of the all diamonds containers by appropriate instances brings a particular service system as an instance of that model.

Attention and its dynamics is the phenomenon which help us to recognize whether a Service System is alive.

Statement 8 (Service System's being alive) *Service System is alive if the function describing attention distribution in a time-point is not constant in a continuing time interval, i.e.*

$\partial F_{att}(t)/\partial t \neq 0$ *for* $t \in \langle R, \infty)$,

where R *is a time point in the Past.*

Thus, any living being could be seen as a Service System. And there are context, in which a group of living beings in a way organized forms a Service System, too. The important message is that the attention plays a crucial role in Service Systems lives.

Entrepreneurship activities and actions are in each particular case accompanied by risks. The more risky action the less probability to reach the aimed goal. Thus whatever could decrease the risk is very important from the entrepreneur's point of view.

Statement 9 (Risk decreasing statement.) *Let there is a set of agents with at least one commonly shared goal, i.e. each of these agents wants to reach the goal. Then the probability of success, i.e. the probability that each agent of the set reaches the goal, will be greater under the following conditions:*

1. *each agent in the set wants to reach the goal together with the others,*
2. *agents form a Service System the conception of its functioning and behavior is driven by Diamond of Diamonds.*

A plausible reasoning proof of this statement could be done following the principles of proof by contradiction in maths. I try to lead you through this „proof" in the following paragraphs.

Team is more powerful than individual. The first condition violation. Let's assume each agent do it's best to achieve the goal. The expression „want to reach the goal together" means they behave in mutual co-operation and mutual dependence. If one of those agents is excluded from the set the probability of success of the remaining set of agents cannot be greater than the one of the original set of agents. The excluded agent behaves independently on the rest of agents. Whatever the probability of success of the excluded agent is, the resulting probability of the success of both, the one separate agent and the remaining set of agents, as they are independent phenomenons, is less than the original probability of success of the whole set of agents. This is evident: the product of two numbers between zero and one is less than each of the factors. Thus the violation of the first condition leads to less probability of success.

Second condition violation What means a violation of the second condition? One possibility is to forget harmonization in to the Present, to the Future and to the Past attention concentration. The other possibility is to forget to leave the attention to circulate over the four diamonds. The first one is simple and I devote to it the next paragraph. The second one is much more complicated, so it will be dealt with in the left fourth paragraphs.

Time dimension insufficient attentiveness. Let's start with the Present, Future, Past attention focusing in the conception model: If attention is not enough focused to the Present, the system will do nothing, it just will be out of condition to act. If attention is not enough focused to the Future, no reviews and no corrections of initially set partial goals will be performed and the road to the desired goal starts to be illusion only. If attention is not enough focused to the Past, the results of reviews and corrections of partial goals on the road to the desired goal will lack any feedback, lessons learned, experience from the Past. In all cases we are watching probability of success decreasing.

The second possibility how to violate the second condition is to stop with attention circulation on the four conceptual models represented by the four diamonds. This means at least one of this models is not seriously taken into account.

Diamond of Predictive Behavior insufficient attentiveness. First, let the Diamond of Predictive Behavior is not focused sufficiently. Some of the seven topics, namely

- Requirements
- Goals
- Services
- Contexts
- Use-cases
- Agents
- Models

is not well recognized and/or appreciated or some of relationships between them described by the Fourth Diamond is missed.

Not well recognized requirements and/or their relationships evidently leads to damaging the way to success (to reach the desired goal).

When goals are missed or not well recognized or in a way confused there is no possibility to be successful. But missing goals is a very often cause of non well working teams, not effective behavior of organizations, bad atmosphere in project teams, not well functioning systems. Such situations are repaired by various management actions and the poor results are pled by many words. All this done with no positive result. The problem lies in goals establishing. A crucial condition is: the goals must be established from inside of the Service System. This means agents of the Service System have to establish goals, have to believe this goals. The environment could bring some inspiration but it cannot establish the goal for the Service System. There could be goals and the Service System can decide to follow them. But it will be the same as if the Service System establishes the goals. When there is any kind of disorder in goals, the probability of success is going down.

The third topic are services. When a Service System strive for desired goal there exist special behavioral expressions of it giving sustainability to its path. These behavioral expressions interact with the Service System's surrounding producing benefits for particular agents in its environment. These sustainability supporting special behavioral expressions are services. But often a service is not cerebrated of. The actors think and communicate of poor products, only. And their mistake is in groundless belief that delivering a product is enough in serving a customer. The problem is there are no direct benefits for client when poor product is delivered only. The client's benefits come only when service is provided. Misunderstanding in this issues decreases the probability of success.

Contexts and the fact that in any situation there can be found lot of contexts, from which perspective the situation could be evaluated, are the most confusing issues on the road to the goal. It is important to know that not a message itself is of top importance but the shared context is. A typical situation on a project is: a message is send „from one context to another context", i.e. the sender doesn't ensure it's context will be shared by the receiver. Usually a big misunderstanding appears. Inappropriate work with contexts decrease the probability of success.

Use-cases serves to better understanding of what matters. Use-cases are scenarios of possible or future processes. Omitting use-cases or not clear description of them leads to misunderstandings and again decreases the probability of success.

There are agents inside and outside of Service System. Each of them has its own wants and desires. Reaching a common goal is a project as we know from the Third Diamond. Some of those agents are stakeholders for this project. The first rule of project management success is well done stakeholders management. Inappropriate focus to agents causes low probability of success, i.e. common and shared reaching of the goal.

Last memory container in the Fourth Diamond is the container for models. Models when remembered and than used help to understand a matter of facts of the problems solved step by step on the road to the goal. Models are tools for understanding. Nobody of us understand anything having no model of it. Shared models represent common dictionary helping in mutual communication of agents and in thinking process of each particular agent, too. Moreover, shared dictionary/models helps the team of agents thinks on common and shared topics which is important for co-operation. Models bring order into cyberspace; ideas are sorted into containers and then they can be systematically processed. Everything is pigeon holed so as in the great brain of Mycroft Holmes ⋇ [STA]$_{126}$:cite A.C Doyle povidka s Mycroftem Holmesem. Not well created model, which is in a sense inappropriate, leads to mistakes in visualization and in thinking. Absent common and shared model leads to a situation in which each agent operates according its private model which it creates to understand state of affairs. There could not be any co-operation in such situation. Thus underestimating of models role surely decrease the probability of success.

Let' take the edge Agent–Service, e.g. It represents the following relationships:

(#Agent)-s which are clients of given (#Service) / 0,M:0,M.

(#Agent)-s which are providers of given (#Service) / 0,M:0,M.

(#Agent)-s which are involved as co-operators for producing given (#Service) / 0,M:0,M.

(#Agent)-s requesting given (#Service) / 0,M:0,M.

(#Service)-s applicable and useful for given (#Agent) / 0,M:0,M.

(See definition 62.) If any of these relationships is underestimated the probability of success will decrease: let's take the

(#Agent)-s requesting given (#Service) / 0,M:0,M;

not working with this relationship we can simply miss a possible business.

The same way we can argue in case of each of the 36 edges of the Diamond of Predictive Behavior. I leave it to a patient reader as an exercise. I ask you urgently to try to do this argumentation process with some of those edges. This is the only way to start to understand „the message“ of the Fourth Diamond. By this exercise you will advance in building the appropriate context to understand Service Systems deeply.

As a last pattern I show here one of the R-edges which are a little bit mysterious. Let us consider the relationship

(#Context) which in other context plays the role of given (#Requirement) /0,1:0,1.

A requirement is a record documenting a single need of what a particular product or service should be or should do (cf. definition 46).

As the context is a sequence of manifestations (cf. definition 16), i.e. of elementary beliefs that say an item belongs to a category with a measure of certainty and a given attention is paid to this fact, it expresses a state of affairs. Such state of affairs can be a desired one in another context; this is the meaning of the R-edge r03. Focusing this edge contributes to the approach „attention is paid to all connections without artificially created boundaries“. As I argued in chapter 3, where Service Systems were defined, boundaries produce misunderstandings in agents communication and in agents thinking. Misunderstandings lead to less probability of success.

Diamond of Organization insufficient attentiveness. Second, let the Diamond of Organization is not well focused. The plausible reasoning proof continues in the same manner as we do it in case of Fourth Diamond. Some of the following topics

- Agents
- Goals
- Activities
- Actions
- Flows

is not well recognized and/or appreciated or some of relationships between them described by the Third Diamond is missed.

Let's note, the attention is concentrated to organization of the Service System, now. Paying inadequate attention to agents will cause organizational mistakes. Thus, the whole will functioning worse and the probability of success will go down. It is easy to see the same is true with goals, activities, and actions, as they are wrappers for activities.

Remaining topic is Flows. This is typically missing topic in a range of projects or programs or portfolios being recently solved. Let's go again through the definition 24 of flow and the following part of the section 5.3. Considering carefully the description there we can conclude the probability of success will decrease if the flows are omitted or inadequately focused.

Inappropriate focusing of the Third Diamond relationships means not well performed project management, i.e. some shortcomings in working in alignment with project management standards. Each such shortcoming is widely recognized to be a cause of success probability decreasing. But is it true with the diamonds relationships focusing? Let's turn to figure 5.4 of the Third Diamond. The Agent–Goal relationships are clear: not well established goals or agents not oriented to follow given goals–this is a poor project management. The four Agent–Activity relationships are more interesting. They carry the organizational game with attention planned and really paid to particular activities. They support evidence of planned and done work. The motor of acting are requirements as we know from the Fourth Diamond. Requirements are produced by agents. And the requirements convenient to a successful fulfillment of the goal could be well supported by appropriately set attention assignment to (agent, activity) pairs as well as they could be damaged by unappropriate setting. Thus, the game with learners, collaborators, authors and supervisors directly influences the success. Moreover, we can see the importance of attention switching between the Third and the Fourth Diamonds. The remaining edges can be considered in a similar way; I leave this reasoning to a patient reader.

Diamond of Cognitive Elements insufficient attentiveness. Third, let the Diamond of Cognitive Elements is not focused explicitly and in sharing mode between agents of the Service System. The meanings of both, the structural and the behavioral patterns are unclear and each agent has its own opinion about it. What is worse the agents do not know that there are big differences and using marginal indirect indicia they believe they understand patterns in the same way. A great misunderstanding can appear.

Some people try to overcome this trouble by seeking one common true: e.g. one common type system enabling to classify all things and all issues into one order. Unfortunately these attempts are not successful. Another extreme is, when one common classification is not possible, than we resign the endeavor to understand meanings or to model semantics in artificial systems. In this case we do not know what a semantics understanding is for a particular agent. Thus, we cannot simulate it in artificial agents and lot of decisions made by human agents is not comprehensive, more of them is not comprehended. This is currently a typical situation which negatively influences the probability of success.

Diamond of Attention Focussing insufficient attentiveness. Fourth, let the Diamond of Attention Focussing is not focused. Than we miss the universal structure enabling to remember anything new for the (artificial) agents and being the base for common dictionary helping communications of (human) agents. To understand a domain we need technical terms. These technical terms name objects of our interest, their specific connections, operations that can be done with these objects, categories ordering these objects, and rules of the whole game. Particularly, in each domain there occurs a need to focus to some connections or operations or categories or rules as to objects. This way the First Diamond leads our attention, generally agents attention, to understand the domain well. At the same time it contributes to sharing seen pictures between agents. Models dwelling in the (#Model) container of the Fourth Diamond will be constructed more effectively if this approach is taken. One can miss this opportunity when thinking of one problem and one model. But more and more we are faced to necessity to compose two or more existing models into a new one, a composite model. If the models were created separately without the Diamond of Attention contribution, and the composition itself is done ad hoc, such a composition could be difficult. So, again, the absence of attention or insufficient attention focused to the First Diamond decreases the probability of success.

I hope that the reason for and the possible utility of the Diamond Path Framework is now more clear. This will be the basis for Service Systems modeling explanation, services execution and Service Systems operating explication and than their education (both, education of a Service System and education for a Service System) and evaluation in next chapters. Accordingly, it could be a basis for Service Systems economic description and their evolution.

Bibliography

[1] Daniel Clement Dennett. *Kinds of minds: Towards an Understanding of Consciousness.* Basic Books, 1996.

[2] Marie Duží. Semantics information connected with data. In *Database Theory - ICDT'92, Lecture Notes in Computer Science*, number 2, pages 376–390. Springer-Verlag, 1992.

[3] Marie Duží. Propositional attitudes & synonymous expressions. In T. Childers, P. Kolář, and V. Svoboda, editors, *LOGIKA'96*, pages 309–321, Prague, 1996.

[4] Marie Duží. Logical foundations of conceptual modeling, 2000.

[5] Marie Duží, František Krejčí, Pavel Materna, and Zdenko Staníček. Method of the database design (a functional approach to information representation). Research report, Brno University of Technology, Brno, 1986.

[6] Marie Duží, Pavel Materna, and Zdenko Staníček. Information capability of a database schema. In *In: Proc. 10th International Seminar on DBMS*, Cedzyna Poland, 1988.

[7] Paul P. Maglio and Jim Spohrer. Fundamentals of service science. *Journal of the Academy of Marketing Science*, (36):18–20, 2008.

[8] Paul P. Maglio, Savitha Srinivasan, Jeffrey T. Kreulen, and Jim Spohrer. Service systems, service scientists, ssme, and innovation. *Communications of the ACM*, 49(7):81–85, 2006.

[9] Pavel Materna. Concepts and objects. In *Acta Philosophica Fennica*, Helsinki, 1998.

[10] Alva Noë. *Action in perception (Representation and the Mind)*. The MIT Press, January 2005.

[11] Filip Procházka. Modelování kognitivních procesů přístupem konceptuálního modelování. Master's thesis, Masarykova univerzita, Brno, 2001.

[12] Jim Spohrer. Service research to design & innovate service systems, 2007. Presentation.

[13] Jim Spohrer, Paul P. Maglio, John Bailey, and Daniel Gruhl. Steps toward a science of service systems. *IEEE Computer*, 40(1):71–77, 2007.

[14] Jim Spohrer and Doug Riecken. Introduction. *Communications of the ACM*, 49(7):30–32, 2006.

[15] Zdenko Staníček. Universální modelování a jeho vliv na tvorbu is. In *Sborník konference DATAKON 2001*, Brno, 2001.

[16] Zdenko Staníček. *Univerzální modelování a konstrukce IS*. PhD thesis, Masarykova univerzita, Brno, 2003.

[17] Pavel Tichý. *The Foundations of Frege's Logic*. De Gruyter, Berlin-New York, 1988.

[18] Stephen L. Vargo and Robert F. Lusch. Evolving to a new dominant logic for marketing. *Journal of Marketing*, 68(1):1–17, 2004.

[19] John A. Zachman. A framework for information system architecture. *IBM Systems Journal*, 26(3):276–292, 1987.

List of Definitions