

Téma 9

SoaML a návrhy GIS v architektuře SOA, modelový příklad návrhu GIS

1	Vyjadřovací prostředky SoaML	2
2	Příklad návrhu GIS v SoaML.....	6
2.1	Zadání.....	6
2.2	Kontext navrhovaného systému v SoaML	6
2.3	Servisní kontrakty	7
2.4	Architektury participantů	8

1 Vyjadřovací prostředky SoaML

Pro modelování služeb využijeme modelovací jazyk SoaML, což je zkratka pro Service oriented architecture Modeling Language. Je to nadstavba modelovacího jazyka UML, která doplňuje tento jazyk o prvky a postupy pro modelování služeb. Specifikace jazyka zahrnuje metamodel a UML profil pro návrh služeb v rámci servisní architektury. Poskytuje také základ pro další rozšiřování a integraci s Business Motivation Model (BMM), Business Process Modeling Notation (BPMN) a dalšími metamodely. Současná verze jazyka disponuje následujícími modelovacími možnostmi:

- Identifikace služeb, požadavků na ně a vztahů mezi nimi
- Specifikace služeb zahrnující jejich funkční schopnosti, co se očekává od konzumenta, protokoly nebo pravidla pro jejich používání a informace vyměněné mezi producenty a konzumenty
- Definice producentů a konzumentů služeb, toho co vyžadují, jaké služby využívají či poskytují, jak jsou připojeni nebo spojeni a jak jsou funkční schopnosti služeb využívány resp. implementovány konzumenty resp. producenty ve smyslu konzistence se specifikací služby i s dodržnými požadavky
- Politiky používání a poskytování služeb
- Schopnost definovat klasifikační schémata beroucí zřetel na velkou škálu architektonických organizačních a fyzických schémat dělení a omezení



SoaML se zaměřuje na základní koncepty modelování služeb se záměrem dalšího rozšíření a použití s dalšími metamodely OMG (Object Management Group).

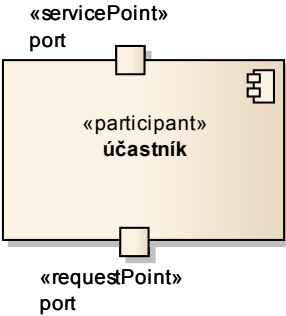
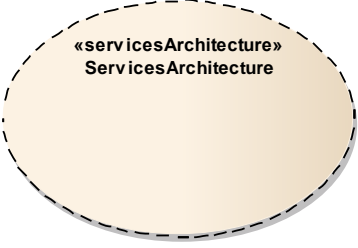

Klíčovým konceptem při modelování pomocí SoaML je služba, která je definována jako poskytnutí nějaké hodnoty jiné zúčastněné straně. Přístup ke službě je poskytnut na základě předem definovaného kontraktu, ve kterém jsou mimo jiné definovány omezení a politiky týkající se používání této služby. Služba je poskytována účastníkem vystupujícím v roli poskytovatele a spotřebovávána případným konzumentem.

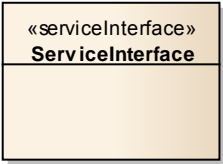
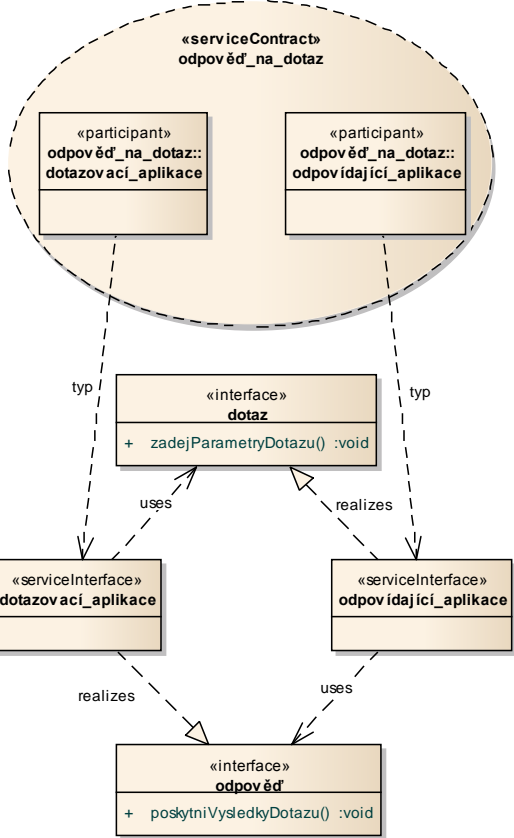

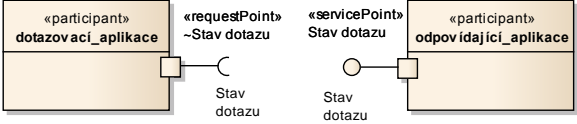
Při modelování se využívá převážně dvou diagramů:

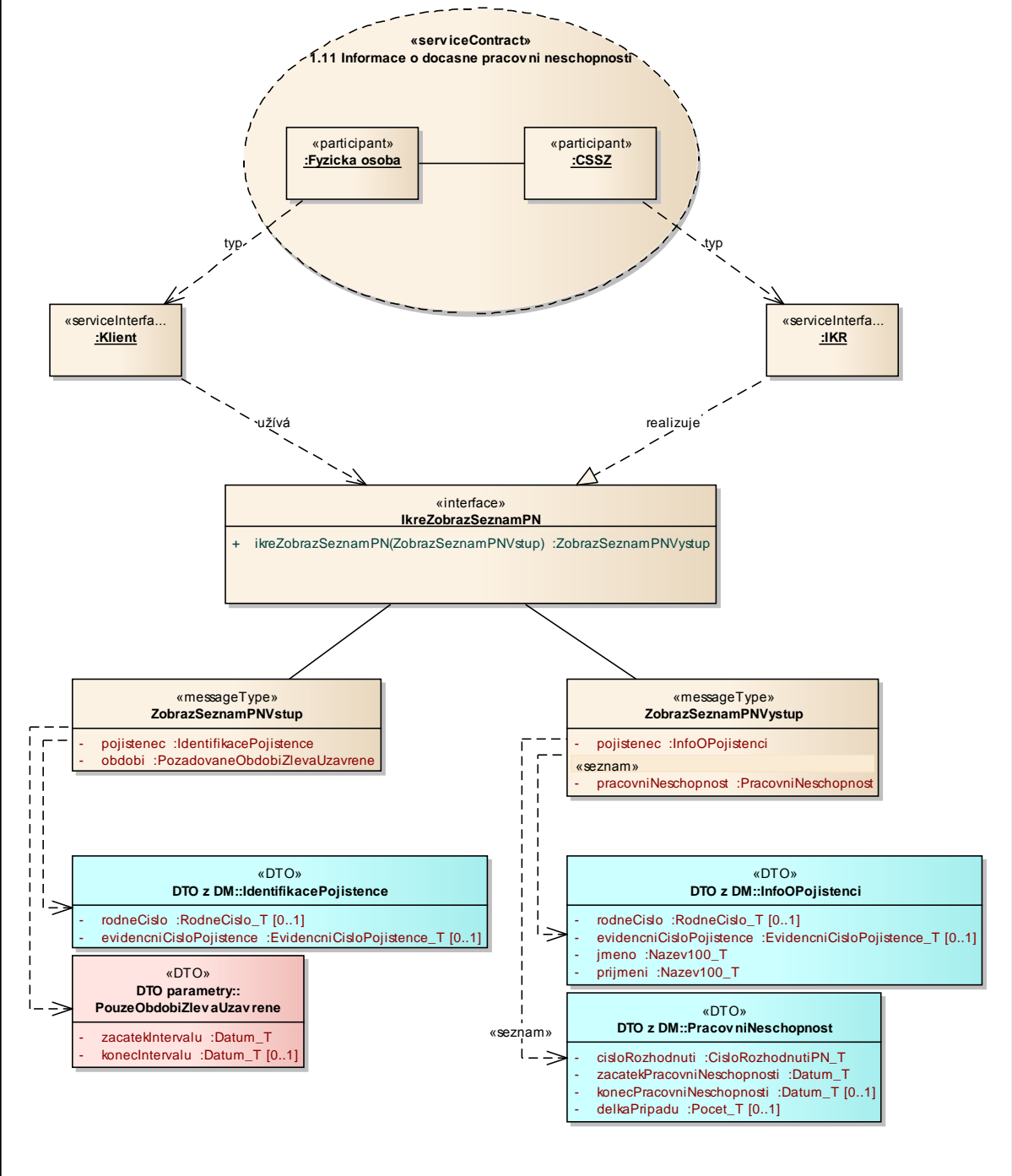
- Diagramu komponent, který vyjadřuje strukturu a architekturu služeb a účastníků
- Sekvenčního diagramu, který znázorňuje chování jednotlivých účastníků v rámci rozhraní služeb

Jednotlivé prvky diagramu komponent jsou popsány v následující tabulce, prvky sekvenčního diagramu v předcházející kapitole.

Objekt	Definice / Použití
<p style="text-align: center;">Participant – účastník</p> <div style="text-align: center;">  </div>	<p>Prvek Participant modeluje účastníka poskytujícího či odebírajícího jednotlivé služby. Je to například softwarová komponenta, organizace, systém nebo jednatel. Participant vystupují v metodice SoaML v několika rolích, každá se váže k jednomu kontraktu, kterého se účastní.</p>
<p>Participant Architecture – architektura účastníka</p> <div style="text-align: center;">  </div>	<p>Participant Architecture modeluje architekturu účastníka. Ta specifikuje složitější architekturu pro konkrétního účastníka systému, než je tomu u prvku Participant. Tento stereotyp ilustruje, jak komunikují účastníci na vyšší úrovni podrobnosti v kontextu daného účastníka, a je často složen z dalších architektur služeb nebo kontraktů služeb.</p>

Objekt	Definice / Použití
<p style="text-align: center;">Port</p> 	<p>Port modeluje místo komunikace služby nebo pouze bod, ve kterém je služba využívána nebo nabízena k využití účastníky. Na straně poskytovatele je port označován stereotypem service (servicePoint), na straně konzumenta stereotypem request (requestPoint).</p>
<p style="text-align: center;">Services architecture – architektura služeb</p> 	<p>Service architecture (architektura služeb) obsahuje síť účastníků a odpovídajících rolí, kteří poskytují a využívají služby s cílem splnění svých požadavků. Představuje pohled na systém na vyšší úrovni a zobrazuje vzájemně propojené a komunikující služby (kontrakty) spolu s účastníky.</p>
<p style="text-align: center;">Service contract – kontrakt služby</p> 	<p>Service contract (kontrakt služby) definuje podmínky, které musí účastníci přímo nebo nepřímo splňovat, aby mohla být služba vykonána. Modeluje dohodu mezi zúčastněnými rolemi. Důležité součásti kontraktu, které plně specifikují službu, jsou role, choreografie a rozhraní. Po řadě vyjadřují:</p> <ul style="list-style-type: none"> • role, které hrají participanti na dané úrovni služby • posloupnost informací vyměňovaných mezi stranami zúčastněnými v dané službě • způsob komunikace na portech účastníků <p>Na úrovni kontraktu jsou role dvojího typu: role konzumenta a role producenta.</p> <p>Choreografie je často modelována pomocí sekvenčního diagramu.</p>

Objekt	Definice / Použití
<p data-bbox="276 257 699 286">Service interface – rozhraní služby</p>  <p data-bbox="236 555 738 611">Vztah prvků service interface, participant a (UML) interface</p> 	<p data-bbox="798 304 1369 707">Prvek service interface je typem participantova portu. Service interface určuje odpovědnost participanta vzhledem k dané službě. Service contract vymezuje povinnosti všech participantů vzhledem ke službě, service interface je typ speciální pro každého participanta. Oproti klasickému UML rozhraní (interface), které lze pro porty také použít, má Service interface navíc schopnost definovat obousměrné služby, ve kterých má jak poskytovatel, tak konzument povinnost přijímat a odesílat zprávy a provádět akce. Toto rozhraní je definováno z pohledu poskytovatele a dělí se na tři části.</p> <ul data-bbox="798 728 1396 1485" style="list-style-type: none"> • poskytnuté a požadované rozhraní – jsou to klasická UML rozhraní (Interface), jedno je poskytovatelem realizováno a druhé užíváno. Realizované rozhraní (realized) slouží pro definování službou poskytované funkcionality a zpráv, které mohou být přijaty poskytovatelem (resp. odeslány konzumentem). Na obrázku je to rozhraní „dotaz“. Užívané rozhraní (used) definuje požadované schopnosti, zprávy nebo akce, které obdrží konzument (resp. pošle producent na konzumentovo rozhraní). Většinou se jedná právě o odpovědi na požadavky. Na obrázku je to rozhraní „odpověď“ • role účastníků – definuje, jaké role zastupují jednotliví účastníci služby. Jedna role odpovídá jednomu rozhraní. Roli, která je typována realizovaným rozhraním (z pohledu poskytovatele služby), hraje poskytovatel. Roli, která je typována užívaným rozhraním (z pohledu poskytovatele služby), hraje konzument služby. Role jsou identifikovány v kontraktech služeb (service contract) • chování specifikuje správnou interakci mezi poskytovatelem a konzumentem pomocí tzv. „protokolu interakce“ s využitím například sekvenčního diagramu či diagramu spolupráce dle standardu UML
<p data-bbox="247 1556 726 1585">Simple interface – jednoduché rozhraní</p>  	<p data-bbox="798 1648 1332 1821">Jednoduchá rozhraní definují jednosměrné služby, které nevyžadují protokol. Takové služby mohou být definovány pomocí jednoduchého UML prvku – interface a poté poskytovány na portu typu „service“ a konzumovány na portu typu „request“.</p>

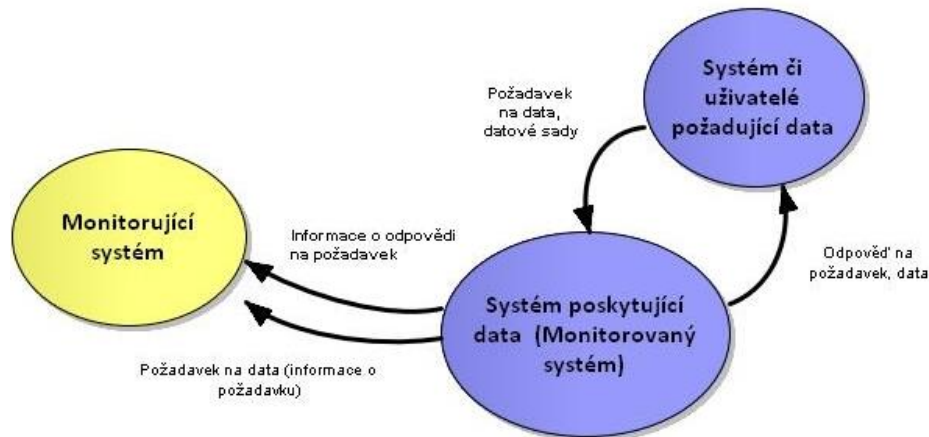


2 Příklad návrhu GIS v SoaML

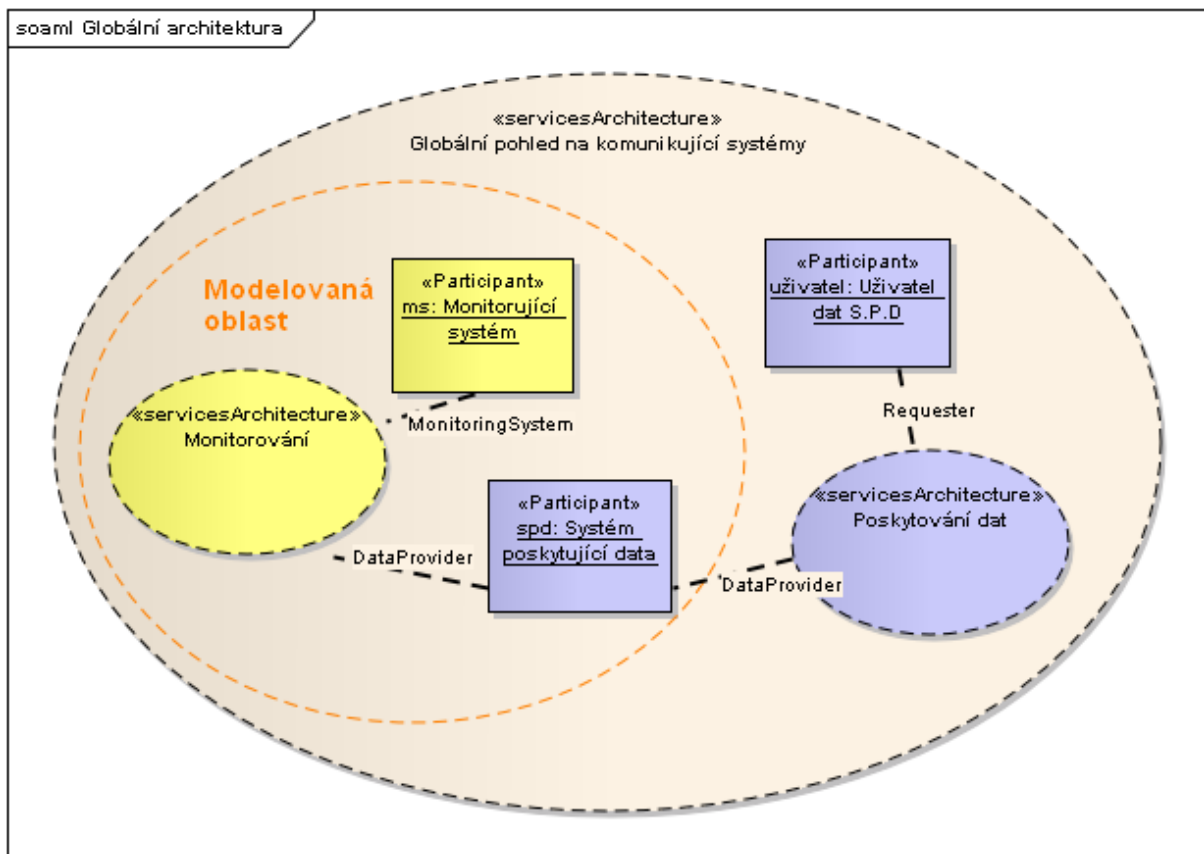
(převzato z DP Martina Mináře: Návrh systému pro monitorování poskytování prostorových dat)

2.1 Zadání

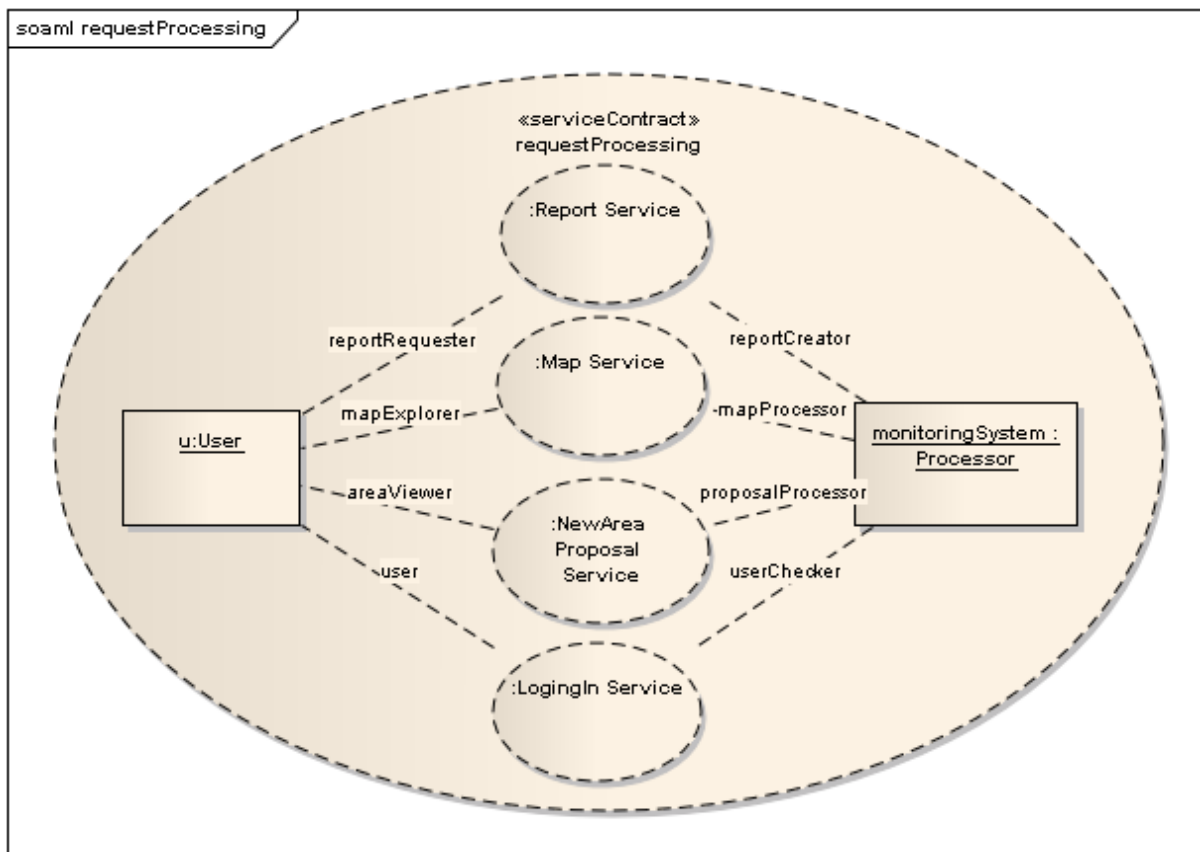
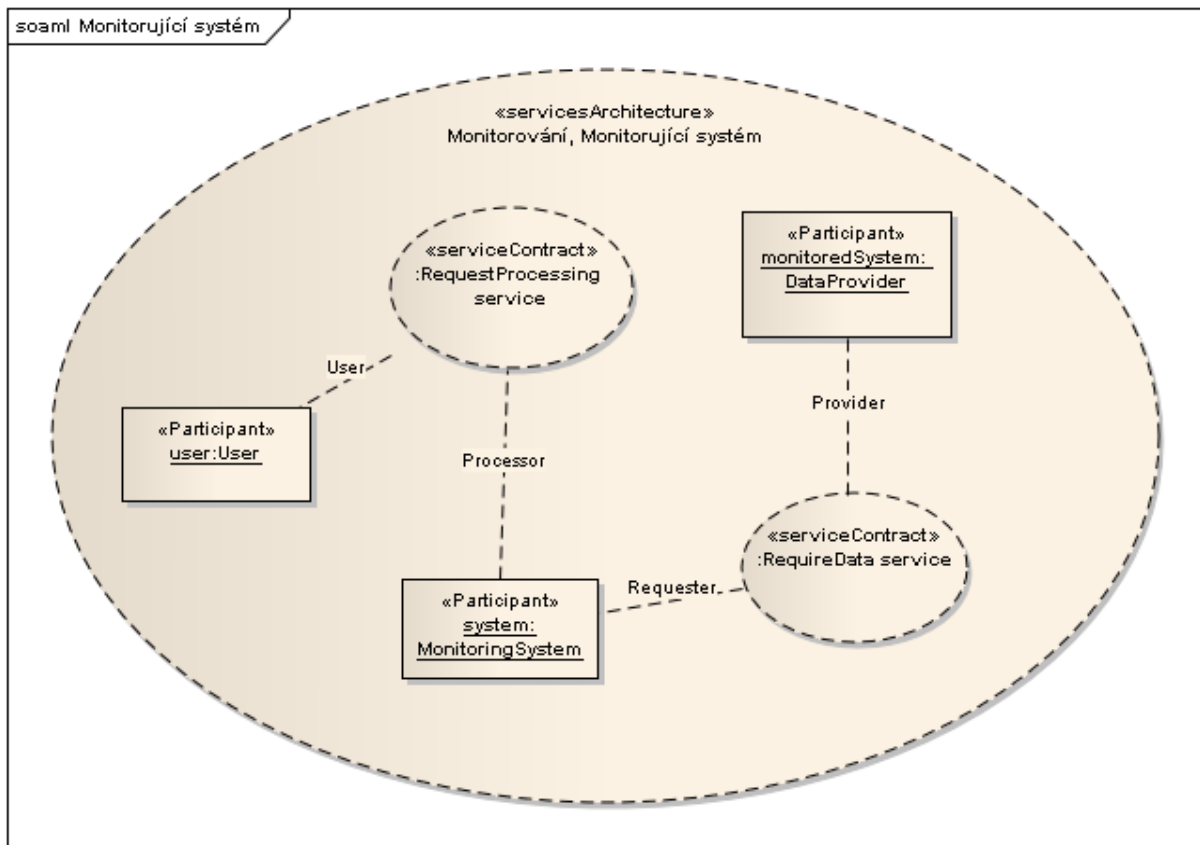
Návrh systému, který bude registrovat požadavky na poskytnutí prostorových dat a evidovat poskytnutá prostorová data externím systémem. Systém bude zobrazovat rozsah a typ požadavků na data i poskytnutých dat nad přehledovou mapou. Systém bude poskytovat podklady pro plánování aktualizace prostorových dat. S externím systémem poskytujícím data bude navržený systém komunikovat pomocí webových služeb. Návrh systému bude doplněn prototypem vybraných funkcí.

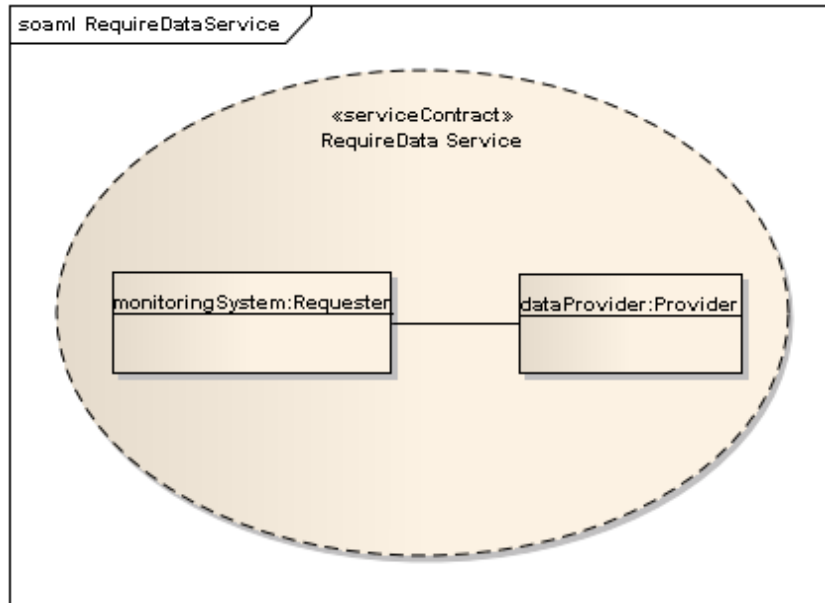


2.2 Kontext navrhovaného systému v SoaML

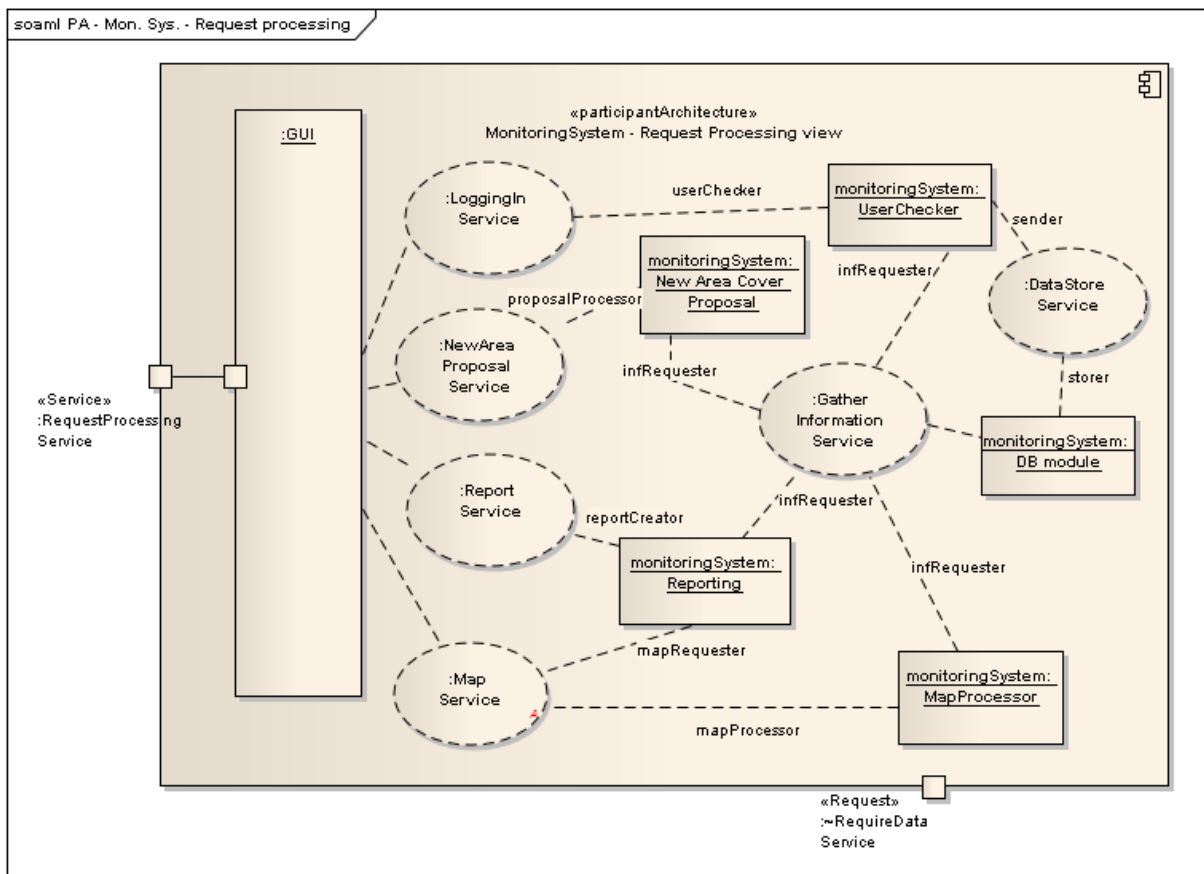


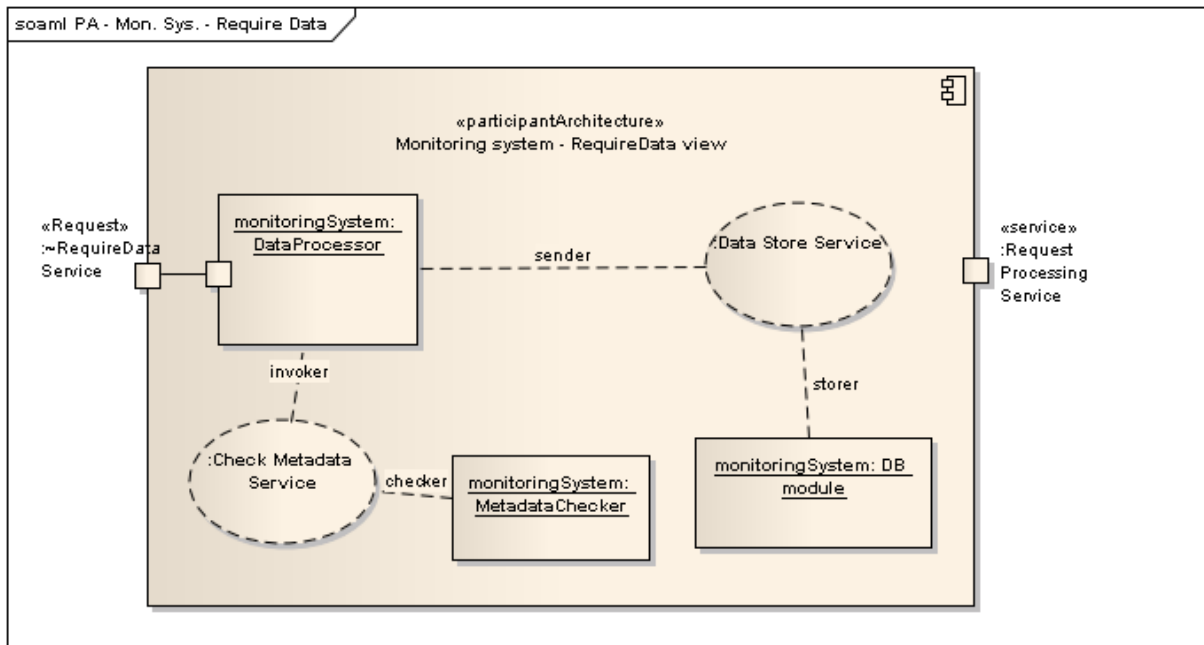
2.3 Servisní kontrakty





2.4 Arhitektury participantů





Připomínky a dotazy k obsahu lekce posílejte, prosím, na adresu:
 Rudolf Richter, richter@fi.muni.cz