

An Agile Model-Based Framework for Service Innovation for the Future Internet

Arne J. Berre

SINTEF, Oslo, Norway
arne.j.berre@sintef.no

Abstract. Service innovation for the future internet, with service design and service engineering, can benefit from a combination of an agile and model-based development approach. An agile approach is focusing on early understanding of user needs and service touchpoints to optimise for the best user experience and rapid adaptation to emerging user needs. A model based approach allows for a combination of various domain specific language that are suited for expressing services on different abstraction levels, aimed at supporting higher productivity and quality in service engineering. This approach presents an agile model-based framework using the emerging OMG domain specific standard languages VDML, BPMN, IFML and SoaML for service design and engineering, relating value models, process models, user interface and interaction flow models, and service architectures and service contract models. The associated methodology and set of practices can be supported by the emerging OMG FACESEM standard (a Foundation for the Agile Creation and Enactment of Software Engineering Methods).

Keywords: Future Internet, Agile, Model-based, Service Innovation.

1 Introduction and Motivation

To better support the creation of innovative services for the future internet, it is suggested to combine the practices and techniques of the service design [1] and the software engineering communities [2]. Service innovation for the future internet, with service design and service engineering, can benefit from a combination of an agile and model-based development approach. An agile approach is focusing on early understanding of user needs and service touchpoints to optimise for the best user experience and rapid adaptation to emerging user needs. A model based approach allows for a combination of various domain specific language that are suited for expressing services on different abstraction levels, aimed at supporting higher productivity and quality in service engineering. The different domain specific languages can be selected to support the different perspectives required for a holistic system specification, based on the perspectives identified in various Enterprise Architecture frameworks. These frameworks typically describes a system both on different abstraction levels and from different stakeholders view, as well as from different perspectives like structure, function, process, data etc.. In order to support agility, the description languages need to support different abstraction levels.

2 Agile Model-Based Framework

There is a number of Enterprise Architecture frameworks with a multi viewpoint approach for abstraction levels and perspectives (Zachman., RM-ODP, TOGAF, UPDM). The ASD (Agile Service Development) framework [2] is one that also includes user interaction as a separate perspective in addition to structure, function, coordination, information and extra functional aspects. The abstraction levels goes from requirements to design, implementation and infrastructure. Figure 1 illustrates a proposal for how different recent modeling languages can be used to support the various perspectives. Each of these comes with associated development practices. The suggested modeling languages are all existing standards from OMG, OASIS or W3C, or emerging languages in this context. These languages are well suited for their particular perspectives, but currently not designed to work harmonised together, and sometimes also contains more concepts and complexity than needed for a certain abstraction level. It is thus suggested to start an effort to create a coherent set of co-working languages with a foundation in the conceptual model of the existing languages. The agility emphasizes strong interactions with users and practitioners in dynamically adapting the models to emerging user needs. The emerging FACESEM/ESSENCE standard [6] from OMG (Foundation for the Agile Creation and Enactment of Software Engineering Methods) can be used to support the composition and agile adaptation of such practices.

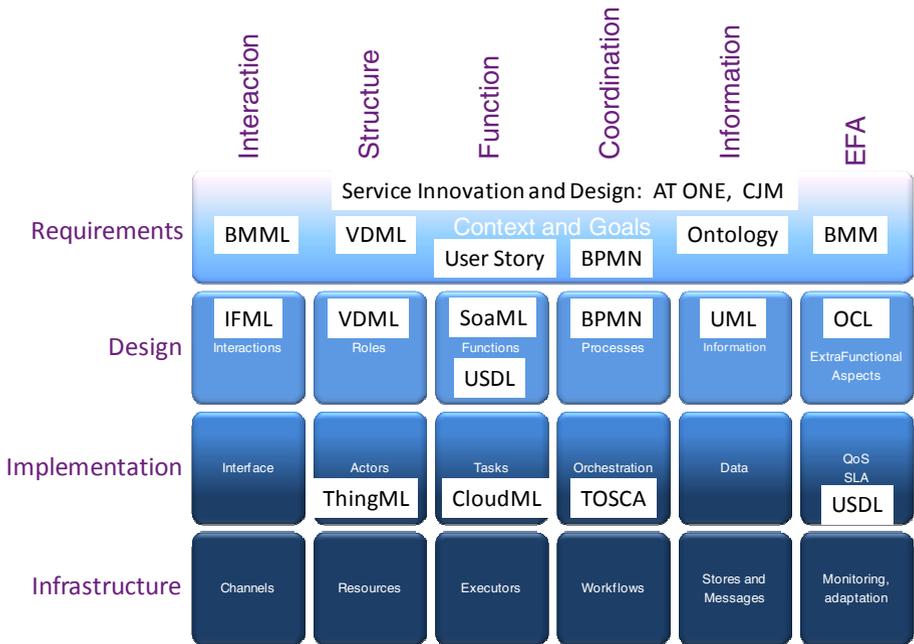


Fig. 1. Agile Service Development framework with modeling languages

3 Service Innovation and Service Design for the Future Internet

The community of service design has created a number of methods and practices independent from the software engineering community [1], they tend to focus more on interactive workshop activities and informal “sticky notes” for their documentation, rather than using formal modeling languages.

As an example the AT ONE method [1] takes an agile workshop-oriented approach to the identification of Actors, Touchpoints, Opportunities, Needs and Experiences, with a foundation in identifying innovative services with good user experiences. One integrated technique for this is CJM (Customer Journey Maps) [1], which illustrates the user experiences through the service journey between service provider and service user. This comes together with a number of creativity techniques to foster the innovation of services. The requirements engineering and software and service engineering communities could benefit well from integrating and adapting a number of the service design practices into their practices portfolio, and the service design community would benefit from the possibility to easier create prototype services for earlier and quicker user feedbacks. In particular for the creation of innovative services for the future internet it is of high importance to be able to do this as quickly as possible. Both a top down and a bottom up approach should be combined for this.

4 Business Level Requirements with Modeling Language Support

The following shows a number of modeling languages which are suitable for a direct language support for the various perspectives in the Agile Service Development framework [2].

Interaction can be supported with user interface mockups, as has been illustrated in [3], with mockups that also can be a basis for further generation of actual user interfaces through exchange languages such as BMML. Structure can be supported with VDML– Value Delivery Modeling Language [6] – an emerging standard from OMG focusing on value transfers in role collaboration and business activities. Function can be supported in an agile way by user stories in the spirit of agile requirements engineering [4] with a potential further refinement into use cases. Coordination can be supported by business level BPMN (Business Process Modeling Notation) [6]. Information can be supported by term definitions and relationships through ontologies, and representations in various graphical and textual forms as in ODM (Ontology Definition Metamodel) [6]. Extra functional aspects can be supported by BMM (Business Motivation Metamodel) [6], which focuses on identification of business vision and goals related to strategies with supporting processes and services.

5 Design and Implementation with Modeling Language Support

The following shows a number of modeling languages that are suitable for representation of the different design and implementation perspectives. Interaction can be supported through the emerging IFML (Interaction Flow Modeling Language) [6], which provides a platform independent approach for the specification of user interface dialogues. Structure with roles can be supported with VDML– Value Delivery Modeling Language [6] and the role models being related also to roles in SoaML for provisioning of services and further to the role of sensors in ThingML [7]. Function with services can be supported by service descriptions in SoaML (Service oriented architecture Modeling Language) from OMG, and also be further supported by USDL (Universal Service Description Language) which is being studied in W3C [8]. This can be supported further for realisation in Cloud environments using CloudML [7]. Coordination with processes can be supported by system level BPMN (Business Process Modeling Notation) [6], and further mappings to BPEL and TOSCA [9] for implementation level support for services and cloud environments. Information can be supported by further modeling through UML enhanced class diagrams, and have further mappings to representations in databases and semantic technologies. Extra functional aspects can be supported by constraints specification in various constraint languages, such as OCL (Object Constraint Language) [6].

6 Conclusion

Agile and model driven development for the Web and the Future Internet is a promising combination, in order to rapidly produce innovative services. Emerging modeling languages from OMG and others, with appropriate integration of language subsets, can be used as a good foundation for executable support for the various perspectives. There is a potential good synergy between the service design community and the software engineering community to foster the support for better service and software innovation. Thanks to partners in CSI, Center for Service Innovation, and the European projects NEFFICS, REMICS and ENVIROFI, for stimulating discussions and early experiments related to this approach.

References

1. Stickdorn, M., Schneider: This is Service Design Thinking. Wiley (2012), 1118156307
2. Lankhorst, M.: Agile Service Development. Springer (2012) ISBN 3642281877
3. Rivero, et al.: From mockups to user interface models. In: MDWE 2010 (2010)
4. Leffingwell, D.: Agile Software Requirements. Addison-Wesley (2010), 0321635841
5. Balsamiq – BMML, <http://www.balsamiq.com/>
6. OMG, OMG modeling standards, <http://www.omg.org>
7. SINTEF, ThingML and CloudML, <http://www.thingml.org>, <http://www.cloudml.org>
8. W3C, USDL, <http://www.w3.org/2005/Incubator/usdl/>
9. OASIS, TOSCA, <https://www.oasis-open.org/committees/tosca>