The Role of the Unified Process for Semantic Interoperability

Lopez DM[1], Blobel B[2]
[1]*Telematics Department, University of Cauca, Colombia*
[2]*eHealth Competence Center, Klinikum der Universität Regensburg, Deutschland*
dmlopez@det.uvigo.es

Introduction For ensuring quality and efficiency of patients' care, currently all industrial countries are looking for a change in care paradigm towards highly distributed, labor-sharing care settings. This process of moving from organization-centered care to process-controlled care (managed care, disease management) has to be continued towards the individualized personal care. The aforementioned paradigm shift also concerns the supporting information and communication technology (ICT). Several intents have been provided to deal with systems interoperability, from technical and data exchange interoperability, to semantic interoperability. However the service-interoperability where services are interchanged is still a big challenge.

Material and Methods For ensuring division of labor and the ability for integration in highly distributed architectures, definition and implementation of a unified process is inevitable. This concerns the entire lifecycle of a system from analysis over development up to implementation and deployment. Currently, the software industry has adopted a de facto standard to support the development process of software systems: The Rational Unified Process (RUP) [1]. The RUP covers the entire software development lifecycle by the definition of workflows. A workflow is configured by a set of roles (who), activities (how), and artifacts (what) necessary to transform the user requirements into a software system. The process identifies six core workflows: business modeling, requirements, analysis and design, implementation; test, and deployment. Besides, to define how the process rolls out over time, the RUP proposes four iterative phases: Inception, Elaboration, Construction, and Transition. Each phase is implemented as a set of workflows where, according the requirements of the system to be developed and the software development team involved; the number of iterations, roles, activities and artifacts are configured and extended. On the other hand, the HL7 community is on the final stage of defining a unified process for the development of HL7 standards specifications: The HL7 Development Framework (HDF). HDF replaces the former Message Development Framework (MDF) and documents the processes, tools, actors, rules, and artifacts relevant to development of all HL7 standard specifications, not only messaging. HDF defines the route of HL7 on the process of supporting the development of distributed health information systems, moving from a communication standard paradigm towards an architectural paradigm. However, its use as a comprehensive unified process for the specification of semantic interoperable health information systems is not clear enough. In the paper the characteristics of a unified process for semantic interoperability are defined, and then the RUP and HDF methodologies are compared based on the established requirements. The framework for the comparison is the Generic Component Model [2] which describes any system from 3 dimensions: the different domains involved, the level of granularity composition/decomposition of components, and the different views of systems architecture described in the RM-ODP process [3]. The comparison is restricted to two dimensions: the granularity of components and the RM-ODP.

Results The requirements defined for designing semantic interoperable health system are separated in two groups: 1) those related to the unified process: iterative process, component-based architecture, reusability, a common representation language and tooling support and 2) those required to reach semantic interoperability defined in [4]: composition rules and constraints, agreed terminology and ontologies, common reference model and service-oriented architecture. The results comparing RUP and HDF are summarized in Table 1.

Iterative process	RUP	Fully iterative. Define cycles, phases, and iterations and provide plug-ins to configure the process milestones according to the software project extension.
	HDF	Define a set of workflows with iterative activities, but not a configurable process.
component-based architecture	RUP	Starting from a use cases description, the software architecture is expressed as components. However, the architectural design is mainly performed during the analysis/design stages. An Enterprise Architecture is not fully supported.
	HDF	Provide reference information models that can be specialized according to the health information system requisites. This information architecture is the reference base for component-based architecture, however only attributes but not operations and interfaces have been specified.
Support Reusability	RUP	The iterative approach allows developers to identify components progressively and decide which ones to develop, which ones to reuse, and/or which ones to buy. However, only when components are designed using reference models and by the definition of enterprise architecture, components are reusable. The grade of flexibility of RUP on defining models is a problem.
	HDF	Information models such as RIM, DMIM, RMIM, CMET and HDM can be reused. Aggregating behavior to these models, reusable-components are feasible.
Common representation Language	RUP	Fully supported through the Unified Modeling Language UML.
	HDF	Most of the HL7 models are represented as UML v 1.4 diagrams. However, there exist some HL7 specific models. Referencing the model interchange format (MIF), inconsistencies between the HL7 meta-model and the UML meta-model have been resolved with a UML-conformant HDF meta-model included in the HDF specification.
Tooling support	RUP	The RUP is supported by an immense set of tools that automate the design process. Unfortunately, these are commercial tools for large enterprises.
	HDF	The HDF process can make use of the available tools for messaging design. e.g., RMIM Designer, Rose Tree, etc.
Composition rules and constraints	RUP	RUP is a constraint process defining a set of workflows and core artifacts. The resulting artifacts need to be defined according to reference models, specific architectural styles and design rules, however. These requirements can be freely defined by the software development team and normally relays on experiences on previous software projects.
	HDF	Constraints and composition rules are clearly defined for generating a message. Constraints are defined in attributes, data types, associations and cardinalities.
Terminology, ontologies and reference models	RUP	The vocabulary concepts are defined according to the specific requirements of the project. Domain models are normally only available on certain domains. Its utilization depends on the development team experience.
	HDF	The Activities in the HDF reference to a globally acknowledged RIM. HL7 also provides a rich vocabulary to define the concepts involved in several healthcare settings.
SOA	RUP	RUP allows the identification of services at the business modeling discipline. However, the alignment of those services with the software components generated on the analysis and design disciplines is not supported.
	HDF	Not service-oriented architecture supported yet, but the vocabulary and reference models envisage a SOA approach.

Discussion From the analysis of both methodological approaches, it was found that the RUP is an excellent methodological approach for specifying semantically interoperable systems because it is a fully iterative process, supports the composition/decomposition of components, uses a unified representation language, is flexible and configurable, and supports reusability. However, its main weakness is its grade of flexibility allowing the different stakeholders to define their own models according to the specific requirements of the system. It creates problems when designing semantically interoperable systems because the models and components obey to different rules. It is not possible to reuse the services if the system structure differs among systems. HDF lacks some best practices of a unified process developed by RUP such as the definition of iterations, configuration and project management. But it adds the constraining process required for RUP for the design of Health Information Systems. Domain-specific requirements and conditions can be consistently described by the RIM using object oriented and UML-based methods. The resulting models can be transformed in architectural components by adding operations to the classes and by defining interfaces to access to the services provided by the components. HDF should take a look to RUP as a complete process. It is possible to define HDF as a RUP plug-in that takes advantage from the knowledge generated in HL7 regarding health systems, reference models, vocabularies, data types, etc. These constraints add the value demanded by software engineering practitioners to realize semantically interoperable health systems.

Conclusion In the near future, it would be possible for HDF to evolve to a methodology for the design of HIS services, not only messages. The adoption of RUP would facilitate the process, however. The software engineering community would be happy to have a plug-in proposed by the HL7 community, providing not only the process framework for developing eHealth systems, but also service-oriented architectures and tools.

References

[1] Kruchten P. The Rational Unified Process. An Introduction. Third Edition. Addison Wesley. 2003.

[2] Blobel B Analysis, design and implementation of secure and interoperable distributed health information systems. Amsterdam: IOS Press; 2002.

[3] ITU-T Recommendation X.902 | ISO/IEC IS 10746-2, Information technology – Open Distributed Processing –Reference Model: Foundations, 1996.

[4] Blobel B, Engel K, Pharow P. Semantic Interoperability- HL7 Version 3 compared with advanced Architecture Standards. Methods Inf Med, 2006. (In print)