On Modeling “Web-Service-Based” Processes for Healthcare

Amir Afrasiabi Rad, Morad Benyoucef, Craig E. Kuziemsky

Telfer School of Management
University of Ottawa
55 Laurier Ave. E, Ottawa, Ontario, K1N 6N5, Canada
a.afrasiabi@uottawa.ca
{Benyoucef, Kuziemsky}@telfer.uottawa.ca

Abstract: In terms of business process-modeling, healthcare is a rather complex sector of activity. Indeed, modeling healthcare processes presents some special requirements dictated by the complex and dynamic nature of these processes as well as by the specificity and diversity of the actors involved in these processes. We discuss these requirements and propose a framework for evaluating process-modeling languages based on such requirements. The proposed evaluation framework is tested using BPEL and BPMN to model a complex healthcare process and the results of the evaluation are highlighted.

1 Introduction

Web services foster the integration of disparate systems despite being developed at different times by different people. Efforts on standardization of web service composition and modeling have resulted in the release of different standards, namely PDL (Process Definition Language), XPDL (XML Process Definition Language), BPSS (Business Process Schema Specification), BPML (Business Process-modeling Language), WSDL (Web Services Definition Language), WSCI (Web Services Choreography Interface), ebXML (Electronic Business using eXtensible Markup Language), BPMN(Business Process-modeling Notation), and BPEL4WS (Business Process Execution language for Web Services). The increase in the number of modeling standards and the diversity of business sectors raised the question of identifying suitable standards for specific business sectors. Many frameworks have been proposed for evaluating process-modeling standards; most of them focusing on one or two aspects of business process languages. Furthermore, evaluation results represented overlaps, limitations, inconsistencies, and ambiguities. These limitations are more observable on the modeling of complex systems.

Healthcare business processes are complex owing to integration of workflows, collaborations, and data transactions between different units, and also the existence of different medical data transaction standards (HL7 [He08]) and DICOM [Ne08]) contribute to the complexity of healthcare processes. They are profiled in the IHE (Integrating the Healthcare Enterprise) technical framework. Moreover, an increase in the number of medical disciplines and the dynamic nature of healthcare delivery call for change tolerant process models [AD04].

A process model is not just a graphical representation but rather it should also serve as a communication base both for communicating domain details between stakeholders and for communicating domain details to system designers [Ge04]. The ability to use a process model for communication becomes difficult if the modeling specialists are the only people who understand the models. Therefore the models should be representative enough to be also understandable by healthcare administrative or clinical stakeholders. A further issue is integration. If each healthcare unit uses a modeling language that best fits its requirements, the whole system's model will be a combination of several standards making the integration of models and their understandability difficult.

Scheduled workflow (SWF) is one of the most complex IHE workflow. It is representative of healthcare processes and provides a complete test base for the evaluation of our results because of its complexity, large amount of domain communication, and service like behavior. Most research initiatives on healthcare process-modeling within a web service environment have chosen BPEL as their modeling standard, and there has been limited research done on the suitability of business process-modeling languages in healthcare, particularly in complex work processes.

This paper reports on our ongoing research in process-modeling in healthcare by identifying the complexities of healthcare and proposing a methodological approach for applying modeling languages to healthcare. The paper is organized as follows. Section 2 presents background and related work. Section 3 investigates the applicability and suitability of process-modeling languages for the healthcare domain, Section 4 proposes an evaluation framework for modeling languages based on requirements specific to healthcare, and Section 5 applies the framework in a case study. The paper concludes with a discussion of the main findings and future research arising from these findings.

2 Background and Related Work

Modeling languages have fundamental differences with regard to Expressiveness, Flexibility, Adaptability, Dynamism, and complexity aspects [LS07]. A consequence is that different classes of modeling languages suit different business sectors. Explorations of common factors of modeling languages concluded that the understandability and complexity of models have a positive relationship [GL06]. However, to date, the focus has been on common features of all modeling languages while unique features have been neglected.

Existing research has developed a quality framework for evaluation of modeling languages [NK05]. However the quality framework does not suggest any definite metrics for evaluating languages and the comparison of modeling languages is only case-based and tightly related to the domain appropriateness quality area. Other research [NRC07] evaluated the understandability of models, providing clear metrics that make the measurement of understandability more accurate. Moreover, Luo and Tung [LT99] classified the characteristics of modeling languages and defined a set of steps for their selection. However, their proposed framework is general while healthcare has some specific requirements, such as permanent evolution and high value of communication, which are not considered in the framework in question.

Workflows and data flows are an important factor in an evaluation framework. Researches like [Wo02, Wo06, Va02, Va03] extracted Workflow, Data flow, and Control flow patterns, and provided detailed evaluations of modeling languages based on formerly mentioned patterns. Green et. al [Gr07] provided an ontological framework for evaluating process-modeling languages. They evaluated languages based on their ability to model ontological constructs. Results show that the number of constructs that each business sector implements is different, and some modeling standards are unable to represent some ontological constructs, or they are too complex in representation. In our research, we apply these frameworks to healthcare.

Anzbock and Dustdar [AD04] explored IHE workflows to study healthcare process-modeling in a web service based environment. Their work is limited to the evaluation of BPEL in test cases and the identification of considerations for modeling healthcare web services in BPEL. They defined transport, security and reliability as healthcare process model requirements and provided solutions for them in BPEL. Since their work is limited to the applicability of BPEL, and they have not analyzed the resulting models based on an evaluation framework, we believe further research should be done on the appropriateness of modeling languages besides their applicability. Furthermore, the authors argue that some healthcare process-modeling challenges, which come from the immaturity of the web service stack standards, remain unsolved.

3 Suitability of Process-modeling Languages for Healthcare

Healthcare processes are complex since modern health services involve care delivery by members of multiple groups with different knowledge levels and often residing in different physical locations. The complexity of a process can be measured by the number of units and the number of transactions between collaborating parts. For instance, the radiology sub system in the IHE framework contains 34 units and more than 65 transactions, and each of them contains several message types. Meanwhile, the fact that departments communicate and interact with each other adds to the complexity of healthcare processes. In this section we investigate the complexity of healthcare processes from a modeling perspective. We then elaborate on some features that should be included in modeling languages to support for modeling of healthcare systems.

Factors contributing to the complexity of processes include the fact that healthcare systems are constantly evolving and dynamic in nature. New expertise emerges and processes change in order to adopt new services. Healthcare actors' tasks are interconnected in a way that a change in one agent's task may affect others and departmental processes. Meanwhile, due to the dynamic nature of healthcare, agent's actions do not always follow normal routines since many emergency situations happen in healthcare systems. In emergency situations speed has the more important role, so agents do not exactly follow the routines, and sometimes their actions conflicts with others' tasks. Modeling languages should be able to represent the dynamic nature of healthcare processes through exception handling.

Influence of change in complex systems can go further if the inter-departmental interaction is tight. A change in an agent's task or a non-routine action can easily affect other agents, so changes can transmitted to other units in an incremental manner due to tight interconnections between healthcare units. Consequently, with regard to interconnection, a change in routines makes a huge change in the system. A simple example would be an emergency surgery which needs fast collaboration of many departments such as pathology laboratory, radiology, pharmacy, admission, and so on. Modeling languages should be able to represent this situation, so models flexibly represent routine and non-routine exceptional actions. Meanwhile, Models should always be loosely designed to accommodate independent actions of agents and their successive units.

Different agents in complex systems may simultaneously be member of more than a unit and memberships may change frequently due to unforeseen events. A small change in the processes of a unit in the healthcare system may create conflict in departmental business models. Nested healthcare processes and agents create inherently non-linear relationships and remove the boundaries between sub-processes. This fact complicates the modeling task and increase difficulty of change in models as internal rule sets are not very strict, coordination of units in complex systems cannot be represented as a rigid model. For the long lasting models, modeling languages should be able to represent healthcare's fuzzy departmental boundaries, and provide features that make the maintenance process easier. Modeling languages, also, should be able to model this high level of communication. Meanwhile, a modeling language should be able to adapt to wide range of changes in models without needing extensive remodeling. Modularity and support for abstraction in models are key concepts for change adoption. Completeness and extensibility of the language provide means of modifying the model to adapt it to changing processes.

The behavior of a complex system emerges from the interaction among the agents, departments, and even systems. Different sub systems may have dissimilar features thus divergent modeling requirements. In order to accommodate all requirements in models, sometimes, it is essential to use diverse modeling language for different departments. These models should be able to communications between departments and it is not possible if modeling languages do not have matching features and ability to be mapped to each other.

Security and privacy are crucial issues in healthcare systems, particularly when the communications are done over the Internet and the structure is service-oriented. Thus the models need the ability to represent security and privacy considerations.

To deal with the complexity of healthcare processes, process-modeling languages should allow the modeler to create optimized models with different numbers of specifications for different purposes. For models used by healthcare staff, for instance, more detailed models are confusing; and thus should only represent the high level graphical representation of processes. However, models used by IT department staff require detailed specifications. To comply with the need for different views, flexibility is required in the level of details and representation.

4 Evaluation Framework

Process-modeling languages are innovatively designed or developed from existing modeling approaches by unifying several methods or adding features to an existing one [POB00]. We developed a framework (Fig.1) for evaluating modeling languages for service based healthcare environments. We consider languages' unique features in addition to their common features since the unique features are key for specific uses. We also need to consider general modeling qualities, which are required by all systems, in addition to healthcare specific requirements. We identified important features that allow us to measure the quality of modeling languages. From the work performed in [LM04], we understand that models should be representative, understandable, easy to use, support abstraction, and be optimized in the level of details. In the following paragraphs we explain our framework and the metrics used for measuring each evaluation criterion.

Security: Security is important when dealing with service oriented architectures and communications over the internet. For all the various uses of the models (training, development, documentation, etc.) security and privacy issues must be considered. For healthcare processes, where security has a special importance, modeling languages must represent inter-departmental, general access privileges and secure interactions.

Pattern representation: The understandability of a modeling language is a fundamental requirement for increasing its usability, and is tightly related to the process modeler's capabilities in addition to the modeling language's features. Factors such as the modeler's level of expertise, creativity, and familiarity with the business, and tool, are inevitable. Pattern representation capability of a language for data flow, communication, and control flow patterns is an important criterion for the understandability of languages. Studies in [Va02, Va03] showed that some patterns cannot be modeled by any modeling language, and some patterns can only be modeled using a limited number of languages. Thus the understandability of modeling languages depends on the ability and methods of representing patterns. If there is a construct in the language that directly represents a pattern, the understandability of the language will increase. However, combining several constructs in order to represent a pattern decreases the understandability and makes the language ambiguous.

Fig 1. Evaluation Framework

Ontological constructs: Language concepts are evaluated using ontological constructs. The BWW (Bunge-Wand-Weber) ontological framework is proposed in [WW93] for evaluating modeling grammars. The authors argue that modeling notations which are not able to represent all of the ontological constructs are incomplete. Language incompleteness reduces understandability because incomplete language features and structures force the modeler to ignore some features of models as there is no construct supporting those specific needs. Nevertheless, while there may be other ways to represent the needed constructs in order to compensate the languages' incompleteness, these ways are diverse and cause ambiguity.

Extendibility: Extensions help to support different technologies and different business sectors, and also to overcome modeling language deficiencies. However, since model users are familiar with basic model notations, extensions in the language can create confusion for those who are not familiar with the extended features. Consequently, extendibility is a compromise between increased capability and general understandability.

Notations: Modeling languages are categorized, based on notations, into graph based, rule based languages, and their combinations [LS07]. Language notations can be measured by the way they follow the standards e.g. standard element size and colors for graphical notations. Textual notations can be measured based on the same concept. The keywords representing concepts and descriptive words should follow their existing standards. For languages that provide execution features, separation of computation and representation of processes differentiate two diverse aspects of modeling, so each actor can access a part of the model which addresses his needs. Meanwhile, uniformity (use of the same set of notation with unique and unified meanings) and formality (choosing commonly accepted graphical notations for language concepts) of notation improve the language's ease of use.

Modularity: The modularity of modeling languages can be measured by considering the support for abstract processes and sub-processes. Abstraction prevents from revealing the underlying layers of a process, hence improving the understandability of process models for non-developer actors. Also, the understandability of developers and designers from models will improve when the unimportant data is hidden. Reusability is an important factor and it is tightly related to modularity. The ability to use previously modeled processes increases the speed and accuracy of modeling. It also reduces the time required for understanding models. Maintenance and modification of models with reused parts is easier. Reusability has a positive connection with understandability in a way that more reused sections in models help decrease the required effort for learning the process. However, it is not necessarily true that languages that support sub-processes also support reusability.

Level of detail: The level of detail provided by modeling languages should be optimized for different modeling proposes. This metric has a close positive relationship with other aspects of a modeling language such as notation, abstraction and ability of the language to execute processes. Detailed documentation of processes improves the level of understanding for the designer and developer actors. However, the end user, who uses the models for training, does not need detailed information, and more detailed documentation may impair his understanding. Consequently, the languages should be flexible in the level of details, and provide facilities to support both detailed documentation and high level representation of processes. This criterion can be measured by support for abstract processes. Also, languages should not force the modelers to create detailed models. Languages that support execution should be able to separate execution and representation in order to hide unnecessary details.

Exception handling: Exception handling is an important area of evaluation. Exceptions have been an indispensable part of business process-modeling for more than two decades [Cu03]. Exception handling features increase modeling difficulty but also increase the adaptability of models to exceptional circumstances. Improved adaptability helps model users to predict all model behavior in the time that exceptions occur, so the models are more understandable. For healthcare processes with a high possibility of exception occurrence, this criterion has high importance. Investigations in modeling languages show that not all modeling languages contain cancelation functionalities. In the case of those languages, the processes end naturally; the modeler does not have the ability to terminate a process at a special time. Being able to define the termination conditions and terminate processes at a certain moment improves the understanding of process timelines.

The evaluation framework we propose here provides a starting point to evaluate different modeling languages in the healthcare domain. .The strength of the framework is its extensiveness with regard to covering unique features of languages, which makes it an unbiased test-bed for comparing different language families. Implications for general modeling are also considered.

5 Case Study

The *Radiology Admission Process* scheduled workflow (SWF) is one of the most complex IHE processes in healthcare. Consisting of twenty transactions and eight different IHE Actors, SWF is representative of healthcare processes and provides a complete test base for the evaluation of our framework because of its complication, large amount of domain communication, and service like behavior. As security, privacy, change, and exceptions are indispensable parts of healthcare processes, this SWF represents all features of healthcare processes and all requirements of healthcare process models.

In this paper we use the *Radiology Admission Process* SWF to evaluate two modeling languages (BPMN and BPEL) using our proposed framework. Note that our ongoing research is not limited to these two languages, and that other modeling languages are being studied. BPMN is a graphical workflow modeling language that does not focus on any technology; it is able to model processes focusing on service oriented architectures. BPEL is a process-modeling language within web service environments. The collection of these two different languages (different from a notation and architectural perspectives) makes a unique test bed for our framework at this stage of our research. We also evaluate the applicability of the two languages to healthcare processes.

Security: None of the languages has constructs to represent security in the models, and security representations are provided in the extensions to the languages. The two languages lack the fundamental security feature without use of extensions. However, it should be noted that some extensions make a significant alteration in the language. BPMN's security enabled version is developed in [RFP07]. In this version the security extension provided additional graphical notations which compromised the standard notation of the language.

Pattern support: Based on our evaluation criteria, one of the measurements is the level of support for the workflow and data flow patterns involved in healthcare processes. Patterns are documented in [Wo08]. Evaluating the patterns using our evaluation framework and the findings in [Wo02, Wo06] are shown in Table 1. The capability of modeling languages in representing patterns and the understandability of their method in representing patterns are separated by a slash (/) in the table. For instance, BPEL is able to represent "Structured Loop" directly using while, but there is no direct representation for the *repeat until* structure. In another example, BPMN supports "Deferred Choice", but there are two different options; either by combining event-based exclusive gateway and either intermediate events or receive tasks.

Ontological Completeness: Findings in [Gr07] on ontological mapping of modeling languages and Bunge-Wand-Weber (BWW) ontological base model show that modeling languages are not able to represent all the ontological constructs documented in [WW93, We97]. We studied healthcare processes for their constructs and mapped them to BWW models. The findings show that not all the ontological constructs are used in healthcare processes. Table 2 illustrates the constructs used in healthcare and the findings in [Gr07] are used in order to show the support of languages for the ontological constructs. For BPMN, we used the same method that is used in [Gr07] to map it with BWW.

Extendibility: Both BPEL and BPMN are extendable and can accept new features to the language. For healthcare modeling both languages need a security extension and BPEL also needs an extension to support sub-processes.

Notations: Both BPEL and BPMN follow an accepted standard for their notations. From the graphical notations viewpoint both languages' representational components are unique. However, BPEL's notation is not understandable for non-modeler actors since its textual and graphical notations are entwined, so a model user must understand the BPEL's background coding as well as the graphical notation. Moreover, this inseparability makes changes to the graphical notation for the non-expert user rather cumbersome.

Modularity: The overload of transmitting information, the high amount of communication and the need for reusability make sub-processes an indispensable feature in healthcare process-modeling. BPMN supports sup-processes in the modeling procedure without extensions to the modeling languages. By contrast, BPEL does not provide any support for sub-processes as standard. IHE transactions take place between IHE actors (healthcare unit), and each transaction contains a set of HL7 and/or DICOM messages. Each actor is assigned an internal process determining the type of messages that will be communicated in a transaction, and creates the information in the way that is necessary for the message type. Consequently, there is a process running inside each IHE actor. Using BPEL, the goal is to model processes in a web services environment. For the whole IHE model, the central process will regulate communication between services, if IHE actors are considered as service providers. In this system, however, actors need to perform internal processes. There are two ways to represent this system in BPEL; the first way is to provide an abstract central process with conceptual transactions between units in addition to actor's internal processes modeled in different BPEL processes with a reference to a central process (see Fig 2). This makes the process simple and understandable without sacrificing the model's execution feature, each sub-process can be executed separately. The second way is to use extensions. While extensions to BPEL increase its understandability to users experienced with extensions, understandability decreases for those who are unfamiliar with new elements. Moreover, the use of extensions decreases the portability of models. On the other hand, BPMN is able to represent sub-processes and IHE's internal and external one-to-many relationships (see Fig 3).

Level of Detail: Different healthcare model users require different views of the processes. Different levels of detail specify the intended use of models. To a large extent, BPEL is inflexible in the level of detail that can be provided to the end user. BPEL process models are associated with coding which contains a high volume of details. BPEL's limitation is the inseparability of graphical notation from background execution code. Healthcare managerial staffs do not need to see detailed code while it is necessary for IT actors. The inflexibility of BPEL at the detail level makes the modeling, on some levels, cumbersome and decreases the understandability of models to specific non-IT actors by confusing them with additional components on models. On the other hand, BPMN is more flexible on the detail level. Although it does not allow the model designer to add coding to the model, additional detail for special model users can be added in the form of comments. BPMN does not force the user to a pre-specified level of comments as opposed to BPEL in which the model designer is forced to create fully detailed models.

Exception Handling: Both BPMN and BPEL provide a complete set of exception handling and compensation features. While these features are more important in BPEL, healthcare processes demand adequate support for exception handling features to increase the understandability of processes. Fig 2 and Fig 3 illustrate the complexity of exceptions and the way that BPMN and BPEL represent them. However, from the representational point of view BPEL is more limited in dealing with exceptions since there is no cancellation element. The modeler cannot decide to end a process instantly; the process continues until it gets to the ending point. This limitation results in larger models with more connections however the flow of model finishing only at the end point makes it easier to follow the process workflow.

Pattern	Standard (capability / understandability)	
	BPEL	BPMN
Sequence	+/ +	+/+
Parallel Split	+/(+/-)	+/+
Synchronization	+/(+/-)	+/+
Exclusive Choice	+/+	+/+
Simple Merge	+/+	+/+
Structured Discriminator	-/-	(+/-)/-
Arbitrary Cycles	-/-	+/+
Implicit Termination	+/+	+/+
Multiple Instances without Synchronization	+/(+/-)	+/+
Multiple Instances with a Priori Design-Time Knowledge	-/-	+/+
Multiple Instances with a Priori Run-Time Knowledge	-/-	+/+
Deferred Choice	+/+	+/(+/-)
Interleaved Parallel Routing	(+/-)/-	-/-
Cancel Activity	+/+	+/+
Cancel Case	+/+	+/+
Structured Loop	+/(+/-)	+/+
Recursion	-/-	-/-
Transient Trigger	-/-	-/-
Persistent Trigger	+/+	+/+
Cancel Region	(+/-)/-	(+/-)/-
Blocking Discriminator	-/-	(+/-)/-
Cancelling Discriminator	-/-	+/(+/-)
Structured Partial Join	-/-	(+/-)/-
Blocking Partial Join	-/-	(+/-)/-
Cancelling Partial Join	-/-	(+/-)/-
Critical Section	+/+	-/-
Interleaved Routing	+/+	(+/-)/-
Thread Merge	(+/-)/-	+/+
Thread Split	(+/-)/-	+/+
Explicit Termination	-/-	+/+

Table 1. Pattern support of modeling languages for healthcare

Constructs	Standard (capability / understandability)	
	BPEL	BPMN
Property	+	+
Class	+	-
State	+	+
Conceivable state space	-	-
State Law	-	-
Lawful State Space	-	-
Event	+	+
Conceivable Event Space	-	+
Transformation	+	+
Lawful Transformation	+	+
Lawful Event Space	-	+
Coupling	+	+
System	+	+
System Composition	+	+
System Environment	-	-
System Structure	+	+
Subsystem	-	+
System Decomposition	-	+
Level Structure	-	+
External Event	+	+
Stable State	-	-
Unstable State	-	-
Internal Event	+	+

Table 2. Ontological representation of modeling languages for healthcare

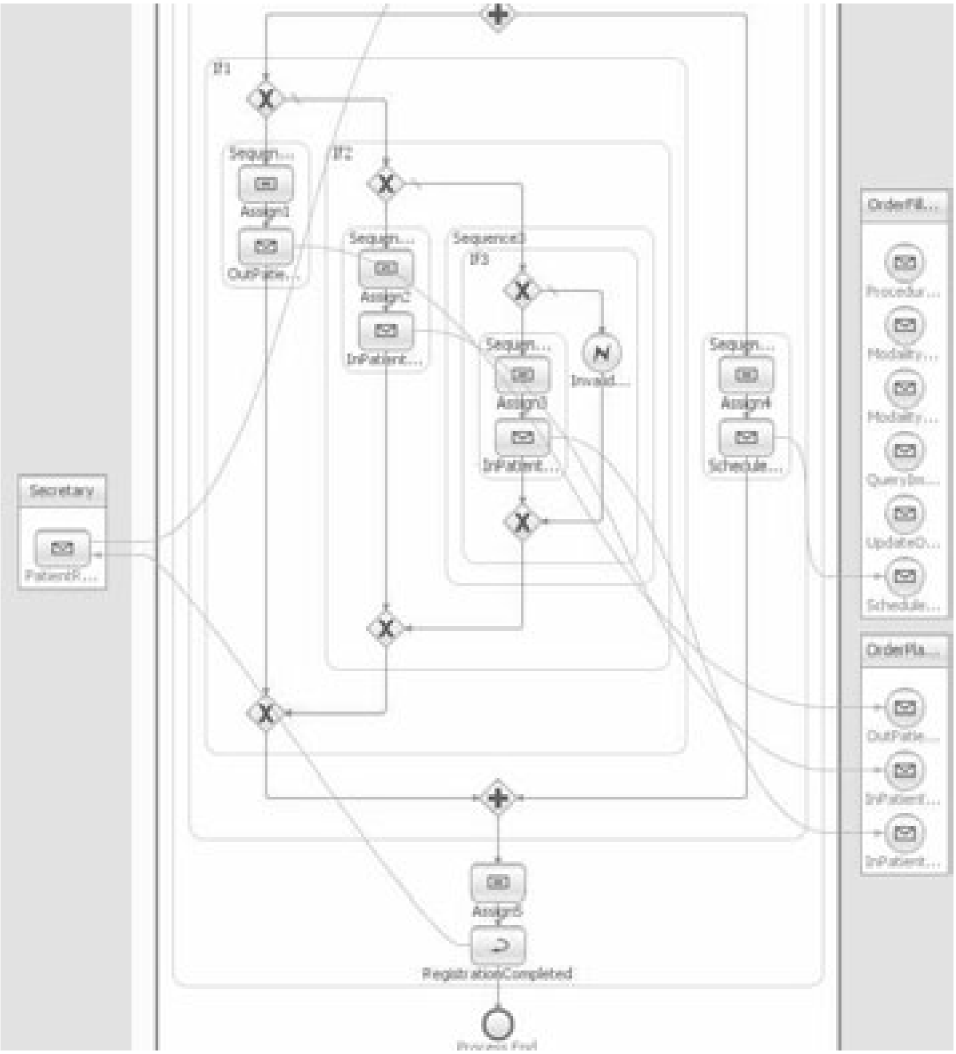

Fig 2. ADT sub-Process and exception handling - BPEL

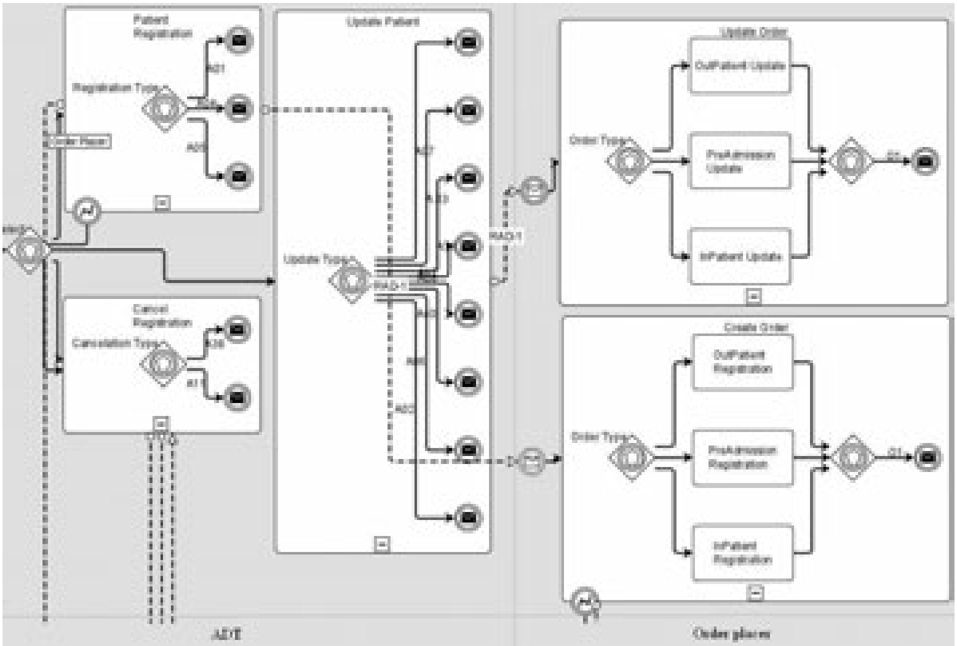

Fig 3. Sub-processes and exception handling - BPMN

6 Conclusion

Many research initiatives have been focused on the ambiguities and difficulties on normalization of models and modeling languages in general business sector focusing on the goal of improve our ability to design information and communication technologies. As specifications of different business sectors and variety of processes imposes the utilization of a subset of modeling features and complex and simple business sectors are fundamentally different, modeling languages should be seperately evaluated on each business sector. This paper presents the preliminary results of our research on modeling healthcare processes within a service oriented environment, taking into account the specificities of the healthcare sector as well as its actors in order to enhance business process-modeling and improve to design information and communication technologies to support complex domains such as healthcare.

We introduced a framework for evaluating business process-modeling languages in healthcare, defined a set of requirements for healthcare process models, and performed a case study on a complex IHE transaction using BPEL and BPMN. So far, results show that neither of the studied languages satisfies healthcare modeling requirements, and that extensions to existing languages are necessary in order to generate models that can suit healthcare modeling requirements.

Most complex systems share a similar set of requirements and specifications. As a result, although healthcare was your domain of focus the findings can support process-modeling in other domains that have complex requirements.

The next step in our research is to study different suitable modeling methodologies for healthcare, merge languages, and possible extensions to selected modeling languages. In addition to BPMN and BPEL, other modeling languages are being studied for their suitability for the healthcare sector.

Bibliography

[AD04] Anzbock, R.; Dustdar, S.: Modeling medical e-services. Lecture Notes in Computer Science - Business Process-modeling Springer Berlin, Heidelberg, 2004; pp. 49-65.

[Cu03] Curbera, F. et. al.: Exception Handling in the BPEL4WS Language. BPM 2003; pp. 276.

[Ge04] Gemino A, Wand Y. A framework for empirical evaluation of conceptual modeling techniques. Requirements Eng, 2004, 9:248-260

[Gr07] Green, P. et. al.: Candidate interoperability standards: An ontological overlap analysis. Data & knowledge engineering, vol. 62, no. 2, 2007; pp. 274-291.

[GL06] Gruhn, V.; Laue, R.: Complexity Metrics for Business Process Models. 9th international conference on business information systems, 2006; pp. 1.

[Hl08] Health Level Seven, Inc.: Health Level Seven, Inc. Available: http://www.hl7.org/ 2008.

[LM04] Laguna, M.; Marklund, J.: Business Process-modeling, Simulation, and Design, Pearson Printice Hall, USA, 2004.

[LS07] Lu, R.; Sadiq, S.: A Survey of Comparative Business Process-modeling. BIS 2007, pp. 82.

[LT99] Luo, W.; Tung, Y.A.: A framework for selecting business process-modeling methods. Industrial Management & Data Systems, vol. 99, no. 7, 1999; pp. 312-319.

[MRG07] Mendling, J.; Reigers, H.A.; Cardoso, J.: What Makes Process Models Understandable?. Business Process ManagementSpringer, Springer Berlin, 2007; pp. 48.

[Ne08] NEMA, O.: DICOM. Available: http://medical.nema.org/, 2008.

[NK05] Nysetvold, A.G.; Krogstie, J.: Assessing Business Processing Modeling Languages Using a Generic Quality Framework. Workshop on Exploring Modeling Methods in Systems Analysis, 2005; pp. 545.

[POB00] Paigea, R.F.; Ostroffa, J.S.; Brookeb, P.J.: Principles for modeling language design. Information and Software Technology, vol. 42, no. 10, 2000; pp. 665-675.

[RFP07] Rodriguez, A.; Fernandez-Medina, E.; Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE Transactions on Information and Systems, vol. E90-D, no. 4, 2007; pp. 745-752.

[RT06] Rossi, D.; Turrini, E.: What your next workflow language should look like. 2nd InternationalWorkshop on Coordination and Organization. 2006.

[Va02] Van Der Aalst, W.M.P.; Dumas, M.; Ter Hofstede, A.H.M.; Wohed, P.: Pattern based analysis of BPML (and WSCI), Queensland University of Technology, QUT, 2002.

[Va03] Van Der Aalst, W.M.P., Ter Hofstede, A.H.M.; Kiepuszewski, B.; Barros, A.: Workflow Patterns. Distributed and Parallel Databases, vol. 14, no. 1, 2003; pp. 5-51.

[WW93] Wand, Y.; Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information Systems Journal, vol. 3, no. 4, 2003; pp. 217-237.

[We97] Weber, R.: Ontological Foundations of Information Systems, Coopers & Lybrand and the Accounting Association of Australia and New Zealand, Melbourne, Australia, 1997.

[Wo02] Wohed, P.; Van Der Aalst, W.M.P.; Dumas, M.; Ter Hofstede, A.H.M.: Pattern Based Analysis of BPEL4WS, Queensland University of Technology, QUT, 2002.

[Wo06] Wohed, P.; Van Der Aalst, W.M.P.; Dumas, M.; Ter Hofstede, A.H.M.; Russell, N.: On the sustainability of BPMN for business process-modeling" in Lecture Notes in Computer Science - Business Process Management Springer Berlin, Heidelberg, 2006; pp. 161-176.

[Wo08] Workflow Patterns Initiative: Workflow Patterns [Homepage of Workflow Patterns Initiative], [Online]. Available: http://www.workflowpatterns.com/ 2008;