Towards an Ontology-driven Framework for Workflow Analysis

Hlomani Hlomani and Deborah Ann Stacey
University of Guelph, School of Computer Science, 50 Stone Road East, Guelph, Ontario, Canada

Keywords: Ontologies, Workflow, Process Models, Workflow Modelling, Workflow Analysis, Ontology-based Reasoning, Knowledge-based Systems, Ontology-driven Compositional Systems.

Abstract: Workflow management and the the whole field of business process management has seen a lot of research interest. This interest has evolved from the initial quest to automate manufacturing processes to the formalization of process models. The reason for this interest can arguably be attributed to the fact that process models form the core of workflow management systems. A plethora of modelling languages and notations have been created through the years, albeit with dominance of proprietary languages that has been argued to be lacking in terms of having formal semantics. The informal languages have seen more adoption at the expense of those that are termed "academic languages" even though academic languages are believed to be more formal. This paper considers the aspects of model transformation with the intension to bridge the gap between modelling and analysis. The paper proposes a semantic approach (using ontologies) to both the mapping and transformation of business process models written in one language (source) to another (target).

1 INTRODUCTION

The workflow concept has existed for many years. While its origins are not from computer science, it is now as much a computing concept as computational analysis is. In its initial conception, the concern was the automation of manufacturing processes. We now see a redefinition of the workflow concept mostly based on the premise that, scientists mostly in collaboration with computing expects have and continue to develop computational models (i.e. simulations of real world etc.) which can be strung together to form well balanced systems. While workflow modelling is at the core of workflow management and analysis, there exist a plethora of modelling techniques.To-date and for a long time, the modelling and specification of these work- flows have always dependent upon proprietary languages and systems which literature has argued the existence of some flaws viz.: lack of formal model basis, lack of support for patterns (and are hence deficient in terms of their expressiveness)(van der Aalst and Hofstede, 2002), rigidity of the languages both at design and execution (Almeida et al., 2004) etc.

Two camps of process modelling languages exists: those that originated from industry and academic languages. Industry languages are believed to be widely adopted and used in the industry while academic languages are said to be less appropriate for being used in concrete industrial application domains. The industry languages have the limitation of not having formal bases (formal semantics), something which academic languages have. It is then apparent that choice between which language to use has been a matter of trade-off. For example, a choice between the easy and intuitive Unified Modelling Language's Activity Diagram (UML AD) that has no formal basis versus Petri Nets which have a theoretical bases but are much more involving.

In light of the absence of that "one" leading notation or language, model transformation would play an important role in the world of workflow modelling. Model transformation would be concerned with the translation of a workflow model described in a source language to a semantically equivalent model in a target model. It is against this background that we propose a framework for workflow modelling and analysis that utilizes an ontology-based knowledge base, a semantic transformation function, and universal analysis engine.

2 BACKGROUND

2.1 Ontologies

A search for a precise and concise definition of *"on-*

Hlomani H. and Stacey D..
Towards an Ontology-driven Framework for Workflow Analysis.
DOI: 10.5220/0004165404050410
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 405-410
ISBN: 978-989-8565-30-3

tology" especially as it relates to computing can be a challenging task. This is perhaps because the concept never really had its origin in computer science. The concept of ontology has its origins in the field of philosophy where it refers to a branch of metaphysics that endeavours to offer a systematic account of existence (Gruber, 1993). In the computer science domain, however, as defined by prominent players in the domain such as the Object Management Group (OMG), World Wide Web Consortium (W3C) and pioneers in the domain (such as Tom Gruber), an ontology is "a formal conceptualization of a domain of interest" (Group, 2009; Consortium, 2009; Gruber, 1993). A conceptualization is in fact an abstraction of that which we wish to represent. In this abstraction, a declarative formalism is used to specify the objects in the domain, their desirable properties and the relationships among them. This forms a shared vocabulary with which knowledge-based systems can be built.

2.2 Ontology-driven Compositional Systems

Motivations for compositional systems abound, even so for those that are ontology-driven. These compositional systems are of varying types (ranging from web services-based, agent-based, resource-based, grid-based etc.). Some of these came about because of the advent of web services that opened up a world of opportunities for developers, service providers, service consumers, and organizations. Web services are networked capabilities that openly exposes interfaces for other machines to discover, rank, bind to and invoke (Blake and Nowlan, 2008). Their characteristics such as interoperability, network and platform independence has sparked a lot of research interest with researchers investigating the many ways in which a varying number of web services can be put together to form a well-balanced system (Charfi and Mezini, 2007). Researchers such as (Cardoso and Sheth, 2005; Kim et al., 2010) have focussed on the web service composition approach that exploit ontologies to provide descriptions of web services thereby aiding in the discovery, selection, composition and execution of web services. These type of systems are made possible by the definition of standards such as the Web Ontology Language for Service (OWL-S) from W3C (wide Web Consortium, 2004).

Another distributed platform that research has focused on (in the context of compositional systems) is that of agent based platforms. A notable work in this regard is that by a group of researchers at the Stanford university who implemented a prototype system called BIOSTORM (Nyulas et al., 2008). In this research, the primary goal was the integration of various heterogeneous data sources, the deployment of a varied number of detection algorithms through the collaboration of multiple agents. Like the BIOSTORM project, Wang et al. (Wang et al., 2005) proposed an agent-based framework that utilizes ontologies as their knowledge-base. While differing in their usage and interpretation of the ontologies ((Nyulas et al., 2008) using ontologies to describe the data sources as well as detection methods and (Wang et al., 2005) focusing on description of the workflow so as to make its execution dynamic) the overall platform remain relatively similar (i.e. multi-agent platform).

Based on the premise that scientist mostly in collaboration with computing experts have created computational models which may or may not be cast as web services, or agents (Hlomani and Stacey, 2009; Gillespie et al., 2011) focused on a "plug and play" bottom-up system composition approach. This approach leverages the user's expert knowledge (captured in ontologies) and assist the user in the composition of a system through the connection of composition units described in the ontologies that underlie the system.

In all these research, the ultimate result was some sort of workflow. The limitation of most of these implementations was that less focus was given to the analysis of the workflow. If any analysis was performed, it was limited to syntactic checks which one may be inclined to say are trivial to detect.

2.3 Workflow Modelling

Business processes are the core of workflow management hence, the paramount need for proper modelling, analysis and verification (Salimifard and Wright, 2001). Process models are an abstraction of the business requirements in terms of executable tasks that put together in the end produce an intended output. It is worth mentioning that there exists many different ways to model business processes (from the simple but intuitive graph theory (mostly directed graphs), to the more complex but somehow mechanical Petri Nets) (Atsa et al., 2011). The question of which technique to follow seems a difficult one to answer since each technique comes with its justification and possibly disclaimers as to what the limitations are. Having said that, (Cardoso et al., 2006) argues that the decision to use a certain modelling technique could be determined by the complexity of the problem. Other reasons for using a certain technique could be motivated by the expressiveness, and formal basis of the model (e.g. W-Nets and YAWL (van der

Aalst, 2003)), coverage of both local and global dependencies (e.g. deterministic graphs and path constraints (Fan and Weinstein, 1999)) or the simple and intuitive nature of control flow graphs and the Unified Modelling Language (UML) etc.

While these process modelling techniques may differ in such things as their expressiveness, formal basis etc., a relative relation can be drawn between them (including the fact that most are graph-based or follow some sort of a graph formalism and that some are derivatives or extensions of the other).

2.4 Workflow Languages

In the studies leading to the proposal of YAWL (van der Aalst and Hofstede, 2002) and later on (van der Aalst, 2003) offer a comprehensive analysis of workflow languages. These motivating factors for YAWL are also echoed by (Weske et al., 2006). In both cases they make two significant observations viz.: lack of formal bases (or at least no practical evidence to that fact), and lack of support for workflow patterns (or some patterns). In these analyses, what is clear is that, despite efforts to standardize by such organizations as the workflow management coalition, consensus with regards to workflow languages is far from agreed. This is perhaps as a result of the many ways in which business processes can be described. Their work is biased towards Petri Nets particularly High-Level Petri Nets (i.e. Petri Nets extended with colour, time, and hierarchy) which further forms the foundation for YAWL. Workflow languages were evaluated to determine their support for workflow patterns and were found to be wanting in terms of their expressive power.

Business Process Execution Language For Web Services (BPEL4WS) offers another interesting case for workflow languages. It is popular in the web service composition domain (if not the de facto standard) (Weske et al., 2006). BPEL4WS carries the blocked-structure and graph-based modelling from its predecessors (XLang and Web Services Flow Language (WSFL), respectively). While it is said to be lacking a graphical notation, some constructs of the BPMN discussed earlier can be directly mapped to BPEL and hence can be executed in a BPEL execution engine.

2.5 Workflow Ontology

Some work has been done towards the ontological representation of the workflow concept. One such work is the IntelLEO Workflow Ontology (Jovanovic et al., 2011). While the motivations and structure of this ontology has a bias towards the description of the learning flows, it has a sense of "traditional" workflows. This is because the ordering (either sequential or parallel) of activities to achieve some goal can be achieved. The ontology, however, lacks the expressive ability to capture even some of the most basic concepts of process models (e.g. the notions of routing beyond just "sequence" and "parallel").

Jenz (Jenz, 2003) explores an interesting aspect of software design: the generation of software artifacts from an ontological knowledge base (KB). Much like the approaches discussed in section 2.2, this approach is dependent on the definition of a knowledge base ($ontology + instances = KB$). Contrary to the previously discussed ontologies, the aim of the business process ontology (Jenz, 2003) is to be generic in its description of business processes. This is achieved through a static representation of a business process by focusing on *activities* or *tasks* as the "building blocks". The representation of the business process in an ontology then achieve two main goals viz.: provide a vendor-neutral and platform-independent description of the business process, and provide both human-understandable and machine-readable description of the business process. The ontology is general to some extent (it covers a larger number of the concepts needed to describe business processes). However, while synonyms and homonyms can be included in the ontology to cater for terminology gaps, there are cases where a one-to-one mapping cannot be achieved particularly where business processes are concerned. To give an example, BPMN, UML activity diagrams and Petri Nets are all business process modelling techniques. While we could say thet a *transition* (Petri Net concept) is synonymous to say an *action state* (UML concept) or a *vertex* (Graph concept), it has been argued that this may work for many of the concepts but not for some constructs (Lohmann et al., 2008). The separation of concerns principle would therefore, be ideal. This would follow a pattern where top level concepts are detached from a specific modelling technique and further specialized going downward. This is opposed to the idea followed in this ontology where there exists a bias towards the use of UML concepts.

3 THE PROPOSED FRAMEWORK

In section 2.3 we mentioned that business process/ workflow modelling is a core element of the business process management and workflow analysis task. We also deduced from literature that there exist umpteen modelling techniques. Suffice it to say that each modelling technique has its role to play in the business process modelling process (depending on the modelling context, the problem in question in terms of size and expressiveness requirement etc.). Having said this, it also suffices to say that choice of which model to use depends entirely on the user, rather, on the modeller.

Our proposal endeavours to ease the burden of modelling technique choice by providing a means to traverse the business process modelling space. In this framework we focus on the transformation from one model to the other with minimal loss of information. This framework is depicted in figure 1. We identify components of the framework to include: Ontology-Driven Compositional System/ Workflow Management System/Business Process Modelling System, Knowledge base, Semantic Transformation, and Universal Analysis Engine (this is not discussed because it will not be a part of the initial instance of the framework). These are discussed in subsections that follow.

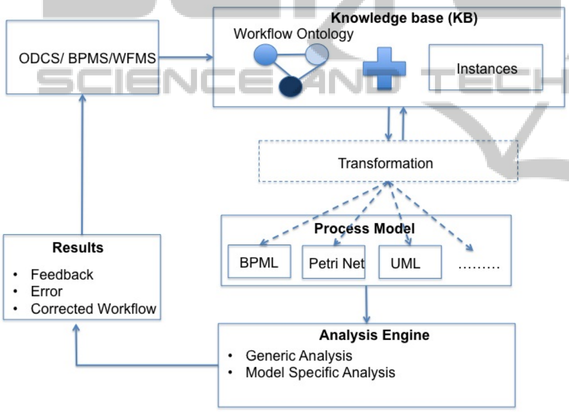

Figure 1: A universal framework for workflow management and analysis. The framework relies on an ontology-based knowledge base.

3.1 The Knowledge Base

The goal is to have a universal (at least in terms of coverage) knowledge base for workflow representation/ modelling models. Following the notion that process models are mostly similar with subtle difference (a one-to-one mapping for some constructs is not possible), the knowledge base would then have a generic top-level description (describing the common constructs) and model specific constructs further to the bottom (describing the constructs that are specific to modelling languages). The most general constituents of a process model are *activity*, and *relation*. In terms of relation, the modelling of *split* and *join* elements is important because these facilitate the creation of workflow patterns (e.g. parallel routing etc.). Figure 2 depicts the structure of such an ontology in which clouds represents imported ontologies (these would in most cases be addressing specialization constructs), rectangles represent the classes, arrows represents relationships, while the lines represents subsumption (sub-classes). The most basic definition of a workflow is a **graph** (ordered pair) $G = (A, E)$ consisting of a set A of **Activities** together with a set E of **Relations**, the ontology is thus, representative of this structure (at the top level). Having defined these general notions (workflow/process, activity, and control flow), specific modelling techniques could then be mapped to these top-level elements as exemplified in figure 2 by sub-classing the workflow class to define a *Petri Net* sub-class and drawing a relation between the Petri Nets constructs and the workflow constructs (e.g. between the *Activity* class and *Transition* class since these classes are semantically equivalent).

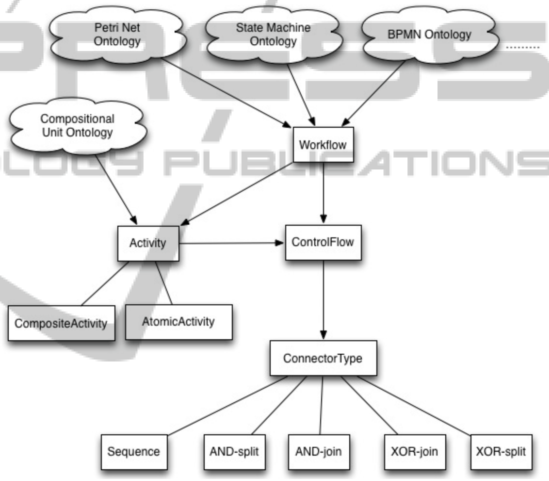

Figure 2: Graphical view of the workflow ontology. Like the framework, the ontology has a generic yet extensible structure to allow for further specialization through mappings to other ontologies and concepts whenever needed.

3.2 The Semantic Transformation

The semantic transformation of process models leverages knowledge embodied in the knowledge base. In the knowledge base, an explicit mapping is created between each of the process models to the generic workflow and thus an implicit mapping also exists between the process models themselves. Lose ends would probably exist as mentioned earlier that some constructs may not be directly mapped.

Since this is envisioned to be a universal framework, our postulation is that mechanisms will be reused from other existing implementations. For example, for a serialization format, standards such as the Workflow Process Definition Language (WPDL) (Junginger, 2000) used for the import and export of workflows between workflow management systems can be utilized or used as a basis. The actual mapping and transformation between process models (especially in cases where semantic mismatches exists) can consider mappings from existing work such as the WPDL standard mentioned earlier, mappings and transformations by (Lohmann et al., 2008) as well as formal and general definitions of the process modelling techniques as provided in literature would serve as input in the creations of the transformation mechanism.

4 DISCUSSION

The most obvious benefit of model transformation is that business process modelling can be performed at any level, and in any process modelling technique (perhaps because of the technique's simplicity or intuitiveness as is the case with UML). This is because you can model with a language of choice and then convert to a more formal tool (e.g. petri nets) for analysis. The potential benefits of this framework emanates from a knowledge base (especially one that is generic) can be reused and shared between projects. Ontologies that describes a construct of interest for example can always be imported. Such has been with the case with prominent ontologies like the dublin core and the friend-of-a-friend(FOAF). The framework is also attractive because of the structure of the ontology which promotes flexibility. Whenever the need to incorporate a new modelling technique arise an ontology for that particular model can be defined and mappings then can be create between the constructs of that ontology and those of the generic ontology.

5 CONCLUSIONS

In this paper we have proposed a universal framework for workflow analysis. We have discussed the notions relating to the knowledge base for the framework, its transform function and the universal analysis engine. The framework was viewd to have attractive traits such as the flexible and extensible structure of the knowledge base, opening of possibilities to model in virtually any language of choice and the universal coverage of the analysis engine. While this work is preliminary, it would close gaps in the business process modelling by offering a universal playground for manipulation of workflows and business processes.

REFERENCES

Almeida, T., Vieira, S. C., and Casanova, M. A. (2004). Flexible workflow execution through an ontology-based approach. In *Workshop on Ontologies as Software Engineering Artifacts (OOPSLA)*.

Atsa, R., Marcel, E., Ndjodo, F., and Aloo, G. A. (2011). A formal framework for business process modeling. *International Journal of Computer Applications*, 12(6):27–32.

Blake, M. and Nowlan, M. (2008). Taming web services from the wild. *IEEE Internet computing*, 12:62–69.

Cardoso, J., Mendling, J., Neumann, G., and Reijers, H. (2006). A discourse on complexity of process models. In Eder, J. and Dustdar, S., editors, *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science*, pages 117–128. Springer Berlin / Heidelberg.

Cardoso, J. and Sheth, A. (2005). *Introduction to Semantic Web Services and Web Process Composition*, volume 3387 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Charfi, A. and Mezini, M. (2007). AO4BPEL: An Aspect-oriented Extension to BPEL. *World wide web*, 10:309–344.

Consortium, W. W. W. (2009). *OWL 2 Web Ontology Language, Document Overview: W3C Recommendation 27 October 2009*. W3C. http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.

Fan, W. and Weinstein, S. (1999). Specifying and reasoning about workflows with path constraints. In Hui, L. and Lee, D.-L., editors, *Internet Applications*, volume 1749 of *Lecture Notes in Computer Science*, pages 25–54. Springer Berlin / Heidelberg.

Gillespie, M. G., Stacey, D. A., and Crawford, S. S. (2011). *Designing Ontology-Driven System Composition Knowledge and Processes to Satisfy User Expectations (in publication)*. Lecture Notes in Computer Science. Springer-Verlag.

Group, O. M. (2009). *Ontology Definition Metamodel - OMG Document Number: formal/2009-05-01*. OMG.

Gruber, T. R. (1993). Toward principles for the design of ontologies used for knowledge sharing. In *International Journal of Human-Computer Studies*, pages 907–928. Kluwer Academic Publishers.

Hlomani, H. and Stacey, D. A. (2009). An ontology driven approach to software systems composition. In *International Conference of Knowledge Engineering and Ontology Development*, pages 254–260. INSTICC.

Jenz, D. E. (2003). Simplifyingthe software development value chain through ontology-driven software artifact generation.

Jovanovic, J., Siadaty, M., Lages, B., and Spors, K. (2011). IntelLEO workflow ontology.

Junginger, S. (2000). The workflow management coalition standard wpdl: First steps towards formalization. In 7^{th} *European Concurrent Engineering Conference (ECEC2000)*, pages 1–6.

Kim, J., Gil, Y., and Spraragen, M. (2010). Principles for interactive acquisition and validation of workflows.

Journal of Experimental and Theoretical Artificial Intelligence, 22(2):103–134.

Lohmann, N., Verbeek, E., and Dijkman, R. (2008). Petri net transformations for business processes - a survey. *Transactions on Petri Nets and Other Models of Concurrency*.

Nyulas, C., OConnor, M., Tu, S., Buckeridge, D., Okhmatovskaia, A., and Musen, M. (2008). An ontology-driven framework for deploying jade agent systems.

Salimifard, K. and Wright, M. (2001). Petri net-based modelling of workflow systems: An overview. *European Journal of Operational Research*, 134(3):664 – 676.

van der Aalst, W. (2003). Pi calculus versus petri nets: Let us eat ”humble pie” rather than further inflate the ”pi hype”.

van der Aalst, W. and Hofstede, A. H. M. T. (2002). Yawl: Yet another workflow language. *Information Systems*, 30:245–275.

Wang, S., Shen, W., and Hao, Q. (2005). Agent based workflow ontology for dynamic business process composition. *International Conference on Computer Supported Cooperative Work in Design*, 1.

Weske, M., Vossen, G., and Puhlmann, F. (2006). Workflow and service composition languages. In Bernus, P., Mertins, K., and Schmidt, G., editors, *Handbook on Architectures of Information Systems*, International Handbooks on Information Systems, pages 369–390. Springer Berlin Heidelberg.

wide Web Consortium, W. (2004). OWL-S: Semantic Mark-up for Web Services. ”http:// www.w3.org/ Submission/ OWL-S/”.