META MODEL FOR TRACING IMPACT OF CONTEXTUAL INFORMATION EVOLUTION IN WEB-BASED WORKFLOWS

Jeewani Anupama Ginige and Athula Ginige
AeIMS Research Group, University of Western Sydney, Locked Bag 179, Penrith South DC NSW 1797, Australia

Keywords: Business process, process evolution, contextual information, web-based workflows, process modelling.

Abstract: Environment that shapes a *business process* consists of regulations, policies, guidelines, goals, etc. referred to as Contextual Information (CI) both external and internal to the organisation. In today's global and competitive business world evolution and changes in CI, forces business processes to change. When processes are supported with *web-based workflows*, CI evolutions are required to be reflected in already automated systems, via process models. Due to limitations of current modelling tools, process related models fail to encapsulate CI that associates process elements to its environment. This creates inconsistencies and errors when trying to change implemented systems to reflect high-level CI changes. To address this, we propose a model, which allows tracing high-level CI changes down to the implementation level artefacts. Such a model needs to map the complex correlation between *CI,* all *process elements* (object, participants, actions and process flow rules), various *web-based workflow artefacts* (data, code and UIs) to a *types of changes* (modify, add and delete). This holistic view of the proposed model makes this research standout among other research work in process evolution area.

1 INTRODUCTION

Organizations have processes to manage their business, which could be automated with the help of ICT. The paradox of automating business processes is their desire to change (Narendra, 2000). In today's global world, process environment that consists of Contextual Information – CI (policies, rules, goals etc), changes frequently. When processes are automated, it is required to alter implementation level artefacts to reflect high-level CI changes (Han, Sheth, & Bussler, 1998). At present this is done (largely) by humans in two phases. First, high-level CI changes are reflected into models. Then these models introduce changes into implemented systems. The ability to do this flawlessly rest with; human's capability to reflect CI changes in models cohesively and automated system's ability to adapt to changes.

When processes are relatively large and complex, the task of reflecting high-level CI changes in models and implemented systems, could lead to errors and inconsistencies. While some previous workflow evolution researches address the issue of making automated systems adaptable, the problem of reflecting high-level CI changes in implemented systems is not researched adequately. Thus, work here is focused on addressing the gap of linking CI with implementation artefacts, via process models.

In order to understand the specific research question and to define the scope of this research, first the 'big picture' of process automation is discussed below.

1.1 Paradigm of Process Automation

The 'big picture' of paradigm of business process automation consists of four levels; pragmatics, semantics, syntactic and implementation (figure 1). This model extends the 'workflow life cycle' introduced by (Zur Muehlen, 2004). The extensions are; i) new level named pragmatic to record CI which shapes a business process and ii) identification of certain information that gets 'leaked-out' or 'injected-in' at the transformation from one layer to another.

In figure 1, each box indicates a step towards process automation. The downward arrows represent the flow of information from one level to another that helps to construct the component parts at the target level. The dotted outward arrows represent the

Anupama Ginige J. and Ginige A. (2007).
META MODEL FOR TRACING IMPACT OF CONTEXTUAL INFORMATION EVOLUTION IN WEB-BASED WORKFLOWS.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - ISAS*, pages 410-415
DOI: 10.5220/0002377104100415

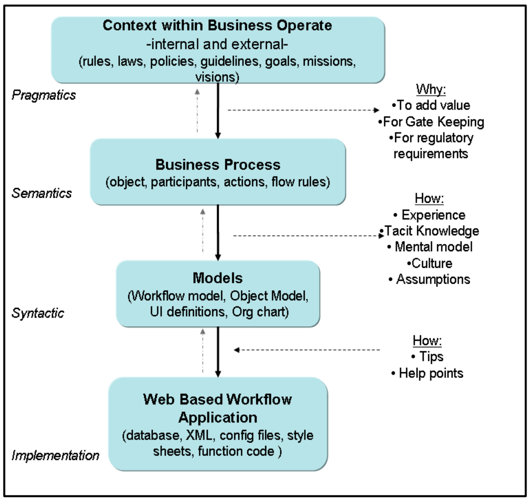

Figure 1: Paradigm of Business Process Automation.

'leakage' of information. The inward arrows illustrate the artificial 'injection' of information, which does not flow from the main channel. The upward dotted arrows from one level to another represent the influence and certain information flow that lower levels may have on re-shaping the upper levels (Maus, 2001). This concept of re-shaping upper levels as a result of lower level changes is outside the scope of this paper.

Let us breifly explore each of these levels.

Pragmatic Level - CI is the environment that sets the scene and need for a business process (Ramesh, Jain, Nissen, & Xu, 2005). CI usually contains policies, goals, strategic planes etc.; both external and internal to the organisation.

Semantic Level – Presentes actual business process. Here processes are not necessarily visualised into formal models. However there is a general understanding among participants on how things are done in these 'invisible processes' (Senge, 1994). The visualisation of processes takes place when processes are re-engineered or automated (McCormac & Rauseo, 2005).

Syntactic Level – Presents processes getting visualised into models such as workflow models, object model, organisational chart, etc. Most tools available for process element modelling are focused on capturing information required for implementation only. Thus result in leaking tacit knowledge; such as experience, mental models, culture, etc.

Implementation Level- Presents the implemented system, web-based workflow. There are three types of artefacts; i) to store persistence information (database, XML, documents, etc.), ii) to record logic (code, configuration files, etc.) and iii) for humans to interact - user interfaces, defined using (stylesheets, templates, etc.). To assist the use of implemented system certain tacit knowledge may get fed back, in the form of help points, tips for use, FAQs, etc.

In the 'paradigm of process automation', we discuss issue of reflecting changes in high-level CI to the implementation level. To solve this problem we propose a meta-model. This model links the complex correlation between the following; references to CI, all process elements (object, participants, actions, flow rules), various web-based workflow artefacts (data repositories, function code, UIs) and different types of changes (modification, add, delete) that may take place.

A detail discussion on the meta-model is presented in the section 3. Section 4, validates this meta-model is by applying through an emperical strudy. Next, we will discuss some previous work, which are closely related to this research.

2 PREVIOUS RELATED WORK

There are numerous researches that discuss handling of process evolution through workflow evolution. Some of the notable contributions come from (Bachmendo & Unland, 2001; Casati, Ceri, Pernici, & Pozzi, 1998; Chiu, Li, & Karlapalem, 2001; Ellis, Keddara, & Rozenberg, 1995; Governatori, Rotolo, & Sadiq, 2004). This is not an exhaustive list due to space limitations. Most of these previous work is concentrated at the syntactic and implementation levels in relation to figure 1.

Though not directly related to process evolution, there is some workflow research that attempts to link CI with process definition such as (Dong, Chen, Yin, & Dong, 2002; Maus, 2001; van der Aalst, Kumar, & Verbeek, 2003).

Ramesh's (Ramesh et al., 2005) work bares a close resemblance to the goals that we plan to achieve in our work, in terms of linking CI evolution as a way of guiding changes to business processes. Their work particularly focuses on identifying the re-design needs of business processes according to CI changes. While the knowledge management approach used in this work is acknowledged, their work differs from ours due to following reasons. Firstly it only maps CI to the business process in the semantic level (see figure 1), where our aim is to have a mechanism to link this information up to the lower level implementation artefacts. Secondly the knowledge reasoning based automated approach may give recommendations for re-design that are not particularly practical in a certain organisation, for example due to budgetary constraints. Hence it gives the need for human intervention for process changes with all the relevant information provided to them.

Most of the above work concentrates on evolution of *flow control* aspect only. Works of (Reichert & Dadam, 1997; Zhang & Wang, 2005) identifies the importance of linking object data with workflow models, but in a different perspective to our work presented here.

According to the literature we have reviewed, it is apparent that there is a gap in linking the CI to workflow level artefacts, with a holistic approach. Therefore we in the next section, present a meta-model to address this gap.

3 EVOLUTION META-MODEL

The meta-model (figure2) consists of four parts; a) process, b) set of registries, c) mapping of elements and d) an evolution mapping.

a) ***Process Elements –***

The representation of the process elements is based on the (WfMC, 1999)'s meta-model that defines the top level entities of a workflow. As depicted (figure 2) the processParticipants and processActions refer to various roles that perform workflow actions to reach the end goal of the process. wfRelevantData refer to various instance level rules that the workflow should refer to in the enactment of the process. In addition it includes the process object (processObject) and a mechanism to refer to other information (in the entity otherSupportInfo), as help files, FAQ's, etc..

b) Registries –

Our work is not aimed at automating the evolution process by synthesising CI changes as proposed in (Ramesh et al., 2005). This is due two reasons; firstly, we believe that when provided with all the necessary information human business analysts are much more capable of making appropriate decisions according to the domain requirements, which may not be captured in any of the CI. Secondly, there are inherent difficulties in modelling and capturing CI into a formal representation, as those come from variety of sources and formats; such as external or internal policy statements, guidelines, goal statements, laws, regulations, etc. Therefore, we propose only to refer to the CI in a registry. In the proposed model there are three types of registries to keep record of; i) CI (contextRegistry), ii) different kinds of models used for process automation (modelRegistry) and iii) various web based artefacts used to implement the process (registryOfImpleArtefefacts). The CI registry will record references to the particular CI in the following format;

```
scope (internal | external), type (law |
regulation | policy | ..), section,
location (URL | document | ..),
effectFrom, expiryDate, overidingContext
<self reference>
```

The overridingContext is a self-reference that denotes any CI, which is likely to override that particular entry.

Similarly, the registries that keep record of various models and web-based workflow artefacts gives relevant reference to these components.

c) ***Element Mapping –***

Element mappings indicate the relationship between process elements and registry elements identified in b) above. The contextMapping entries are similar to the format (The underline word denotes a foreign key

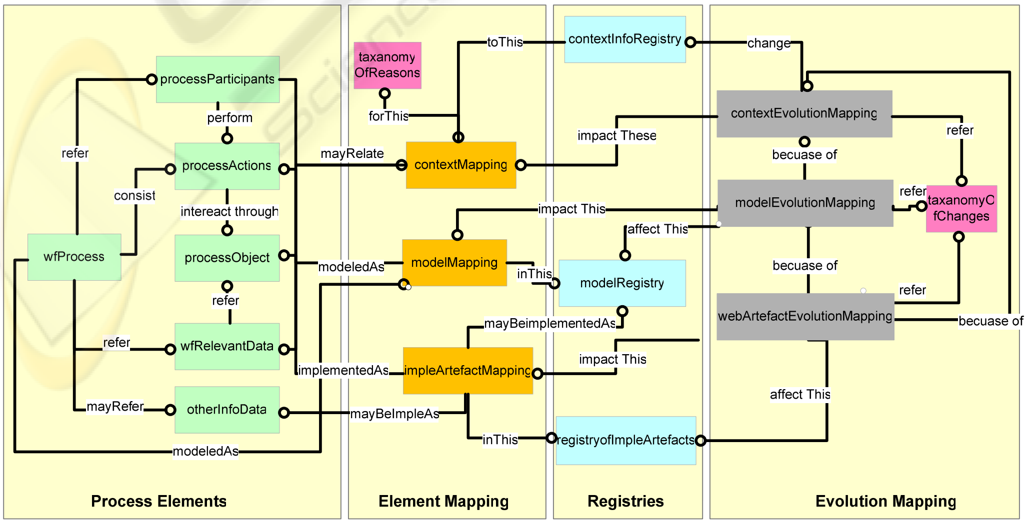

Figure 2: Meta-Model to Support CI Evolution.

relationship with another entity in this models.);

```
this process element is introduced for
this reason in this context
```

The *reasons* for introducing various types of *process elements* are three fold; for value adding to the end goal of the process and/or as a gate keeping function and/or for any legislative obligations.

The modelMapping denote the link between various models created for the process automation with various process elements. An entry in the modelMapper reads as;

```
this process element may be
represented in this model
```

Some process related additional information such as help files or some static pages would usually be directly implemented without being modelled. However, the other process related elements such as process object elements may be modelled in a class diagram. These models are used to create the implementation artefacts such as DBMS, XML files, text files, etc. Therefore, to capture both possibilities an entry to impleArtefactMapping reads as;

```
this process element OR model is
implemented in this workflow artefact
```

d) ***Evolution Mapping –***
Evolution mapping is one of the outcomes of this meta-model which was inspired by the work of (Felici, 2003). There authors introduces a model to keep track of requirement changes in software applications. Similar to that the evolution mapping not only allows the business analyst to trace the components requires changes according to the CI changes, but also to keep record of each change over a period of time.

Basically there are three types of changes in, i) CI (contextEvolutionMapping), ii) models (modelEvolutionMapping) and iii) implementation artefacts (webArtefactEvolutionMapping).

Entries in contextEvolutionMapping would read as;

```
as a result of modify|add|delete of
this context these process elements
need to modify|add|delete
```

Entries in modelEvolutionMapping would read as;

```
as a result of process-element-
evolution these affected models need
to modify|add|delete
```

The webArtefactEvolutionMapping records the following mapping;

```
As a result of model-evolution OR
process element-evolution these
affected web workflow artefacts need
to modify|add|delete
```

The referential information used (taxonomyOfReasons and taxonomyOfchanges) are not considered to be an integral part of the meta-model. The taxonomyOfReasons capture the three types of reasons identified in the paradigm of process automation; for value adding purposes, for gate keeping purposes and for legislative reasons, and link them with appropriate process element for informing the business analysts the reason for process elements existence. The taxonomyOfChanges identify the three type of schema changes that can take place; modify, add and delete, which was inspired by the early works of (Banerjee, Kim, & Korth, 1987) in database evolution and requirement evolution work of (Felici, 2003).

In the next section, a practical use of this model is presented using a process modelling automation task carried out for a tertiary education institute in Australia.

4 EMPIRICAL VALIDATION

The courses approval process in University of Western Sydney (UWS), Australia is shaped according several CIs. At the highest level it is affected by the Higher Education Act 2003 of Australia. Then there are organisational (university) and departmental (college and school) level guidelines, goals and values, which shapes the courses approval process.

The use of the proposed model will be demonstrated using the following example related to the course approval process of UWS;

Example - *As indicated in the UWS mission statement one of the values is ‘collegiality and participatory decision-making’. To support this clause the courses approval process elements have the following; Process object has the facility to attach an ‘internal consultation report’. There is a process action to ‘upload the consultation report’ by the actor ‘project manager’ and endorsement action by the ‘associate dean academic’ to ensure the ideas raised are adequately incorporated to the course under development by the project manager.*

For the above scenario let us consider a variety of entries that would be made in the meta-model in figure 2.

CI registry entries

```
• contex18, Internal, value statement,
item4
http://www.uws.edu.au/about/university/mission,
2004, 2008, Australian Higher Education
Act 2003
```

Model registry entries

```
• model67, objectmodel, classdiagram,
http://portal.cbeads.org/ocasstage/main.pl  model
documents, version4
```

```
• model68, UImodel,
dataUIMappingSheet,
http://portal.cbeads.org/ocasstage/mian.pl
model documents, version3
```

Implementation artefacts registries entries

```
• artefact5, database.ocasMaster,
table.courseInstance,
http://portal.cbeads.org/ocasstage/main.pl
• artefact6,
UI.uploadInternalForumnReport,
http://portal.cbeads.org/ocasstage/main.pl
• artefact7,
UI.EndorseInternalForumnReport,
http://portal.cbeads.org/ocasstage/main.pl
```

contextMapping entries

```
• cm100,
courseObject.internalForumConsultation
Report is introduced for value adding
purposes in the context18
• cm101
processActions.uploadInternalConsultai
tonReport is introduced for value
adding purposes in the context18
• cm102
processActions.EndorseInternalConsulta
tionReport is introduced for gate
keeping purposes in the context18
```

Model mapping entries

```
• mm10,
courseObject.internalForumConsultation
Report is represented in model67
• mm11,
processActions.uploadInternalConsultai
tonReport is represented in model68
• mm12,
processActions.EndorseInternalConsulta
itonReport is represented in model68
```

Implementation artefact mapping entries

```
• am11, mm10 is implemented in this
web-workflow artefact5
• am12, mm11 is implemented in this
web-workflow artefact6
• am13, mm12 is implemented in this
web-workflow artefact7
```

Let us assume aforesaid value statement (`contex18`) get removed from CI. Once this is notified to the business analyst, he would query the meta-model using a query similar to;

```
SELECT process elements FROM
contextMapping WHERE context=Context18
```

The results would inform him that the course object attribute named 'internalForumConsultationReport' and the two-process actions uploadInternalConsultationReport & EndorseInternalConsultationReport are the affected process items. Business analyst would then make a decision (in consultation with process owners), changes that need to happen in elements and records them in the evolutionMapping tables as follows;

```
• conevo97, as a result of delete of
this context18 these
courseObject.internalForumConsultationRe
port need to delete
• conevo98, as a result of delete of
this context18 these
processActions.uploadInternalConsultaito
nReport need to delete
• conevo99, as a result of delete of
this context18 these
processActions.EndorseInternalConsultati
onReport need to delete
```

Using the following query, the models that are affected due to above changes can be identified.

```
SELECT models FROM modelMapping WHERE
element="identified process elements"
```

Results would indicate that the two models model67 and model68 are affected. Hence, the business analyst would record the changes in the modelEvolutionMapping as follows;

```
• modevo87, As a result of conevo97
model67 need to modify
• modevo88, As a result of conevo98
model68 need to modify
• modevo89, As a result of conevo98
model68 need to modify
```

Then the programmers would query the meta-model as follows;

```
SELECT artefacts FROM
impleArtefactMapping WHERE
element="identified process elements" or
model="identified models"
```

The results would indicate that the artefact5, artefact6 and artefact7 require changes. Hence, the programmer would either use changed models to generate the new artefacts or do the changes manually. These changes are recorded in webArtefactEvolutionMapping as follows;

```
• artevo77, As a result of modevo87
artefact5 need to modify
• Artevo78, As a result of modevo88
artefact6 need to delete
• Artevo79, As a result of modevo89
artefact7 need to delete
```

Based on above responses 'consultation report' attribute of the course object can be removed and the UIs to perform the two tasks 'upload internal forum consultation report' and 'endorse internal forum consultation report' can be deleted. Most importantly, when programmers are making these changes it can be assured that errors and inconsistencies are minimal.

5 CONCLUSION

This paper attempts to solve the problem of reflecting high-level CI changes and evolutions in web-based workflow artefacts, with minimal errors and inconsistencies. The approach used here is the use of a meta-model, which reference to all-important component parts of each level in automation framework; from pragmatic level down to implementation. With an empirically validation it was demonstrated how this meta model could be used for the purpose of tracking impact of a high-level CI change down to the implementation level.

Various future researches could begin based on this work. For example, first it requires identifying appropriate data structures to represent all types of registries, particularly process elements and various dependencies among them. Then an implementation of the proposed meta-model is required, for practical use.

REFERENCES

Bachmendo, B., & Unland, R. (2001). Aspect-Based Workflow Evolution. Workshop on Aspect-Oriented Programming and Separation of Concerns (Lancaster).(2001).

Banerjee, J., Kim, H. J., & Korth, H. F. (1987). Semantics and implementation of schema evolution in object-oriented databases. ACM SIGMOD Record, 16(3), 311-322.

Birnbaum, R. (2000). The Life Cycle of Academic Management Fads. The Journal of Higher Education, 71(1), 1-16.

Casati, F., Ceri, S., Pernici, B., & Pozzi, G. (1998). Workflow Evolution. DKE, 24(3), 211-238.

Chiu, D. K. W., Li, Q., & Karlapalem, K. (2001). Web-Based Workflow Evolution in ADOME-WFMS. Proceedings of the Second International Conference on Advances in Web-Age Information Management, 377-389.

Dong, X., Chen, G., Yin, J., & Dong, J. (2002). Petrinet-based Context-related Access Control in Workflow Environment. Paper presented at the The 7th International Conference on Computer Supported Cooperative Work in Design.

Ellis, C., Keddara, K., & Rozenberg, G. (1995). Dynamic change within workflow systems, Proceedings of conference on Organizational computing systems (pp. 10-21).

Felici, M. (2003). A Formal Framework for Requirements Evolution. Edinburgh, United Kingdom: LFCS, School of Informatics, The University of Edinburgh.

Governatori, G., Rotolo, A., & Sadiq, S. (2004). A model of dynamic resource allocation in workflow systems. Paper presented at the Proceedings of the fifteenth conference on Australasian database-Volume 27.

Han, Y., Sheth, A., & Bussler, C. (1998). A Taxonomy of Adaptive Workflow Management. Paper presented at the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW-98), Workshop Proceedings - Towards Adaptive Workflow Systems Seattle, WA, November 1998.

Maus, H. (2001). Workflow Context as a Means for Intelligent Information Support. Paper presented at the Third International and Interdisciplinary
Conference, Dundee, UK.

McCormac, K., & Rauseo, N. (2005). Building an enterprise process view using cognitive mapping. Business Process Management Journal, 11(1), 63.

Narendra, N. C. (2000). Adaptive workflow management—an integrated approach and system architecture. Proceedings of the 2000 ACM symposium on Applied computing, 858-865.

Ramesh, B., Jain, R., Nissen, M., & Xu, P. (2005). Managing context in business process management systems. Requirements Engineering, 10(3), 223-237.

Reichert, M., & Dadam, P. (1997). A Framework for Dynamic Changes in Workflow Management Systems. Paper presented at the DEXA Workshop.

Senge, P. (1994). The fifth discipline: The art and practice of learning organizations. NY: Doubleday.

van der Aalst, W. M. P., Kumar, A., & Verbeek, H. M. W. (2003). Organizational modeling in UML and XML in the context of workflow systems. Paper presented at the Proceedings of the 2003 ACM symposium on Applied computing.

WfMC. (1999). Workflow Management Coalition Interface 1: Process Definition Interchange Process Model (Specification): Workflow Management Coalition.

Zhang, S., & Wang, B. (2005). The Research on Decision Approach of Data Dependence in Dynamic Workflow System, Parallel and Distributed Computing, Applications and Technologies, 2005. PDCAT 2005. Sixth International Conference on (pp. 1027-1029).

Zur Muehlen, M. (2004). Workflow-based Process Controlling: Logos-Verl.