IADIS International Journal on Computer Science and Information Systems Vol. 1, No. 2, pp. 100-116 ISSN: 1646-3692

A SYNTHESIS APPROACH FOR DERIVING REFERENCE MODELS FOR SOA FRAMEWORKS

Balbir Barn¹, Hilary Dexter², Samia Oussena¹, Jim Petch²

¹Thames Valley University, Wellington Street, Slough, SL1 1YG, Berkshire, UK ²University of Manchester, 186 Waterloo Place, Manchester, M13 9PL, UK

ABSTRACT

Improving business processes and services is a challenge that can be met by a model-driven approach to service design and development. This approach rests on defining reference models of the enterprise business processes that will become the drivers of service frameworks. As part of a national program for developing such models within a Service Oriented Architecture (SOA) framework for e-learning and research in higher education, a canonical reference model for course validation was used to demonstrate the feasibility of the approach. Course validation processes in four UK Higher Education Institutions (HEIs) were analysed and modelled using interviews and process documentation. Each institution's process was modelled with UML Activity Diagrams and its domain information with Class Diagrams. The four models were synthesized into a single canonical reference model of the validation process. This required resolving process model structures and element granularity. Synthesis of the canonical model demonstrated a methodological basis for developing service specifications, within a SOA framework that could serve all institutions in the sector.

KEYWORDS

Reference model, process model, course validation, service oriented architecture

1 INTRODUCTION

The design and implementation of enterprise information systems have seen an evolution from implementation on centralized mainframes to bespoke application development using clientserver architecture and on to enterprise application integration using distributed architecture principles. There is currently a convergence to so-called Service Oriented Architecture (SOA) for application design (Ort 2005). A separate move towards a focus on application integration led by business process modelling is being enabled by the developing SOA principles. There is an emphasis on driving this from a solid understanding of business processes and aligning developed or procured services to support those processes (Frankel 2005, Blinco et al., 2004).

One way to provide services in close alignment with business processes is to adopt a formal model-driven development process that can link the business processes to the sets of implemented services required to support them, managing the whole service provision lifecycle. The starting point for this model-driven development is a business process model that is rich, adaptable and precise enough to become a reference model that is then used to direct the provision of services.

The research question addressed in this paper is whether it is possible to combine variants of models of a particular business process from different enterprises in the same sector to produce a useful, single, flexible, customizable canonical process that can become a reference model for service provisioning across the sector. The paper discusses an approach to providing a flexible, customizable template or canonical process model that each organisation can implement with a set of ready-to-use or developed services. These services may be discovered in frameworks set up for the whole sector or within organisations' own frameworks. This paper tackles the first part of the SOA service provision lifecycle, that is, the analysis of business processes and synthesis of a process model from multiple organisations, including the information flows within the process, from which a system model for service specification may be developed.

2 THE PROBLEM

The idea of Service Oriented Architecture is both confusing and challenging. It is confusing because there are a myriad of terms that seem to lead to the same basic notions. It is challenging because it is the direction in which the IT industry is heading led by the vision of the virtual enterprise and this is still in to uncharted territory. The vision of the virtual enterprise where applications in different enterprises are connected via an overarching business process (e.g. an order procurement process that goes across enterprise boundaries) is predicated on the need for standards and frameworks that go beyond the wire standards currently available, for example the Web Services protocol, SOAP and interface specification language, WSDL, (Fremantle et al. 2002).

Achieving a sufficient level of process and information integration is a challenge facing the industry and is currently the focus of new developments in service oriented architecture. Given the pervading confusion surrounding SOA, what is needed is a clearly articulated standard framework that addresses issues of terminology, functional requirements, information integration requirements and method/techniques requirements. Part of such a framework is the notion of the reference model. This has been recognized and a new initiative led by key industry players has been announced to develop a SOA reference model (Oasis 2005). Unfortunately, reference models themselves are the product of interpretation and of modelling preferences with no standard, industry-tested structures or methods yet available.

2.1 Reference Model Discussion

The debate about the value of information modelling has a long history (Chen 1976) and continues to be relevant to the IS research agenda (Wand and Weber 2002). Information modelling itself can be usefully partitioned into three constituent parts: *Application information modelling* – where an application information model is used to represent a particular solution to a specific application requirement; *Enterprise information modelling* –

where an information model is used to represent potential solutions to multiple types of application within a particular enterprise; And *Reference modelling* – where the model represents a general solution to a particular class of problem in a specific domain.

The current general trend from bespoke development to tailoring and adaptation (exemplified for example by the SAP approach (Themistocleous et al. 2001)) means that the importance of *Reference modelling* is increasing and the oft-cited benefits include: re-use of knowledge, a rise in quality, and corresponding reductions in risk, cost and time (Esswein et al. 2004, Thomas 2006). Even allowing for these attributed benefits, Thomas (2006) further argues that "no uniform grasp of the term reference model exists". This confusion partially arises out of the tendency to declare Application information models and/or enterprise Information models as "reference models". An instructive discussion of this tendency is provided by Thomas (2006).

Given this confusion about what is meant by a reference model and the recent activity in the development of so-called reference models in the UK HE sector it is instructive to discuss reference models and state what we mean by the term. We are interested in this because we believe that there is value in using appropriate reference models in taking a model-driven approach to application development so that a business process model can become the principal element in a reference model to guide the provision and use of services and current trends in usage provide a risk to these objectives.

Reference models and the specification of standards have played a key role in the integration of information systems and there is some evidence of the need to relate reference models and standards. A key player in helping developing these standards is the Object Management Group (OMG). This organisation provides a sophisticated mechanism for the development and specifications that can be implemented by suitable vendors as exemplified in the development of the Unified Modelling Language (UML) Specification (OMG 2005); Common Object Request Broker Architecture (CORBA) Specification (OMG 2004) and the Enterprise Distributed Objects (EDOC) specification (OMG 2004).

While Esswein et al. (2004) state that a reference model can be constructed in one of two ways, best practice or theoretical analysis, they do not address the underlying issue of declaration of a model as a reference model. Thus it may be more appropriate to identify emergent properties or characteristics for reference models and thereby articulate an appropriate definition.

One approach for seeking clarity on what is meant by a reference model is to review the outputs of several existing reference models and from these synthesize a definition of a reference model. The following key sources are used:

Reference Model for an Open Archival Information System (OAIS)

http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf

Workflow Reference Model

http://www.wfmc.org/standards/docs/tc003v11.pdf

Topic Maps Reference Model

http://www.isotopicmaps.org/TMRM/TMRM-latest-clean.html

Java Security Reference Model

http://java.sun.com/security/SRM.html

Sharable Content Object Reference Model (SCORM) http://www.adlnet.org/

ISO Open Systems Interconnect Reference Model (OSI) http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=20269

Inspection of the reference models points to emerging common features. There is an effort made to define common terms, a well-defined framework for extending aspects of the specification, attempts to define a general, overarching structure for the domain and a focus on interoperability and standardization. These aspects are the *lingua franca* for a reference model. If we then consider the software engineering community as a specific example, we can observe that in this community reference models provide a common language and define structural relationships. In addition, a reference model specifies the logical structure of the external interfaces to other systems with enough precision to be practically realizable in an efficient manner while remaining deliberately independent of any particular implementation. Such a framework can then be used for specifying requirements and performance benchmarks in procurement or development of complete systems comprising people, processes and technology. The codification of the interface structure will also encourage the development of software tools to enable the development of systems that conform to a particular reference model. Thus the reference model will provide a strong (perhaps enforceable) steer on how systems for a particular domain (and with specific requirements on interoperability) should be implemented.

While this research was based on a limited review of existing reference models, a more detailed and extensive review focused on reference models for Business Process Modelling was conducted by Fettke, et al. (2006). The review provides for an examination of approximately thirty well known reference models and utilizes a framework for the articulation of key features, requirements and characteristics. The framework is a set of criteria classified into General Characterizations, Construction and Application. This delineation appears to be fairly arbitrary as characteristics could conceivably be located in other groups. For example, the "Domain" characteristic is located under Construction, but could easily be located in General Characterization or Application. Esswein et al. (2004) provide a generic set of requirements of a reference model including universal validity, completeness, adaptability, extendibility, usability, re-usability and acceptance. Of these requirements, universal validity and completeness need definition. The universality of a reference model refers to notions where it is possible to derive multiple models from a universal model where the derived model follows rules and constraints and is complete in terms of the originating model. Completeness refers to an orientation within a methodical framework (Esswein et al. (2004). That is, all necessary structures for process and data are integrated, thus forming a model-driven architecture. These requirements thus allude to the actual artefact(s) of the reference model.

We propose a number of additional enhancements:

It is not certain whether method, guidance, rules and language to produce artefacts are also part of the reference model. We propose that such guidance is appropriate (Dexter and Petch 2006, Barn, Dexter, Oussena and Sparks 2006) and could be regarded as an additional classification category to those proposed by Fettke et al (2006). Indeed, we are currently investigating content and mechanisms for delivery of guidance to support the reference model that we have developed and described in this paper.

We agree with Thomas (2006) that the assertion by developers of a reference model is not sustainable unless there is at least one application of the reference model – "This attribute can ultimately be proved only by way of the model being applied at once" (Thomas 2006, p22).

The ability to represent re-usable knowledge at different levels of abstraction is also a key characteristic of reference models. Indeed, acceptance of a reference model by a community can occur at different levels of abstraction. This requirement pre-supposes the existence of method frameworks and supporting guidance at different levels.

The reference model should additionally have a set of use cases or capabilities that will enable the model to function as a strategic planning tool. Such capabilities could include for example assessment of new requirements and services (functional and non-functional) a provision of support to service developers.

In summary, A reference model is based on a small number of unifying concepts and is an abstraction of the key concepts, their relationships, and their interfaces both to each other and to the external environment. A reference model may be used as a basis for education and for explaining standards and methods to a non-specialist and can be viewed as a framework for comparing architectures and operations of existing and future systems.

The task then, is to develop and evaluate a reference model for a specific problem domain that contains the elements described above. This is described in the remainder of the paper.

3 THE CASE STUDY

The e-Learning Framework (ELF) is an initiative by the U.K.'s Joint Information Services Committee (JISC) and Australia's Department of Education, Science and Training (DEST) to build a common approach to Service Oriented Architectures for e-learning (JISC 2005). Within this initiative, JISC has requested projects to develop reference models for a number of domain areas. Domains have been defined by CETIS (Centre of Educational Technology Interoperability Standards) Special Interest Groups (SIGS). A list of the domains and the role and purpose of the CETIS SIGs is described in their website (http://www.cetis.ac.uk). This paper presents our approach to developing a reference model for the "Enterprise" domain in the area of "Course Validation". Currently, Course Validation within ELF is un-developed in the sense that, there is neither an accepted definition, nor a reference model defined for it. For our purposes, given the absence of a definition, we define Course Validation to be: The process by which a judgment is reached as to whether or not a course and its modules, designed to lead to an academic award of a specified level, meet the nationally accepted criteria for that award. The Course Validation process is an important business processes within Higher Education Institutions (HEIs) and between HEIs and other institutions. New courses and the continuation of existing courses are the direct outputs of this process.

3.1 Scope

Our understanding of the scope of the application domain is as follows. Course Validation can include the specification of new courses at various levels (e.g. undergraduate and postgraduate). Course Specifications address areas such as rationale, appropriateness, justification, marketing analysis, resources required, economic viability of the courses, and detailed descriptions of the courses in terms of outcomes, aims and objectives and so on. Much of the scope of course validation is determined by local institutional constraints (e.g. relationship to other courses and university regulations) but there are wider requirements of

national bodies that impose a significant overhead on the developmental process for validating new courses.

Course validation is further complicated when we consider the modes of deployment of new courses e.g. based on learning technologies or on traditional modes of delivery to support students, and the mode of attendance e.g. full time, part time, continuous professional development and distance learning. E-Learning in particular, raises additional issues of complexity. There are no standards or benchmarks that allow us to understand the impact of course validation processes on qualifications which are delivered entirely using an e-Learning approach. Even though HEIs may differ in the implementation of business processes to support course validation the content and constraints imposed by external bodies such as the Quality Assurance Agency (QAA) provide some standardization for the validation process and its outputs. These constraints are a basis for defining a canonical business process for supporting course validation.

3.2 Needs and benefits

Course validation is typically a relatively well-defined business process within an institution which implements that institution's rules of governance in the production of course specifications. There may, however, be several ad hoc and different business processes for different types of course validation and for different circumstances as indicated above. Each of these factors presents risks for institutions and for collaborative working and indicates the need for a common reference model.

The need for a reference model comes also from the need to align internal processes. The validation process is the defining source of core information on programmes, courses and modules for an institution. This information is typically consumed by marketing, student records, finance (fees section), planning, and many other functional areas of institutions as well as by staff and students who manage selection and progression. By improving consistency and ensuring the availability of information at the right time and in the right place, alignment of internal supporting business processes will be possible.

4 METHOD

Our method is derived from elements of RUP (Rational Unified Process) (IBM-Rational 2004) and is strongly tailored to the delivery of our main artefact, a reference model for course validation. The method takes a model based approach using UML; is an application and adaptation of software engineering and is iterative. While the method addresses domain modelling, service interface specification and service implementation through to a test execution of the business process, this paper will focus on the domain analysis stage, and on evaluation of the reference model in supporting the service development process up to this point. This entails a study both of dynamic (process) and structural (data) information. Analysis of this stage requires the capture of roles/responsibilities (including teams), activities in the process, routes through the process, triggers, information consumed and produced by activities, constraints and interfaces with other information systems. The process followed by the project team to gather the required information in order to build a model of course validation for each institution entailed several iterations of interviews with stakeholders that

centred on empirical models of the individual institution's process that were refined at each iteration.

4.1 Modelling the process in each institution

Each institution's course validation process was modelled as an Activity Diagram with activities grouped into assemblies (nested activities) corresponding to stages in the business process that were referred to as such by those responsible for setting procedure. An activity was created for each discernible task or action, carried out by an individual or by a collaboration of people that could be seen to produce a defined output in the course validation domain. The output was either in the form of a document, a decision reached or an organisational structure such as a committee being readied for work. Items used and produced by the process activities were modelled as object flow states. The state changes in the lifecycle of key documents, such as going from "for review" to "approved", were captured in the activity diagram. The existence of guidelines or checklists for the execution of activities also was documented and placed in the object flow of the diagram. The description of each activity included the involved roles. If the activity was carried out by a group of people in collaboration (such as a committee) all the member roles and the rules controlling the frequency and ways in which that group operates were collected. The individual and group roles were represented as swimlanes (partitions) in the Activity Diagram and as Classes in the domain information model. Any constraints for activities were noted on the diagram and attached to the relevant activities. These were often based on availability of particular documents or of people for committee meetings. Alongside the Activity Diagram of each institution's business process, a UML Class Diagram was created in order to capture the set of elements and roles in the course validation domain. This domain information model was kept at a high level of abstraction with only the key relationships between the elements being included. An example is given in Figure 3 of part of one of the domain information models that illustrates the main kinds of elements and the level of abstraction.

4.2 Synthesizing the process models

Software design has always needed to consider the tension between designing for purpose for a specific client versus designing for potential re-use. This can be further characterised by the dichotomy of the two polar positions sometimes described as commonality versus variability – that is describing those software features that are common and fit for purpose for multiple clients and those features that are adaptable by the end-user or by the software provider in order to maximise software fit for a particular client. This debate on commonality and variability (C/V) has been elaborated from a methodology requirements perspective by Leishman (1999). Leishman provided an analysis of implementation technologies that provide re-use/adaptability support. The technologies included component based design, software packages and software design frameworks. This has been revisited and c/v has been subjected to further critique in the context of Enterprise Software (large scale business critical applications or packages) (Nordheim 2006). In particular, Nordheim presents a new viewing of the debate by using the lens of Dialectics theory to characterize C/V.

Dialectics is a form of rational thinking that provides a method for reaching an agreement on ideas about a phenomenon. It is based on the presentation and resolution of a proposition (or thesis) and its antithesis. The resultant dialogue between the opposing states creates a new balancing position that contains a synthesised view of the original phenomenon whilst at the same time providing opportunities for creating and refining knowledge of the phenomenon.

Nordheim (2006) provides an analytical structure for a dialectic view of C/V which we can utilize in our synthesis approach. Certain features (a subset) of the course validation business process should not be subject to change – the commonality thesis [**Thesis A**]. Variability is the antithesis representing change – those features that suit, or may be tailored to suit, specific requirements [**Thesis B** or let us say \neg **A**]. While the process of dialectic debate will result in a synthesis which we denote as $\neg \neg$ **A**, our approach to synthesis is case study based – that is we identify the set of features that meet the condition of $\neg \neg$ **A** through an analysis of a number of studies of organisations.

While common and varying features of software can be supported by a number of implementation technologies such as interface based design (a specification focus) or class inheritance frameworks (an implementation focus), understanding and supporting C/V tension earlier in the software lifecycle is much more limited. Jacobsen et al (1997) discuss how software product families / product line architectures can be developed by placing notions of designing for re-use at the centre of method architecture. In particular, variation at use case level is described. However, there is limited evaluation of the effectiveness of the methods and techniques described. The approaches presented are largely theoretical. Similarly The UML 2.0 Use Case Diagram provides extensive abstraction concepts for specifying and accommodating C/V but it is difficult to ascertain if features (beyond "extends" and "uses" relationships) are extensively used. Dobing and Parsons (2000) identified a research agenda for exploring the application of Use Case Diagram in more detail and in a more recent empirical study of the use of UML diagrams presented evidence that focused mainly on specification of use case narratives (Dobing and Parsons 2006).

Within the context of service oriented architectures where business process led development is a primary driver there is a greater need to identify and support C/V even earlier in the software lifecycle. The synthesis approach described in this paper provides a potential approach to documenting the features that could contribute to the synthesis ($\neg A$ as per the previous discussion) in a dialectic debate.

Benefits of early evaluation of a synthesis position (for example during the business process definition outlined in this paper) provide opportunities for the development of a rules based approach to handling variability. There is currently ongoing research which will be applying XML rules to utilize the synthesis position we describe in our business process definitions (Oussena et al. 2006).

Following the two iterations of interview and model refinement in each of the four institutions, a process model and a domain information model of each were prepared for the synthesis process. The required levels of granularity and abstraction had been directed by a set of guidelines for the modelling, written by the team prior both to the interviews and to modelling activities, but it was necessary to review all four models together to ensure that this had been achieved. A set of characteristics of the processes that had been modelled was proposed as the basis for comparison and synthesis of the four process models. These characteristics were also refined by the team's experience in actually building the models. The set of characteristics is shown in Table 1 with the strategy employed for each one's synthesis.

In producing the synthesis the resulting model was created to represent an aggregation of concepts. That is, we did not employ any "re-engineering" to optimize the process. Figure 1 below shows the pattern that was used for each of the characteristics in order to produce the canonical core model and the set of options or extensions that would be required to enable customization for any HEI.



Figure 1. A pattern for synthesizing the canonical model

5 RESULTS

In this section examples of the results obtained in this first part of the research and development effort are presented.

5.1 Process Model Sample

Figure 2 shows a section of a process model showing nested activities, object flow and constraints taken from part of one swimlane (the responsibility of a single role, the Programme Proposer) in an Activity Diagram. The academic quality staff who are responsible for the validation procedures were able to follow the workflow in the activity diagram and review it critically. Colour-coding of different kinds of things, such as documents and rules was found to be useful as an aid to communication of the various aspects of the process model: roles, activity sequence, object flow, decisions and constraints.



Figure 2. A section from a course validation process

5.2 Information Model Sample

Figure 3 shows a section of a domain information model for one of the institutions analysed. At this level of granularity the model facilitates discussion with the stakeholders about the kinds of things involved in the process, their interrelationships and their possible states during the process. As is clear from earlier discussion, this in itself is not the reference model rather it is one element of the overall integrated set of artefacts contributing to our candidate reference model.

Figure 3. Sample from one of the four domain information models

5.3 Model Synthesis Results

Using the model synthesis rules given in Table 1, the four individual models were merged into a single canonical model was carried out. Figure 4 shows the result for a small part of the synthesized process model showing the specialisation of activities.

Characteristic	Description	Synthesis Rules
Stage	The principal sections of the process i.e.	The principal stages were identified and aligned
	those sub-processes that made up the	across the four processes. The detailed activities were
	end-to-end process for course validation	compared within the bounds of the high level
	e.g. 'developing the business case'	alignment.
Activities	Executed by roles and collaborations of	Where possible, common activities were identified.
	roles. Activities were defined as having	Where a single activity differed between institutions it
	to produce some substantive change in	was modelled as an abstract activity with four
	the state of the system e.g. a new section	specialisations. If there was a sequence of activities
	in a document completed, a cost analysis	that differed across the institutions a control node was
	completed, a document approved.	entered and a sequence modelled for each.
Role	Primary responsibility for an activity.	The principal roles were aligned and a general name
	Roles could be individuals, teams,	allocated. The equivalent roles in each institution
	organisational units e.g. Dean,	were modelled as specialisations of the general role.
	development team, quality unit	Each activity in the synthesized process model was
		annotated with its responsible role, common activities
		with the general name and specialised activities with
		their own local role name.
Object Flow	Any items used or produced by (i.e.	An abstract object was created for each object, given
	flowing into or out of) the activities in the	a general name and then specialised for each
	process. These were usually documents.	institution.
Object Lifecycle	Key documents passed through a number	Each institution's document lifecycles were modelled

Table 1 Characteristics and rules for model synthesis

Characteristic	Description	Synthesis Rules
	of states as they were written and reviewed. These states form the lifecycle of the document e.g. 'course proposal' that goes from a blank template through stages of approval to final approval.	separately and the appropriate states placed on the specialised objects.
Events	An event was taken to be something that triggered a sequence of activities e.g. approval or rejection of a proposal.	Events were used in the synthesized model at the start of high level activities and to trigger activities by collaborating roles. The synthesis decisions for activities and their responsible roles determined event placement.
Constraints	Business rules that determined allowed states and conditions, such as who may execute an action, when tasks had to be completed, compliance to standards etc.	Constraints were preserved from their source models for specialised activities. If a common activity had been identified the constraint had a CASE statement for each institution.

Figure 4. Synthesis of activities

5.4 Reusable Process Patterns

Process patterns follow the well-established ideas of Design Patterns in that patterns capture best practice, are defined and described in a specific language and may be used or grouped together in collaborations (frameworks). Catalogues of process patterns (pattern languages) are not as well established as design patterns but there are some examples (Barros 2004, Barros 2005, Eriksson and Penker 2000). Barros describes process patterns at a macro level where he argues that the business of enterprise may be represented by four Macro patterns. These are described using the Integration Definition for Function Modelling (IDEF0) notation (FIPS 1993) and may be specialised by at the domain level. Eriksson and Penker (2000)

provide a collection of business process patterns and an extension of UML for process modelling.

In this work, during the analysis of the four processes, patterns emerged that seem to be potentially re-usable in enterprise business processes. Having a library of such reusable patterns should improve productivity in future business process analysis. The following is the preliminary set of process patterns that were identified during the process synthesis.

- Consultation with expert;
- Assessment and Approval;
- Convene Panel;
- Meet Conditions for Approval;
- Proposal Refinement;
- Sign-off Document;
- Roles Collaboration.

These patterns have been identified at the domain specific level but appear to have the potential for generalisation. They contributed to an understanding of this particular business process case in the wider context of producing a generic business process implementation reference model. This will be the subject of future research.

6 EVALUATION OF THE REFERENCE MODEL OBTAINED

Following the earlier discussion of the characteristics of a reference model, we set out the functions, or Use Cases, that would be required for a synthesized business process for SOA reference model to become operationally useful. The reference model was then evaluated against this set of use cases and the results summarized below.

Use Case 1: Provide a catalogue and visualisation of business processes, and subprocesses in the domain.

A visual model is provided as a UML Activity Diagram with the possibility of viewing sub-processes in separate diagrams. The stakeholders of the course validation model found the Activity Diagram a useful and intuitive view of their processes. The tool selected for creation of a process driven knowledgebase also permits the process to be displayed as a Work Breakdown Structure with the information about the roles, activities and process artefacts being entered through forms. Evaluation of the two 'views' of a process will be carried out in future work.

Use Case 2: Provide a visual model of the elements in the domain and their interrelationships.

The course validation knowledgebase contains a glossary of terms used, including synonyms. Local terminology may be mapped to this glossary. The 'elements' in the course validation domain are modelled in the domain information model as a UML Class Diagram. This model was of more use to the developers than to the business domain stakeholders, particularly in matters relating to establishing interoperability between reference models in the context of the e-Learning Framework.

Use Case 3: Provide a vocabulary for discussion of business process design and development.

The process driven knowledgebase for process modelling and service provisioning contains a glossary of terms used in capturing and visualising a business process; for synthesizing multiple process models into a canonical model and for using this as the entry point for service provisioning. The use of UML models proved to be a good basis for communication between related reference model projects in the JISC frameworks domain and it was generally agreed that non-technical people should acquire sufficient familiarity with this kind of modelling in order to participate fully in the design and development of reference models in their business domains.

Use Case 4: Provide a method for executing an MDA approach to supporting business processes with components implemented as Web Services.

The reference model developed here is based on an MDA compliant architecture for service provision to support a business process. The model types at the instance, model, metamodel and meta-metamodel layers are specified in the reference model architecture. Details of this architecture will be published at a future date. The method adopted in this work indicated that a model-driven approach could be taken wherein all the pertinent information was contained within the appropriate model element as defined by the reference model architecture. It was not necessary to have supplementary information held outside the reference model.

Use Case 5: Provide method support to component and Web Service developers.

This paper focuses on the business process modelling and synthesis stage of the development process and a process driven knowledgebase for this is currently being constructed. The lessons learned in the development of course validation services design and development have been written up as guidance elements and attached to a model of the development process. The method support is provided as a process context sensitive web application (for future publication). The development activities carried out by the team provided a good starting set of guidance for future developers coming to this task which will be added to as more experience is gained.

Use Case 6: **Provide a business process driven knowledgebase for the domain process model**.

The business process guidance and business rules were collected from the range of stakeholders. This enabled a rich knowledgebase to be generated for the synthesized and customized course validation process. The Activity Diagrams of the stages of course validation provided the process context in sufficient detail to permit the creation of an operational process driven knowledgebase. This knowledgebase will be piloted as part of the continuing work in this area. Early workshops demonstrating this approach to process support have met with a favourable response from potential users.

Use Case 7: Execute service provisioning planning to support a business process with traceability between business process requirements and service implementations.

Visualisation of an end-to-end business process, partitioning that process into appropriate sub-processes and then reviewing the service provisioning requirements for each sub-process are stages that are beyond the scope of this paper. Evidence from the early stage of the process synthesis and componentization of the information model did indicate that a candidate set of services may be proposed and the subsequent stages of design and development planned. Mapping between the service specifications and the activities in the process will need to be maintained throughout the final application's lifecycle in order to manage traceability.

7 CONCLUSION AND FURTHER WORK

Employing a synthesis pattern allowed the aggregation of multiple detailed inputs from a set of individual process and information models. This synthesis leads to a canonical model that, based on the results obtained to date, appears to make sense for all the participating institutions. The synthesis rules developed in the project provide an understanding of the points at which extensions (options for different institutions) are needed. Thus the commonality / variability tension facing software design is resolvable in a specific way, where common features, variable features and their interrelations are presented in candidate reference model. Once the canonical model is complete it will be tested for an additional HEI that did not participate in its construction.

The canonical reference model could be used directly by an institution by selecting the most appropriate options at each of the extension points. It could also be used as a starting point for an institution wishing to develop its own model and applications to support course validation. The discussion and presentation of a visual model of the course validation process proved to be useful as a quality control mechanism on the process itself with one institution discovering a step missing from its own documentation. It was also welcomed by all four institutions as a means of disseminating procedure and explaining the rationale of the process to all stakeholders involved. Evaluation of the reference model against a set of proposed Use Cases showed that the method can produce an operational reference model. Furthermore the approach adopted here shows potential for generalisation of the reference model for any domain. Further work will be carried out to refine the method and architecture for a reference model to support the provision of services to execute any well-defined business process.

ACKNOWLEDGEMENT

This work has been supported by funding from JISC – Joint Information Systems Committee, e-Learning Programme, http://www.jisc.ac.uk/index.cfm?name=elfref_tvu

REFERENCES

- Advanced Distributed Learning (ADL) (2004), Sharable Content Object Reference Model (SCORM) http://www.adlnet.org/scorm/index.cfm
- Barn, B., Dexter H., Oussena, S. and Sparks, D. (2006) SOA-MDK: Towards a Method Development Kit for Service Oriented System Development, *Proceedings of the 15th International Conference on Information Systems Development: Methods and Tools, Theory and Practice*, Budapest, Hungary
- Barros, O. (2004), Business Information System Design Based on Process Patterns and Frameworks, BP Trends http://www.bptrends.com/resources_publications.cfm?publicationtypeID=DFC61D66-1031-D522-3EBDAB1F65A451AA
- Barros, O. (2005), Business Process Patterns Project http://www.obarros.cl/businessprocesspatterns.html
- Blinco, K. Mason, J. McLean, N. and Wilson, S. (2004) *Trends and Issues in E-Learning Infrastructure Development.*. http://www.jisc.ac.uk/uploaded_documents/Altilab04-infrastructureV2.pdf
- Chen, P. (1976) The entity-relationship model toward a unified view of data. In: ACM Transactions on Database Systems 1 no. 1, pp. 9–36
- Consultative Committee for Space Data Systems (CCSDS) (2002): Reference Model for an Open Archival Information System (OAIS).

http://ssdoo.gsfc.nasa.gov/nost/wwwclassic/documents/pdf/CCSDS-650.0-B-1.pdf

- Dexter, H. and Petch, J. (2006) "Building a Process Driven Knowledgebase for Managing Student Support", Proceedings of the Information Resources Management Association International Conference, Distance Learning Track,, Washington, D.C., USA.
- Dobing, B. and Parsons, J. (2000) Understanding the role of Use Cases in UML: A review and research agenda. *Journal of Database Management* 11, 4, 28–36
- Dobing, B. and Parsons, J. (2000) How UML is used. *Communications of the ACM*, May 2006. Vol.49, No. 5.
- Durusau P., and. Newcomb, S. (2005) Topic Maps Reference Model, http://www.isotopicmaps.org/TMRM/TMRM-latest-clean.html
- Eriksson, H. and Penker, M. (2000): Business Modelling with UML: Business Patterns at Work, Wiley and Sons,
- Esswein W., Zumpe DKS., Sunke N. (2004) Identifying the Quality of Ecommerce Reference Models, Proceedings of the 6th international conference on Electronic Commerce, Delft, The Netherlands
- Fettke, P., Loos, P and Zwicker J. (2006) Business Process Reference Models: Survey and Classification, Lecture Notes in Computer Science, Springer-Verlag

Federal Information Processing Standards Publications (FIPS) (1993) Integration definition for function modeling (idef0). Software Standard. Modelling techniques.FIPSPub183, Computer Systems Laboratory National Institute of Standards and Technology, Gaithersburg, Md. 20899, USA Frankel, D (2005) Business Process Trends.

http://www.bptrends.com/publicationfiles/07%2D05%20COL%20BP%20Platform%20%2D%20Frankel%2Epdf

- Fremantle, P; Weerawarana, S. and Khalaf, R.. (2002). Enterprise Services. *Communications of the ACM* Vol 45 (10).
- IBM-Rational, (2004) The Rational Unified Process (RUP), http://www-306.ibm.com/software/awdtools/rup/
- Jacobsen, I., Griss, M., and Jonsson, P. (1997) "Software Reuse: Architecture, Process and Organisationn for Business Success" Addison-Wesley, USA.
- Joint Informations Systems Committee (JISC) (2005) the e-Learning Framework http://www.elframework.org/

Leishman, D.A (1999) Solution Customisation. IBM Systems Journal, Vol 38, No. 1.

- Nordheim, S (2006) Towards a dialectic understanding of Enterprise Systems Vendor challenges and contradictory rhetoric. *Information Systems Development Conference*, Budapest 2006. (Springer Lecture Series).
- OASIS (2005) SOA Reference Model. http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=soa-rm
- Object Management Group (OMG) (2005) The Unified Modelling Language (UML) http://www.omg.org/technology/documents/formal/uml.htm
- Object Management Group (OMG) (2004) A UML Profile for Enterprise Distributed Object Computing. http://www.omg.org/technology/documents/formal/edoc.htm
- Object Management Group (OMG) (2004) Common Object Request Broker Architecture (CORBA) http://www.omg.org/technology/documents/formal/corba_iiop.htm
- Ort, E (2005) Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools. http://java.sun.com
 - /developer/technicalArticles/WebServices/soa2/
- Oussena, S., Sparks, D. and Barn, B. (2006), in draft, Implementing Reference Models using Service-Oriented Architecture and XML Rules, http://covarm.tvu.ac.uk/covarm/papers.jsp
- Themistocleous, M., Irani, Z., O'Keefe, R.M. (2001), "ERP and application integration", *Business Process Management Journal*, Vol. 7 No.3, pp.195-204.
- Thomas, O (2006) Understanding the Term Reference Model in Information Systems Research: History, Literature Analysis and Explanation, *Lecture Notes in Computer Science*, Springer-Verlag
- Wand, Y., Weber, R.: Research Commentary: Information Systems and Conceptual Modelling A Research Agenda. Information Systems Research 13 (2002) 4, 363-377