Synthesis of Multiple Process Models for Creating a Rich Enterprise Application Reference Model for Service Oriented Architectures

Balbir S. Barn[1], Hilary Dexter[2], Samia Oussena[1], Jim Petch[2]

[1]Thames Valley University
Institute for Information Technology
Wellington Street,
Slough, SL1 1YG, Berkshire, UK

[2]University of Manchester,
The e-Learning Research Centre
186 Waterloo Place, Oxford Road
Manchester M13 9PL, UK

bbarn@acm.org
hilary.dexter@manchester.ac.uk
samia.oussena@tvu.ac.uk
jim.petch@manchester.ac.uk

Abstract: Improving business processes and services in Higher Education Institutes (HEI's) 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. As part of a national programme for developing such models, a canonical reference model for course validation was used as a case to demonstrate the feasibility of the approach. Course validation processes in four UK HEIs were analysed and modelled based on 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 synthesised into a single canonical model of the validation process. This required resolving process model structures and granularity. Synthesis of the canonical model demonstrated a basis for developing reusable process patterns. The canonical reference model appears to be valid for each institution but requires fuller testing.

1. Background

Enterprise Information systems design and implementation has seen an evolution from centralised mainframes to bespoke application development using client-server 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 [Ort05]. A separate move towards a stronger focus on application integration led by business process modelling is being enabled by the developing SOA principles. There is a strong emphasis on driving this move from a solid understanding of business processes and aligning developed or procured services to support those processes [Fra05, Bl04].

One way to provide services in full alignment with business processes is to adopt a formal model driven development process that can link the business processes to the sets of 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 can become a reference model used to direct the provision of services.

The research question addressed in this paper is whether it is possible to combine versions of models of a particular business process from different enterprises in the same sector to produce a useful, single, flexible, customisable canonical process that can become a reference model for service provisioning across the sector. The paper discusses an approach to providing a flexible, customisable 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, synthesis of process models from multiple institutions and establishing the system model required for component based development for an SOA.

1.1 The Problem

Service Oriented Architecture is both confusing and challenging. It is confusing because there there are a myriad terms that seem to lead to the same basic notions – Grid Computing, Web Services Architecture, Service Component Architecture to name but a few. It is challenging because it is the direction that the IT industry is inevitably heading towards led by the vision of the virtual enterprise and this is still 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, for example the Web Services protocol, SOAP and interface specification language, WSDL, currently available [FWK02].

Achieving a sufficient level of process and information integration is the main challenge facing the industry and is what is now being steadily addressed by new developments in service oriented architecture. Given the current confusion surrounding SOA, what is needed is a clearly articulated standard framework that addresses issues of terminology, functional requirements; information integration requirements and methodology / techniques requirements. Part of such a framework is the notion of reference models. This has been recognized and a new initiative led by key industry players has been announced [Oas05] to develop a SOA reference model.

Unfortunately, reference models themselves are the product of interpretation and of modelling preferences.

1.2 Reference Model Discussion

Recent activity in the development of so-called reference models means that it is instructive in the HE sector to enter a discussion on reference models and state what we mean by a reference model. We are interested in pursuing this because we believe that there is value in using reference models in taking a model driven approach to application development so that a business process model can become a reference model to guide the provision and use of services.

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 organization 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 [OMG03]; CORBA 2.0 (Common Object Request Broker Architecture) Specification [OMG99] and the EDOC (Enterprise Distributed Objects) specification [OMG01].

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 synthesise a definition of a reference model – "a proof by induction". 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

Iinspection of the reference models points to emerging common themes. 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 standardisation. 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.

Aspects of the reference model concept may be confused with other related concepts and there is a need for clarification. As we have stated earlier, the reference models provide a means to agree on definitions of terms and their relationships. This appears to have some similarity to notions of "Domain Models" and "Ontologies".

Ontology is generally regarded as the specification of an abstract view of some problem domain in the real world. Ontology will typically furnish concepts, definitions and the structural relationships between concepts (Gr95, HJ02). Taken as whole, the ontology captures the business rules of the problem domain. Hence the purpose of producing an ontology is to provide a common language for sharing and re-using knowledge within the problem domain between different parties within the framework of the captured business rules [NO99]. More specifically, the uses of ontology (and also Reference Models) are characterised as being [GL02]:

1. For communication between implemented systems; between humans and humans and systems.
2. For computational inference i.e. internal representation and manipulation of information for planning; also for analysing the internal structures, algorithms and input/outputs of systems for conceptual understanding.
3. For reuse and organisation of knowledge.

This of course is not new. Business Analysts have been constructing ontologies whenever they have performed analysis when designing information systems. They have called them simply "Domain Models" or "Conceptual Models". Often a core deliverable of an IT strategy for an enterprise will be the production of a "Corporate Information Model". This suggests that there are relationships and overlap between ontology / domain model and reference models [Ba03].

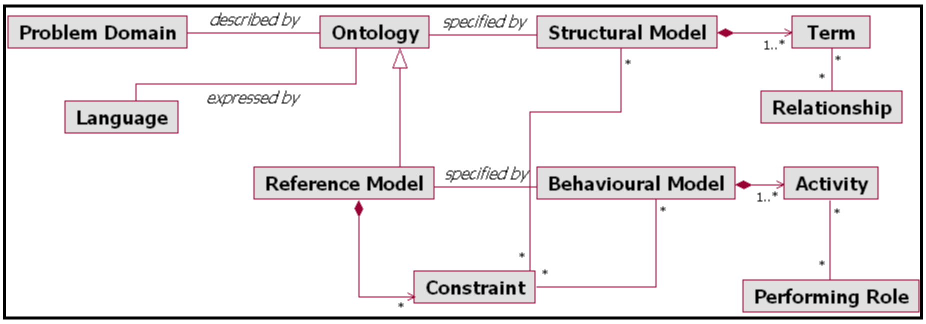

Figure 1 Conceptual Model of Reference Models and Ontology

In the above conceptual model (Figure1), an ontology has only a structural model i.e. it defines a set of terms and the relationships between the terms. A reference model however has additional specification characteristics such as behaviour description and a description of the constraints / rules that may apply between a behavior model and a structural model. An example of a constraint may be one that states that only certain concepts are appropriate when a process is in a particular state. We conclude this discussion by formalising our notion of a reference model as:

> 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 to a non-specialist and can be viewed as a framework for comparing architectures and operations of existing and future systems.

This statement and its meaning is contained in the diagram (Figure1) above. At an abstract level, the Behaviour Model is a description of the interfaces and their constraints. The Activity term can be used to represent both a high-level business process, an activity within the process or even a web service implementing a small, discrete piece of functionality. Roles can be generalised to mean both the entity performing a specific activity and the location of a web service. The domain specific terms are represented by the ontology aspects of the reference model. The task then, is to develop a reference model that contains the elements described above for a specific problem domain. This is described in the remainder of the paper.

2. Case Study

The e-Learning Framework (ELF) [1],[2] 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. Within this initiative, JISC has requested projects to develop reference models for a number of domain areas. Domains have been defined by CETIS (Centre ofr 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 (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:

---

[1] http://www.elframework.org/

[2] http://www.jisc.ac.uk/elearning_framework.html

> 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 one of the most 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.

2.1 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 programme 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 that impose a significant overhead on the developmental process for validating new courses. These wider requirements are determined by the national bodies such as the QCA[3], QAA[4], HESA[5] and UCAS [6]. These bodies collectively ensure that courses are designed and validated to the required level of quality standards.

Course validation is further complicated by the consideration of the modes of deployment of new courses e.g. based on learning technologies or on traditional modes of delivery to support students, and of 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. It is not clear, nor is there any level of standardization, on 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 constraints imposed by external bodiies such as the 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] http://www.qca.org.uk/

[4] http://www.qaa.ac.uk/

[5] http://www.hesa.ac.uk/]

[6] http://www.ucas.ac.uk/

2.2 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, typically 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 also comes 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, alignment of internal supporting business processes will be possible.

3. Methodology

Our methodology is derived from elements of RUP (Rational Unified Pocess) [Ib05] and is strongly tailored to the delivery of our main artefact, a reference model for course validation. The artefact will contain the characteristics described in section 1.2 and conform to the definition stated earlier. The methodology takes a model based approach using UML; is an application and adaptation of software engineering and is iterative. While the methodology addresses domain modeling, service interface specification, service implementation through to a test execution of the business process, this paper will focus only on the domain analysis stage. This entails a study of the both dynamic (process) and structural (data) information. Thus the 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 following diagram (Figure2) shows the process followed by the team to gather the required information in order to build a model of course validation for each institution. The interviews were all conducted by two members of the project team together. The models were built using UML 1.5 with IBM-Rational XDE modelling tool.

It should be noted that the enterprise modelling community does not uniformly employ UML for modelling business processes. The Business Process Modelling Notation (BPMN) is, however, considered to be relatively simple and intuitive for business people to describe their processes and may therefore make dissemination of the reference models more successful. The employment of both UML and BPMN notations need not be an issue since the involved standards organisations, BPMI (Business Process Management Initiative) and OMG, are nearing convergence of the

BPMN with UML via a common metamodel, the Business Process Definition Metamodel [Bp04]

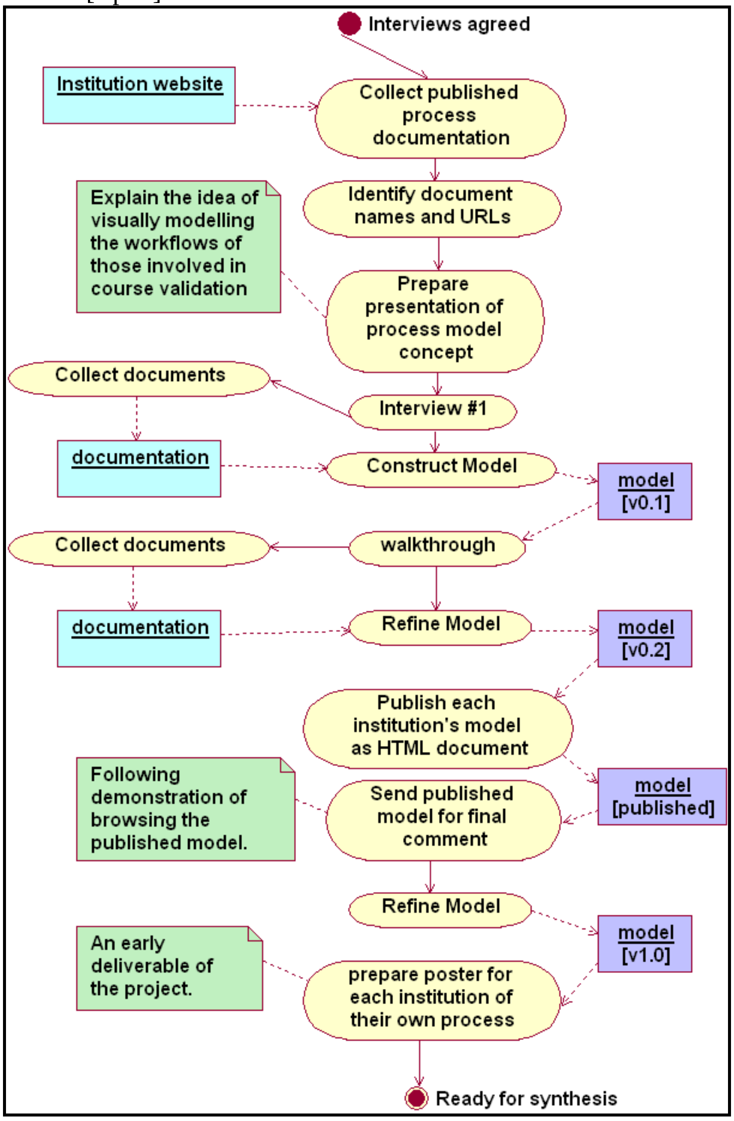

Figure 2: The Domain Modelling Process

3.1 Modelling the process in each institution

Each institution's course validation process was modeled 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 collaboration of people, that could be seen to produce a defined output for course validation. 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 thus captured in the activity diagram. The existence of guidelines or checklists for the execution of activities was also documented and placed in the object flow of the diagram.

The description of each activity included the roles involved. 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 people for committee meetings.

3.2 Creating the domain information models

Alongside the Activity Diagram of each institution's business process, a UML Class Diagram was created 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 included. An example is given in Figure 4 of part of one of the domain information models that illustrates the main kinds of elements and the level of abstraction.

3.3 Synthesising the process models

Following the two iterations of interview and model refinement in each of the four institutions, a process model and domain information model of each one 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 to the interviews and modelling activities, but it was necessary to review all four models alongside each other 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. In using the characteristics to produce the synthesis the following rules were employed. The synthesised model represented an aggregation of concepts – that is, we did not employ any "re-engineering" to optimise the process. Variations between HEI processes were captured as extension points from the canonical process. Transformation relationships would be used to store the details of the variations. The following conceptual model (Figure3) 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 customisation for any HEI.

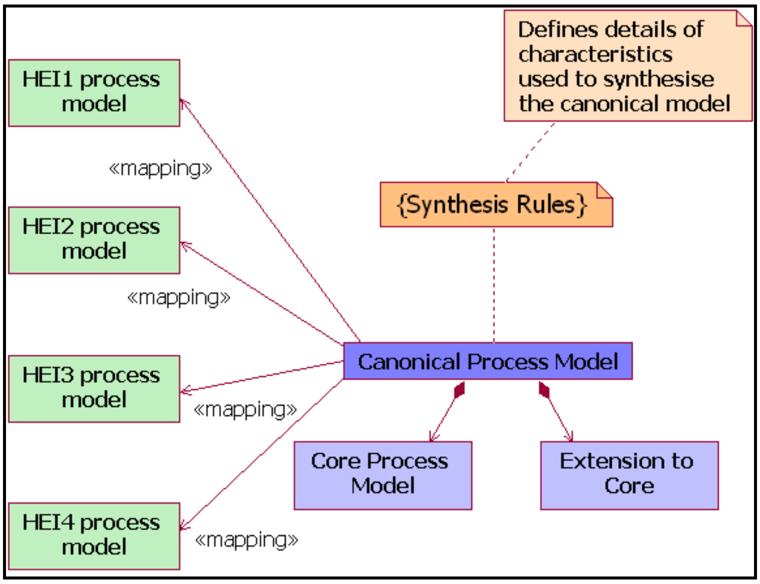

Figure 3 A Pattern for Synthesising the Canonical Model

Table 1 Characteristics for Model Synthesis

Characteristic	**Description**
Stage	The principal stages of the process i.e. those sub-processes that made up the end-to-end process for course validation e.g. 'developing the business case'
Role	Primary responsibility for an activity. Roles could be individuals, teams, organisational units e.g. Dean, development team, quality unit
Object Flow	Any items used or produced by the activities in the process. These were usually documents.
Object Lifecycle	Key documents passed through a number of states as they were written and reviewed. These states form the lifecycle of the document e.g. 'course proposal'
Events	An event was taken to be something that triggered a sequence of activities e.g. approval or rejection of a proposal.
Collaborations	These were activities executed by two or more roles such as a committee.
Activities	Executed by roles and collaborations. Activities were defined as having to produce some substantive change in the state of the system e.g. a new section in a document completed, a cost analysis completed, a document approved.

Characteristic	Description
Constraints	Business rules that determined who could execute an action, when things had to be completed, compliance to standards etc.
e-Learning Activities	Specific activities in the process relating to validation of e-learning sections of the course being proposed.
Reference Document	Documents used to inform the roles creating and evaluating the course proposal e.g. strategy documents
Interfaces	The places in the process where a role interfaces with other enterprise information systems to accesses or provide information.

3.4 Preparing for design and development

Following the synthesis of the four process models into a single canonical model the following stages will be executed:

- define technology independent specifications of services;
- define XML data representations of the information consumed and produced by services;
- generate appropriate implementation models of the service specifications.

The software design and development for supporting course validation will follow the methodology for component based development originally developed at Texas Instruments [BCB98, CD01] and the results will be discussed in a future paper.

4. Results

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

4.1 Information Model Sample

The following diagram (Figure4) 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.

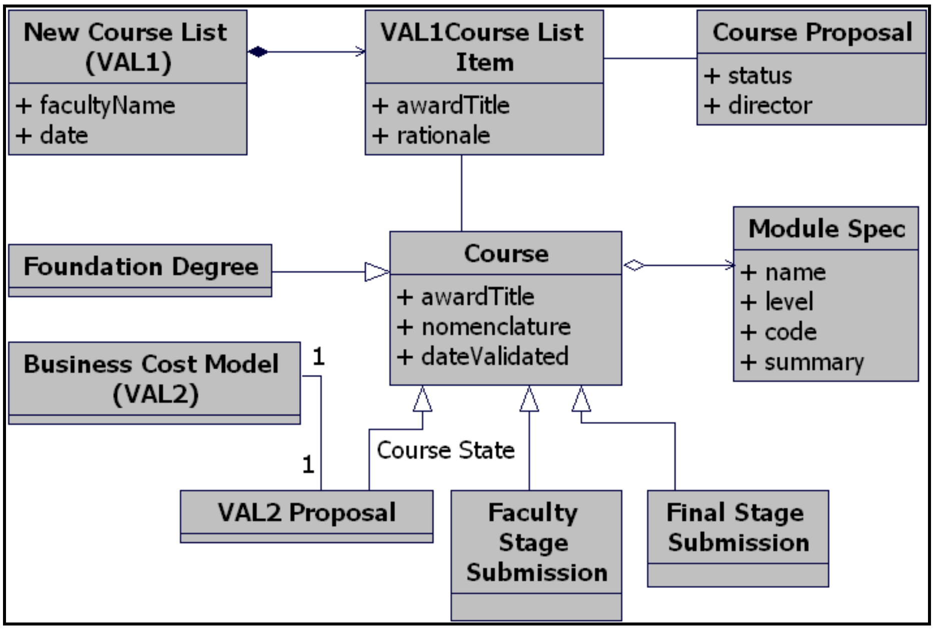

Figure 4 Sample from one of the four domain information models

4.2 Process Model Sample

The following diagram (Figure5) 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 pink for documents, was used as an aid to communication of the various aspects of the process model: roles, activity sequence, object flow, decisions and constraints.

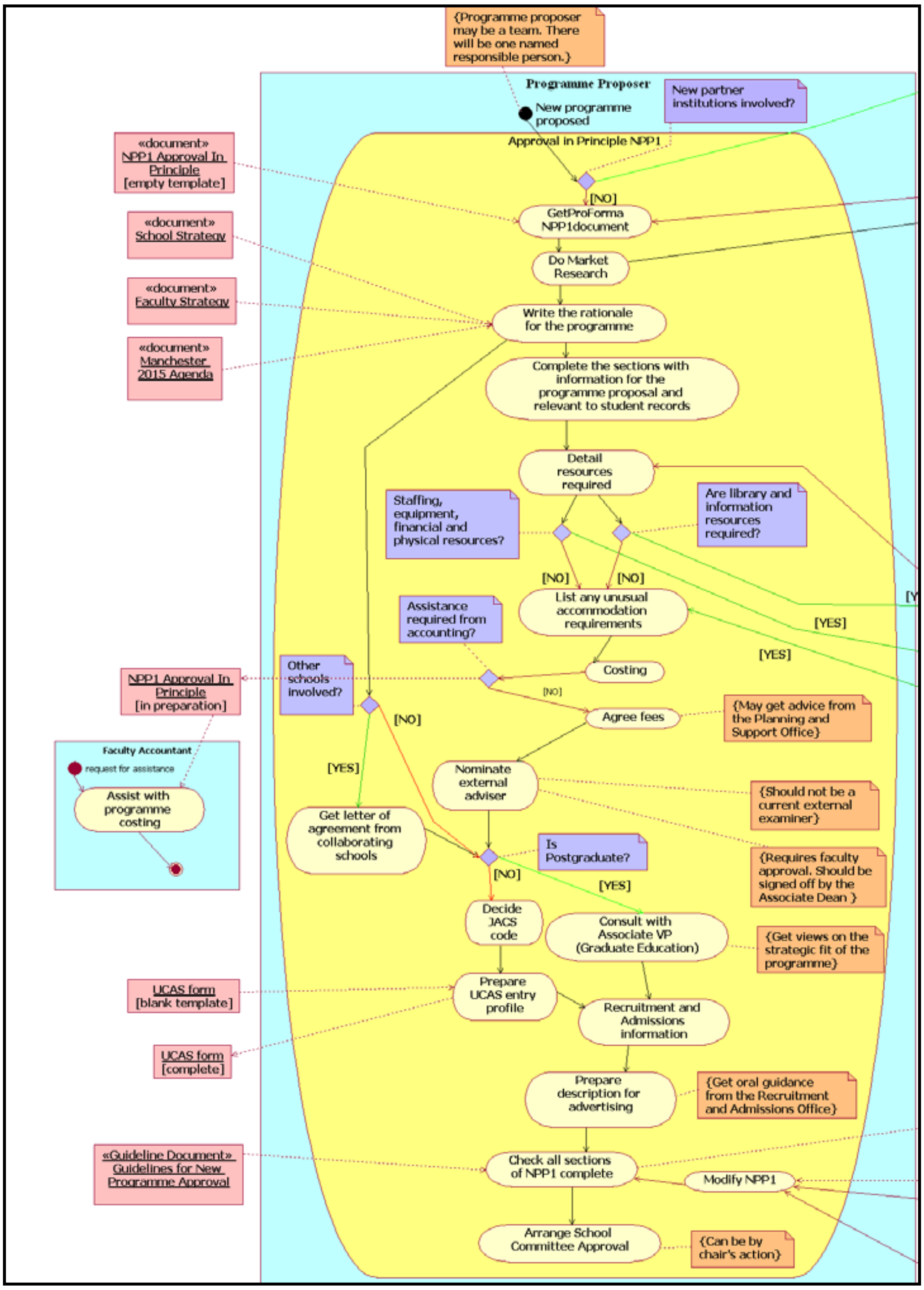

Figure 5 A section from a Course Validation process

4.3 Model Synthesis Results

Using the model synthesis pattern, described in section 3.3 and illustrated in Figure 3, the first iteration in the process of merging the four individual models into a single canonical model was carried out. Each characteristic as described in Table 1 was taken separately and the pattern realization constructed. The following (Figure6) is the result for a single realization of the synthesis pattern, for the characteristic 'stage', and is given as an example of the method:

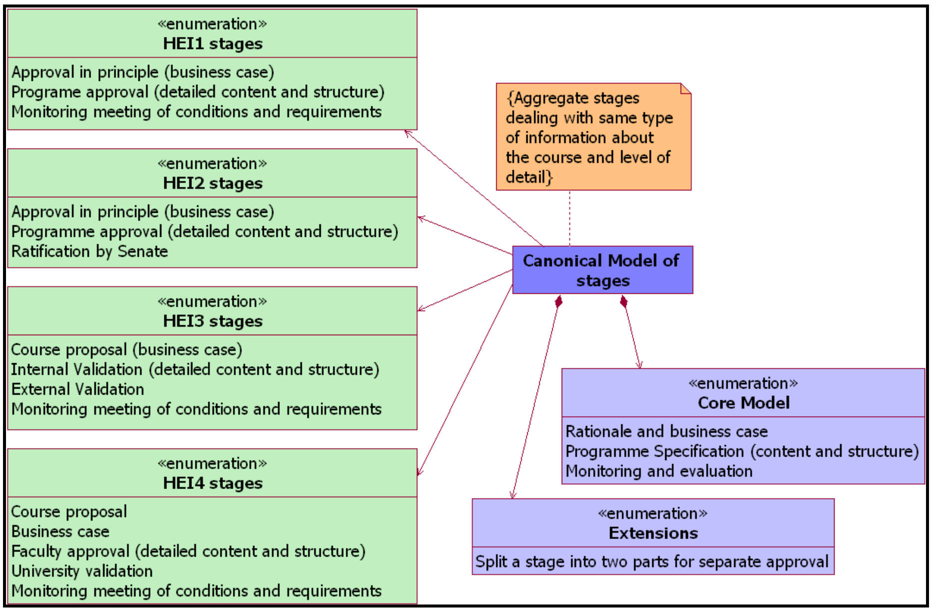

Figure 6 Synthesis of process stages

4.4 Reusable Process patterns

Process patterns follow a similar thrust to 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 [Barr05, Barr04, and Am05]. Barros describes process patterns at a macro level where he argues that the business of enterprise may be represented by 4 Macro patterns. These are described using the IDEF0 notation and may be specialised by at the domain level. Ambler's patterns [Am05] are primarily focused on the software development process.
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 could improve the 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 have the potential for generalisation. This will be the subject of future research.

5. Conclusion

Employing a synthesis pattern allowed the aggregation of multiple detailed inputs from the four individual process and information models. This synthesis leads to a canonical model that, from the results obtained to date, appears to make sense for all the participating institutions. The synthesis rules applied also lead to an understanding of the points at which extensions (options for different institutions) are needed. Once the canonical model is complete it will be tested for an additional HEI that did not participate in its construction.

Terminology differences between institutions emerged as an important issue and we have planned a pilot study, for the near future, with a tool that should permit models to be developed in a manner so that both shared and local meanings can be maintained. The aim will be to produce a transformation model to allow the canonical course validation process model to be expressed in the required local terminology. The results of this pilot will be published in a future paper.

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 parties involved.

References

[Am05] Ambler,S (2005) Process Patterns Resource Page http://www.ambysoft.com /processPatternsPage.html

[Ba03] Barn, B. S. (2003) "Meta Models for Knowledge Management Reference Architectures". Proceedings of KMAC03.

[Bar04] Barros, 0 (2004) O. Barros, Busines Information System Design Based on Process Patterns andFrameworks. *BPTrends* (http://www.bptrends.com *)*, Sept. (2004)

[Bar05] Barros, O (2005), Business Process Patterns Project http://www.obarros.cl/businessprocesspatterns.html

[BCB98] Barn B.S; Cheesman J and Brown A.W; (1998) "Tools and Methods for Component Based Development", with A.W. Brown and John Cheesman. Proceedings of Tools-26, Santa Barbara, CA, USA, Aug 98.

[BHL01] Berners-Lee, T; Hendler, J., and Lassilo, O. (2001) The Semantic Web. Scientific American, May.

[Bl04] Blinco, K. Mason, J. McLean, N. and Wilson, S. (2004) Trends and Issues in E-Learning Infrastructure Development. A White Paper for the Alt-I-Lab 2004 conference. http://www.jisc.ac.uk/uploaded_documents/Altilab04-infrastructureV2.pdf

[Bp04] Revised Submission to Business Process Definition Metamodel (BPD) v1.0.2 http://www.bpmn.org/Documents/BPDM/OMG-BPD-2004-01-12-Revision.pdf

[CD01] Cheesman, J, Daniels, J. (2001) *UML Components*; Addison-Wesley.

[DW98] Desmond F. D'Souza, Alan C. Wills (1998) *Objects, Components, And Frameworks with UML – the Catalysis Approach*; Addison-Wesley, Reading, Mass.

[EM04] Easterby-Smith, S and Mckenna, R (2004), Specification for Developing E-Learning Framework (ELF Reference Models V1. JISC

[FWK02] Fremantle, P; Weerawarana, S; Khalaf, Rania. (2002). Enterprise Services. CACM Vol 45 (10).

[Fo02] Fowler M (2000) *UML Distilled*, 2nd Edition, Addison Wesley.

[GL02] Gruninger, M, Lee, J. Ontology Applications and Design (2002) in *Comms. of ACM.* 45, 2, Feb.

[Gr95] Gruber, T.R. "Towards principles for the design of ontologies used for Knowledge Sharing"; *Int. J. Hum. Comput. Stud.* 43, 5/6 (1995).

[HJ02] Holsapple, C.W and Joshi, K.D (2002) A Collaborative approach to Ontology Design. In *Comms of ACM*. 45, 2, Feb.

[Ki02] Kim, H. (2002) Predicting How Ontologies for the Semantic Web will evolve. Comms of ACM. 45, 2, Feb.

[Me97] Meyer B (1997) *Object-Oriented Software Construction;* 2nd Edition, Prentice Hall

[Mi97] Microsoft; (1997) Open Information Model Overview for the Microsoft Repository.

[NO99] Natvig, M.K; Ohren, O (1999) Modeling Shared Information Spaces (SIS). *Conf. Group 99*; ACM.

[OMG99] OMG TC Document, orbos/99-02-05 (1999) *CORBA Components*.

[OMG01] OMG Document, ad/2001-02-19 (2001) *A UML Profile for Enterprise Distributed Object Computing*. Feb.

[OMG03] The Unified Modeling Language; *Object Management Group*. http://www.omg.org/technology/documents/formal/uml.htm

[Ort05] Ort, E (2005) Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools. http://java.sun.com/developer/technicalArticles/WebServices/soa2/

[Fra05] Frankel, D (2005) Business Process Trends. *BPTrends* http://www.bptrends.com/publicationfiles/07%2D05%20COL%20BP%20Platform%20%2D%20Frankel%2Epdf

[Oas05] Oasis SOA Reference Model. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm

Acknowledgements

This work has been supported by funding from JISC – Joint Information Systems Committee, www.jisc.ac.uk .