Developing an Ontology

Janet Lavery and Cornelia Boldyreff

Department of Computer Science
University of Durham
Science Laboratories, South Road
Durham, DH1 3LE, U.K.
Janet.Lavery@durham.ac.uk
Cornelia.Boldyreff@durham.ac.uk

Abstract

The ever-increasing use of the World Wide Web by staff and students at Higher Education Institutions within the UK has lead to the expectation that a rising number an institution's information based services should be accessible via the web. This entails the migration of existing information services to web based services. In addition, it is expected that work performed for the migration will identify areas ripe for the development of new value-added web based services. The Institutionally Secure Integrated Data Environment (INSIDE) project is a collaborative project between the University of St Andrews in Scotland and the University of Durham in England. The project has been addressing the issues surrounding the development and delivery of web based "joined up systems" for institutions. It is not within the remit of this project to replace any existing systems but instead to work with the existing information bases and to incrementally develop web based services upon them. As the problems and issues that arise are likely to be common to many HEIs, we have sought to identify the issues and to solve the problems at a high enough level of abstraction to give sufficiently generic solutions applicable to other HEIs. To better understand the requirements for an integrated web based service, a business process common to all UK Higher Education Institutions, the registration of new students, has been analysed and modelled at both institutions. This process was chosen because it is practised, in some form, in all UK Higher Education Institutions and because the student data it captures is shared with a variety of systems in academic and non-academic departments. The analysis and modelling of a common business process has provided insights into the Meta-Processes of requirements engineering, such as understanding the domain vocabulary, the existing business processes, and their associated information bases and legacy systems. In particular, through trial and refinement, the project has developed an approach to analysis and modelling using some of the UML notations. Via this approach, the development of a common abstracted vocabulary that began as a simple dictionary of domain terms has evolved into the basis for developing an ontology. This paper outlines the meta-process of the approach we eventually developed and used successfully in what might be termed "brown field site" system requirements engineering.

Keywords:

Domain Analysis, Meta-Process, Ontology, Requirements Engineering, UML

Background

"Throughout the UK there are thousands of sites which have been contaminated by previous industrial use, often associated with traditional processes which are now obsolete, which may present a hazard to the general environment, but for which there is a growing requirement for reclamation and redevelopment." This quote was taken from the UK Government Environment Agency web site [1] and refers to land based brownfield sites. It could also refer to the challenges facing the software engineering industry today in transforming legacy systems with their dated software, distributed data, and entrenched business processes into useful, web accessible systems. Unlike the derelict land brownfield sites chosen for reclamation and redevelopment, software brownfield sites are usually functioning systems supporting an ongoing institution or business in its continued existence while not fully supporting or adapting to the changing needs of their user communities. The reclamation and redevelopment of software brownfield sites requires a multi-layered understanding of the domain in which the enterprise system lives supported by a modelling approach that provides models of the domain at varying levels of generalisation throughout the system evolution process.

Many Higher Education Institution (HEI) systems in operation are comprised of multiple unconnected data repositories, distribute over several sites. Users are often prevented from carrying out work by inappropriate access control mechanisms and the lack of appropriate client software. In an effort to cope with the difficulties numerous ad hoc record systems have been developed at the departmental (academic and administrative) level within the institutions. These systems then replicate work being carried out both centrally and in other departments. The data manipulations are not co-ordinated with each other or central services consequently information exchanges between the centre and rest of the institution are burdened with inconsistent data. In addition, lifelong learning initiatives imply that HEIs can no longer operated in isolation. Lifelong learning initiatives have persuaded organisations of the need to find the means to enable them to exchange learning objects, anything from student records to bench tests, in a variety of formats that can be found in Managed Learning Environments.

The Institutionally Secure Integrated Data Environment (INSIDE) project is a collaborative project between the Universities of St Andrew and Durham that has been addressing the issues surrounding the development and delivery of web based "joined up systems" for institutions. It is not within the remit of the project to replace the existing systems but instead to work with the existing information bases and to incrementally develop services upon them. As these problems are common to many HEIs, we have sought to identify the issues and to solve the problems at a high enough level of abstraction to give sufficiently generic solutions applicable to other HEIs.

An essential aspect of our work is to provide generic solutions; we are endeavouring to develop of a generic model of the domain knowledge pertaining to key HEI business processes. This commenced with the modelling of a single complex process common to all HEIs in the UK, the registration of new undergraduate students. The intention of this process is to register students intending to meet a specific academic target such as gaining a Bachelor of Arts degree with an HEI. The process of registering new full-time undergraduate students begins the same for all UK Institutions when the HE student records for the new academic year entry cohort are distributed from a central "clearinghouse", the Universities and Colleges Admissions Service (UCAS) [2]. UCAS distributes subsets of student records to the central registration service (admissions department) of the HEIs. Each central registration service then distributes the appropriate student records to academic and non-academic departments involved in the institution's registration process. When in the custody of UCAS, the student records have an identical structure and content base. Once in the custody of the HEIs the student records are manipulated to reconcile their content and structure with the needs of a particular institution. Additional manipulation may also occur to suit the specific needs of the academic and non-academic departments within an institution.

Initially an informal model of each individual HEI's registration process was assembled. The two models have been compared with the intention of identifying the commonalities and discrepancy on which to base a generic model of the process and to begin the accumulation of knowledge about the domain. This was considered a necessary first step in the development of the generic model from which sets of core requirements for the registration processes were to be gleaned. As the registration process begins the same for all UK Institutions it is believed that the resulting generic model and other work products may usefully provide the core domain analysis necessary for requirements gathering in the brownfield site of undergraduate registration systems.

In this paper we define the meta-process incorporating UML work products that has emerged as a way to support incremental implementation of value-added services in context of brownfield site systems. This meta-process has as its foundation the well-established domain analysis principles defined by G. Arango, R. Prieto-Díaz and others over the last few decades [3]. This account of the meta-process includes identification of the relationships that exist amongst the generated work products in conjunction with the meta-process specifically the evolution that leads to the initial development of a domain specific ontology to support domain knowledge reuse during requirements and design. The ontology is a key work product within the generic model. Section 2 provides the details of the meta-process, an overview of the registration process and the associated generic registration model. Section 3 concentrates on the progression from the identification domain vocabulary problems to the development of a domain specific ontology. Section 4 discusses the open issues and future work.

2. The Meta-Process

To accurately depict the complexities of an enterprise system in a model, it is necessary that the model exploit the generation and evolution of several work products, some of which may contain many interrelated diagrams [4]. For an enterprise model to be useful throughout the development process, it needs to be made-up of multiple interconnected work products. These work products are used to support and enhance the capture of domain knowledge in conjunction with the evolution of the existing enterprise's processes and the introduction of new value-added services. Each work product's evolution needs to be performed in conjunction with the other work products. In addition, each work product is expanded with the additional domain knowledge gained with the implementation of each new value-added service. Figure 1 below depicts the iterative meta-process that has emerged in our efforts to develop an enterprise system model to support incremental evolution and implementation of value-added services.

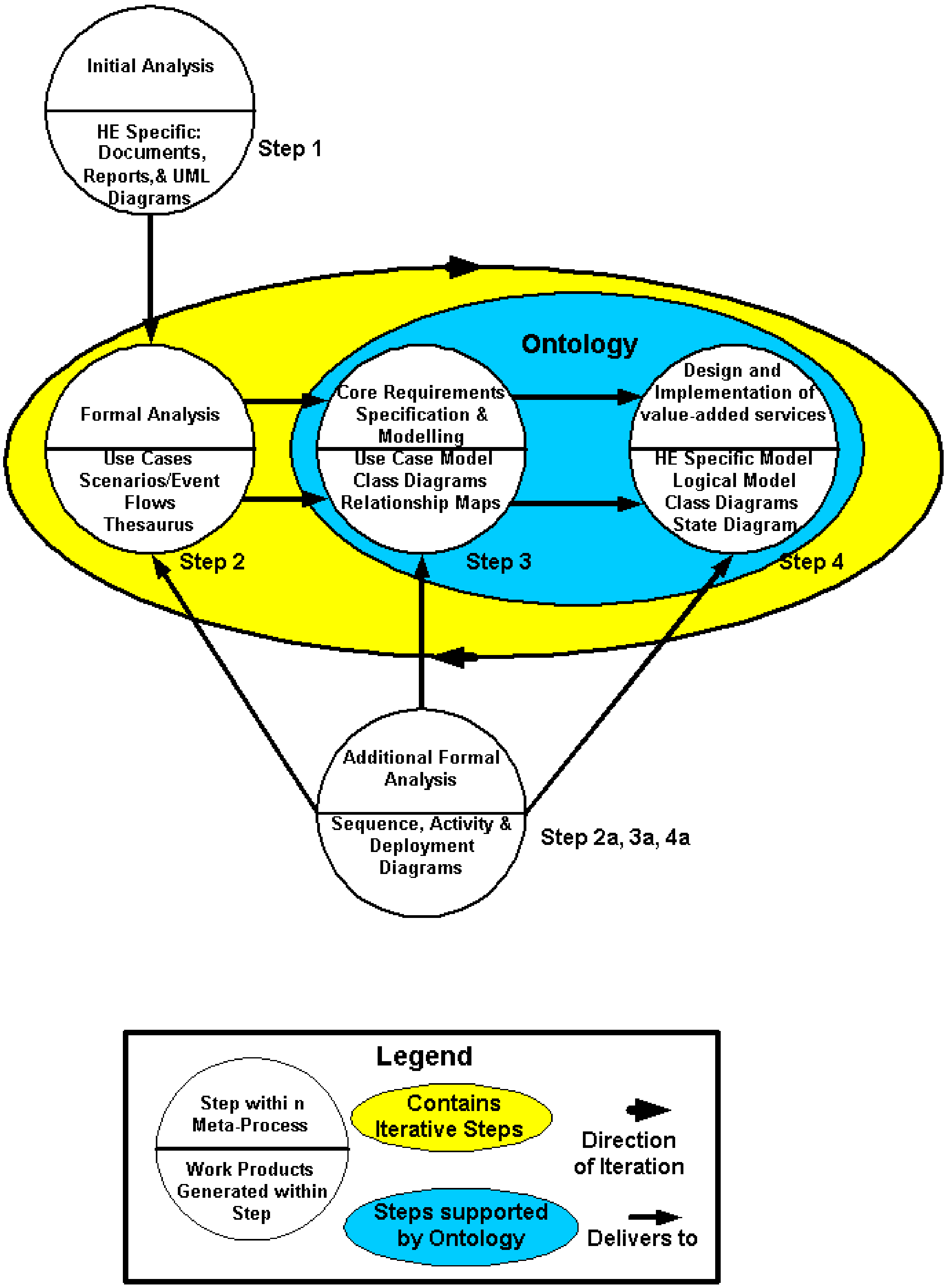

Figure 1 The Meta-Process

In this emergent meta-process, Step 1 Initial Analysis is where the essential work of locating the domain knowledge sources and defining the current domain boundary [5] is performed. Work products are generated to document the informal analysis and capture the first pass of the domain knowledge within a specific area of the domain such as a single business process.

These work products provide the initial input for the work products generated in Formal Analysis and remain fixed from the point of their input into Formal Analysis. The remaining sequence of steps in the meta-process are mutually dependent and performed iteratively. The emphasis is on the capture and modelling of domain knowledge while evolving an existing enterprise system. The cycle is based on the necessity to expand the domain knowledge while performing incremental development of value-added services to specific areas of the existing systems. In Step 2 work products generated in Formal Analysis provide the knowledge base for work products generated during Core Requirements Specification and Modelling, these in turn provide the knowledge base for the work products generated during Design and Implementation. As new value-added services are implemented the foundation on which the previously developed Formal Analysis work products were developed is altered. As a consequence Design and Implementation work products provide the initial source for the next iteration of the Formal Analysis work products. The necessary evolution of the enterprise and its systems will also contribute to the evolution of the model and its associated work products. In Additional Formal Analysis work products are used as input into Steps 2 through 4 as required. They are generated to support analysis with a specific focus; for example, sequence diagrams are used to support the analysis of concurrent processes, an activity that occurs during Core Requirements Specification and Modelling.

In the following sub-sections the Meta-Processes steps are described with reference to the evolution of the HEI generic model that began with modelling the undergraduate registration process.

Initial Analysis

In Initial Analysis, work products are generated to document the informal analysis and capture the first pass of the domain knowledge held in a narrow area of the enterprise domain. For INSIDE the focus of the Initial Analysis was on the individual HEI's central registration activities and the related activities performed within a single academic department within the HEI. Analysts at both institutions based their initial domain analysis process on informal interviews with members of staff (both academic and non-academic) with direct responsibilities relating to the undergraduate registration. Any available HEI documents unique to the undergraduate registration process were also reviewed. The knowledge gained from the informal interviews, and to a lesser extent the existing documentation, was recorded in basic block diagrams. These highly recognisable informal diagrams use analyst-defined boxes, ovals and arrows to represent an understanding of the process. This type of diagrammatic representation relies on additional textual or verbal accompaniment to facilitate understanding. There are three main use for this type of diagram: to demonstrate the analysts increasing domain understanding; to generating discussion amongst domain experts; and to ensure the correctness and completeness of a common understanding of the domain area. For this purpose the basic block diagrams proved highly effective. However, this method of diagrammatically representing domain knowledge can be very specific to an enterprise.

The basic block diagrams developed in Initial Analysis captured domain knowledge specific to a department within a single HEI and used a domain vocabulary specific to the HEI. As the intention was to share the domain knowledge between the two different HEIs, modelling support tools for more generic modelling activities were introduced. It was decided that the more formal and generally understood modelling language, the Unified Modelling Language (UML) would be used. UML was selected because it is an object-oriented notation in widespread use in the software development industry and consequently is effective for use when a common understanding between software engineers is required [6].

Additionally, both HEIs support the utilisation of UML. It was intended that the use of UML would: supplement each department's specific knowledge; aid in the identification of the commonality and disparity between the two different HEIs' registration processes; and facilitate understanding between the analysts. The Dictionary of Generic Terms [2] was developed to support the development of the UML diagrams and to hold the agreed upon generic vocabulary. The UML diagrams and the dictionary are best developed in conjunction with each other. This will ensure that terms used in the UML diagrams are defined in the dictionary. In our work this also ensured that terminology used in the UML diagrams developed at either HEI were understandable to analysts at both universities.

Formal Analysis

The focus of Formal Analysis is the capture of domain knowledge that is then utilised in the modelling of the core requirements. As there has been only one pass over an area of the domain prior to the beginning of the iterative cycle the focus of the first few cycles of the meta-process will be in Formal Analysis. All the work products generated during Initial Analysis are fixed at the completion of Initial Analysis. These work products are used to provide a snapshot of a specific enterprise process at a particular time and as initial knowledge sources for work products in Formal Analysis. Work products built in Formal Analysis are developed to support on-going domain analysis and requirements gathering, and accordingly are developed iteratively as domain knowledge increases. They are developed using support tools and are expected to evolve in conjunction with domain knowledge acquisition and subsequent enterprise system evolution.

To ensure consistency, interoperability, and improved communication between the two HEIs, it was necessary to standardise on support tools that were in common use at both HEIs. The INSIDE project selected UML as the main modelling language in the development of a generic model of the undergraduate registration process. A variety of UML tools were considered and Rational Rose 2000 was selected because it provided the means for INSIDE to develop a sensibly partitioned model. It is common practice to partition models into varying degrees of abstraction. For example, software product line family' models are usually partitioned to reveal the commonalities and variants within the product family. Where models are broken down into kernel representations for those elements or features common to the entire product family and optional models for elements or features specific to a particular member or subgroup of the product family [7, 8]. Another frequent reason for partitioning domain models is to support the view that there are particular domain areas, such as accounting or stock control, that are common to a range of different industries [9]. Here the generic problem domain is modelled then the generic domain model is instantiated by the fine-grain details of a specific enterprise. The INSIDE project has been pursuing the later strategy of domain model partitioning but within the confines of HEIs domain areas. Rational Rose supports models that are divided into two main parts identified as the Use Case View and the Logical View. The Use Case View is used to impart the core or high level business model elements that support domain analysis and requirements gathering [10]. This section contains those elements of the model that are generic and potentially usable by a range of HEIs. The Logical View provides a lower level model used during design and implementation [10]. This section contains those elements of the model that are specific to a particular HEI or a particular subsection of an HEI's system.

In Formal Analysis the emphasis is on the generation of use case diagrams accompanied by detailed scenarios. Burstard et. al. [11] suggest that there are four perspectives from which to view scenarios: process, situational, choice, and use. The process perspective places the focus on events and event triggers. The situation perspective places the focus on "concrete problematic situations". The choice perspective allows for the exploration of a variety of solutions and is for use close to implementation. The use perspective places the focus on the stakeholder view and consequently this is the perspective of scenario used with use cases and most relevant to our generic modelling. Cheesman and Daniels' [12] advocate the use of scenarios as Use Case Descriptions with the emphasis on the goal to be achieved by the enactment of the use case. In our work we exploited structured text based scenarios, Event Flows [13], that capture the sequential flow of the ordinary events that occur in within the confines of a use case and allow both the stakeholder view and the goal to be depicted.

A consequence of the application of scenarios to describe the Use Cases within the model is the use of a domain specific language. To support communication between the HEIs and the generic nature of the evolving model, a dictionary, containing some agreed upon generic terms and definitions, was created. In Formal Analysis the dictionary was used as a foundation for the development Thesaurus. The thesaurus provided the storage and access point for the domain specific vocabulary needed in the development of the generic model. In addition, the equivalent and hierarchical relationships between the generic terms defined in the thesaurus contribute to the domain knowledge when the thesaurus is included as part of the generic model.

Core Requirements Specification and Modelling

The generation of the Use Case Model illustrating and defining the core business elements of the institution provides the model needed to support specification any requirements for proposed evolution. Initially this will consist primarily of the use cases and scenarios describing the current state of the generic organisation generated in Formal Analysis. Subsequent requirements gathering and elicitation will produce additional use cases generated to explore proposed value-added services, such as web access to legacy data stores. Specifying requirements necessitates a more detailed view of the organisation than the one needed in analysis. As a consequence high-level class diagrams concerning domain elements need to be developed. These class diagrams will model elements close to the domain and are directly traceable to implementation [12, 13].

The thesaurus evolving with each pass through Formal Analysis contains an object-oriented classification that provides its overall structure and aids with the generation of the high-level class diagrams resulting from requirements gathering and elicitation. The thesaurus provides traceability of the domain terms throughout the development process.

Design and Implementation

The work products developed during the specification of the requirements for value-added services are used as the foundation for work performed in design and implementation. Here the Use Case Model is evolved to include the Logical View where the domain specific use cases with accompanying scenarios, and class diagrams that are less abstract and close to the actual implementation of the value-added services with each organisation [12, 13] are held. As a consequence the Logical View section of the model is less abstract and of less use outside the institution or an individual department with the institution. The less abstract domain knowledge is passed into the next iteration of the cycle providing the foundation for evolution to the Formal Analysis work products. For inter-organisational system models it is necessary for the design to remain at a higher level of generisity.

Additional Formal Analysis

Additional Formal Analysis is specialised and performed to support analysis with a particular focus. It can be performed at any point in the generation of value-added services but requires the use of tools appropriate to the specific focus. It is the motivation behind the activity that decides the selection of the supporting modelling notation. For example, sequence diagrams are generated to explore concurrent processes, an activity more suited to but not restricted to specifying or modelling core requirements. Where as activity diagrams support a focus on the systems actors by showing the consequences of their key activities when interacting within a process and are useful when a detailed examination of user activities are required [13]. Deployment diagrams support the abstracting away of unnecessary detail from complex distributed system implementations.

3. The Evolution of an Ontology

People's understanding of a language increases when they can place the terms (words and phrases) of the language in context [14]. By placing words and phrases of a language into context and using them, people learn to understand the syntax and semantics of a specific terminology. Within a specific domain the way in which the terms of the domain are applied to specific concepts and the identification of the relationships that exist between the terms provide additional richness to the depicted knowledge of the domain.

Dictionary

The first activities performed during the Initial Analysis of the undergraduate registration processes centred on the assembly of separate informal models of the different HEI's registration process. As the intention was to construct a generic model of the registration process the two informal models were expanded to include UML diagrams providing both HEIs with a common modelling language. It was originally intended that the use of UML would supplement each HEI's specific knowledge and at the same time facilitate understanding between the staff at the two HEIs during the process of comparing domain knowledge. At the time it was believed that the joint use of UML would aid in the identification of the commonality and disparity in the two different HEI registration processes required in the development of a generic model. It was during the comparison that the vocabulary difficulties arose when trying to communicate concepts and process information between the HEIs. Firstly, the basic block diagrams used in Initial Analysis captured only domain knowledge specific to the HEI and supported the domain vocabulary specific to the HEI. For example, the use of the term "Old Shire Hall" is a colloquial way, at Durham, of referring to the university's collection of central administration services. This arises from the fact that the majority of the central administration service departments are located a splendid old building called "Old Shire Hall". Some of the colloquial terms that had seeped into the domain vocabulary used in the informal block diagrams and later used in the more formal UML diagrams. Secondly, there was a significant difference in the domain vocabulary in use at each HEI. The majority of the differences were eventually recognised as caused by the use of synonyms, such as "Registry Office" and "Student Planning and Assessment Department" which have the same general area of responsibility during registration. Initially the individual domain terms were reconciled by identifying their equivalency relationships. The equivalent terms were recorded in a dictionary of domain vocabulary along with their common definition and an agreed generic term to be used in the generic model. This meant that each generic term was linked to a single definition and one or more St. Andrews and Durham synonyms. Figure 2 below is a sample extracted from the Dictionary of Generic Terms.

St. Andrews	Durham	Generic Name	Description
Student	Student	Student	Undergraduate enrolling at HE
Registry Officer	Registration Allocation Desk	Registration Officer	Accepts completed Matriculation and Enrolment Forms (or Registration Forms)
Registry Officer	SPA	Registration Officer	Maintains centrally stored student data.

Figure 2 Dictionary of Generic Terms

Thesaurus

The focus of the effort expended in the first few passes through the iterative steps of the meta-process is on the modelling of the existing enterprise process under investigation, and the gathering and communicating domain knowledge about this area of the enterprise. The key results emerging from our effort to model the undergraduate registration process were the development of the Generic Registration Model [2] consisting of a UML based model and a domain specific thesaurus. The domain specific thesaurus exists as part of the generic model but is also used as a support tool for the work performed in the evolution of the HEI systems, specifically during subsequent Formal Analysis and Core Requirements Specification Modelling. The thesaurus is developed to provide substantial knowledge about the domain necessary in the more formal modelling of an enterprise required by Core Requirements Specification and Modelling.

A thesaurus is a collection of terms used to represent concepts within a specific domain and organised so that predefined relationships between the terms are made explicit [15, 16]. We use a thesaurus to store and define the domain’s terminology. While the dictionary developed during Initial Analysis provided the means to state equivalent relationships between terms the thesaurus is used to make explicit additional relationships between terms, specifically hierarchical and associative relationships. The ISO 2788 standard [15] describes equivalence relationships as those that cover synonyms and quasi-synonyms. Synonyms are terms that have the same, or nearly the same, meaning. Quasi-synonyms are terms that when used in natural languages are considered different but when used within the domain are treated as synonyms. Within equivalence relationships terms are designated as either preferred terms or non-preferred terms. In our thesaurus the preferred terms are the generic terms used in the generic model and the equivalence relationship is defined as the generic term "USED FOR" the Durham and St. Andrew terms.

The ISO 2788 standard [15] defines hierarchical relationships as superordination and subordination relationships. The more general or broader term is SUPERORDINATE to a more specific or narrower term and a narrower term is SUBORDINATE to a broader term. There are three types of hierarchical relationships: generic, hierarchical whole-part, and instance. Generic relationships are used to identify the link between a class and its members, where a broader term is a class and narrower term is a member of a class as in the class 'staff' and the member 'Registrar'. Hierarchical whole-part relationships are for a limited range of relationships where the actual working of the narrower term implies the name of its broader term; as in Durham (narrower term), England (broader term). Instance relationships occur between general terms, the classes, and individual instances of a term. For example 'Computer Science' is an instance of an 'Academic Unit'. Hierarchical relationships are supported in the thesaurus by the application of an object-oriented classification to each generic term. An object-oriented classification was applied to each generic term during Formal Analysis with a view to supporting eventual design and implementation of value-added services and to provide traceability of the terms throughout the development process. This approach was gleaned from Protégé 2000 a tool employed to support the construction of domain specific ontology [17]. During abstract modelling of the enterprise processes the object-oriented classification is usually focused on identification and definition of the super classes, classes and a few key objects. As the model of the process evolves from the Use Cases View to the Logical View, the classifications become detailed and elements such as objects, attributes and operations are identified. The Logic View used primarily in the Design and Implementation phase of enterprise process evolution is where the associative relationships held in the thesaurus are identified and defined. Aitchison and Gilchrist [18] state that associative relationships are the relationships that exists between terms which are bound conceptually in the minds of the members of a community but cannot be defined hierarchically or equivalently. The most common associative relationship in the thesaurus is the relationship between concepts and their properties [15] or classes related to their attributes and operations. For example an attribute of an 'Academic Unit' is the 'Faculty' to which it belongs. Figure 3 below is a sample extracted from the developed thesaurus.

Generic Term	**Classification**	**St. Andrews**	**Durham**	**Term Definition**	**Alternate Definitions**
Faculty	Attribute of AU	Faculty	Faculty	A group of related Academic Units.	An academic staff member of an HE
Academic Unit	Class	School	Department	A unit of research and teaching within a faculty.	
Non Academic Unit	Class	Meta-data term. Specific unit terms such as library are used.	Meta-data term. Specific unit terms such as library are used.	A meta-data name for all units within an HE that are not covered by the term Academic Unit.	

Figure 3 Thesaurus Extract

From Thesaurus to Ontology

In rationalising the domain vocabulary by developing a dictionary, we began to raise the old concept of the data dictionary to a higher level abstraction thereby making it more useful during Formal Analysis. Once the dictionary was progressed to a Thesaurus with a structure based on object-oriented classification and it became useful throughout the iterative development life cycle including the Core Requirements Specification and Modelling. However, the addition of the object-oriented classification area to the thesaurus created some confusion about the relationships that exist between objects in the 'real world' and object-oriented relationship between terms in the model. To clarify the correlation between real world objects' relationships and object-oriented relationship implied by the object-oriented classification, several diagrams depicting the main relationships were provided.

The current large HEI business process under investigation by INSIDE is the exchange of Student Records between UK HEIs, specifically Durham and St. Andrews. This has involved a comprehensive analysis of Student Records including their structure, data content, manipulations and restrictions consistent with work performed in the evolution of the Logical View of the generic model. This has also entailed an exploration into the use of XML making it perhaps necessary to expand the Thesaurus to include the appropriate XML Specification classification. Thereby making it necessary to locate more sophisticated support tools for the Thesaurus that will allow changes within it to be reflected in other work products. The requirement for a multi-layered understanding of the domain in which the enterprise system lives and the need to understand models of the domain at varying levels of generalisation have led us to an investigation into the use of a domain specific ontology.

A domain specific ontology is a knowledge management tool used to support communication and knowledge reuse about a specific domain. Like a thesaurus, an ontology is a collection of terms used to represent concepts within a specific domain and organised so that predefined relationships between the terms are made explicit [16]. An ontology also promotes multi-layered knowledge acquisition and sharing by providing a repository for the general and detailed knowledge about specific domains [19]. However, an ontology can be difficult and time-consuming to produce [20]. Holsapple and Joshi [21] present five basic approaches for ontology development:

- Inspiration where the focus is on an individual viewpoint of the domain;
- Induction where the focus is on in-depth knowledge of a specific area with the wider domain;
- Deduction where the focus is on the general principles of the domain;
- Synthesis where a base set of ontologies are identified and used to represent specific subsections of the domain; and
- Collaboration where the viewpoints of many individuals are requested and then represented.

Several of these basic approaches have been applied to the construction of the ontology for the INSIDE project. The ontology was seeded or based on the terms and relationships contained in the established thesaurus. This base provides an ontology with a focus on the general principles of the domain. This provides an ontology to support domain knowledge reuse of some general HEI domain knowledge and detailed knowledge of the registration process with the HEI domain. It is intended that subsequent modelling of the domain, i.e. subsequent passes through the meta-process, will contribute to the evolution of the ontology increasing both the breath of general domain knowledge and the depth of domain knowledge in specific domain areas. Thereby providing a multi-layered view of the concepts of the domain and the relationships between those concepts. Protégé 2000 has been the support tool selected for use in the development and evolution of the ontology. Figure 4 below illustrates the evolution of the work products in the construction of the ontology.

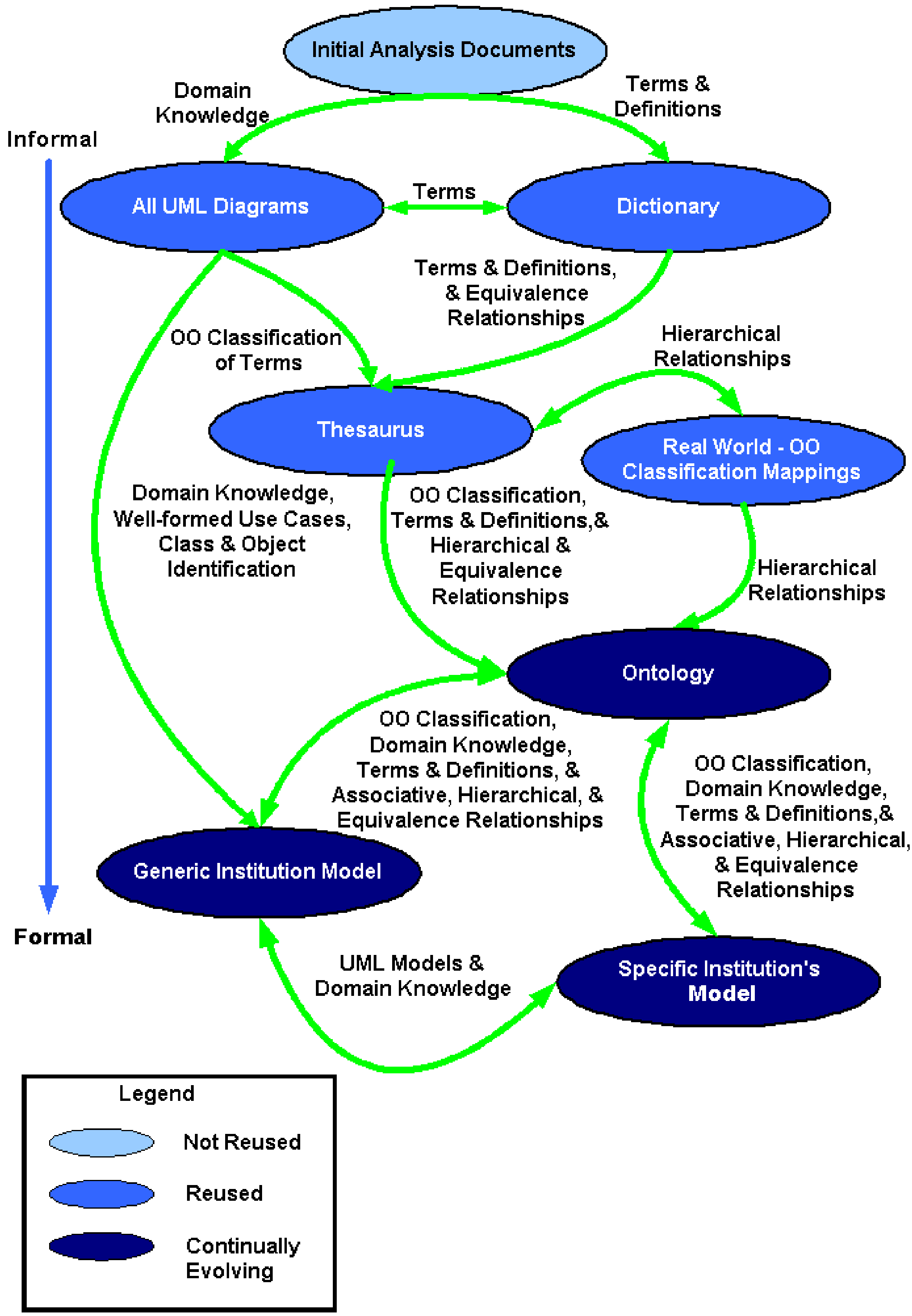

Figure 4 Evolution of Work Products

4. Further Work

The brownfield sites that are the enterprise systems in place in HEIs today are in the process of being reclaimed and redeveloped to provide the users of these systems with more useful web accessible systems. The INSIDE project has abstracted a meta-process to support the evolution of enterprise systems with the incremental development of the value-added web accessible systems as well as inter-organisational enterprise systems, such as those required to support lifelong learning. The meta-process reinforces the iterative nature of incremental domain knowledge capture and modelling in conjunction with iterative development of the value-added systems. In addition, the meta-process demonstrates the practical application of the various UML notation used to support analysis, requirements, and design of legacy systems. However, the evolution of large enterprise systems requires a multi-layered understanding of the domain in which the system lives. Use and evolution of the ontology is one of the issues currently being explored in the development of a Student Records Exchange system. This will provide the opportunity to evolve the Generic Registration Process Model into a more comprehensive and potentially useful Generic Student Information System Management Model. An exploration of the use of XML is part of this investigation. We are currently determining the effects of the use of XML on the ontology and other work products.

5. Acknowledgement

The work described in this paper is part of the Institutionally Secure Integrated Data Environment project (INSIDE), which is funded by the JISC Committee for Integrated Environments for Learners (JCIEL) under the Building Managed Learning Environments in HE (7/99) programme. INSIDE is a collaborative project between the Universities of St Andrew and Durham. The work described here has involved contributions from colleagues at both universities. We are especially grateful to our Durham colleagues Brendan Hodgson and Sarah Drummond; and our research partners Colin Allison and Bin Ling (Jordan) at St. Andrews.

6. References

1 UK Government, Environment Agency, Land Quality, 2002. http://www.environement-agency.gov.uk/subjects/landquality/

2 Lavery, J. (2002): Report on the Generic Model of the Process of Undergraduate Registration at Higher Education Institutions Version 2.0, *Institutionally Secure Integrated Data Environment Report*, January 2002. http://www.dcs.st-andrews.ac.uk/inside/report.html

3 Prieto-Díaz, R. and Arango G., (1991): *Domain Analysis and Software Systems Modelling*, Los Alamitos, IEEE Computer Society Press.

4 Kruchten, Philippe B. (1995): The 4 + 1 View Model of Architecture, *IEEE Software*, Vol. 12, No. 6, pp. 42-50.

5 Prieto-Díaz, R., (1991): Domain Analysis For Reusability, *Domain Analysis and Software Systems Modelling*, IEEE Computer Society Press.

6 Watson A., (2001): OMG and Open Software Standards a *Business and Professional Lecture*, the University of Durham.

7 Gomaa, H., (2000): Object Oriented Analysis and Modeling for Families of Systems with UML, W.B. Frakes (Ed.), Software Reuse: Advances in Software Reusability, ICSR-6 Proceedings, *Lecture Notes in Computer Science*, Vol. 1844, pp. 89-99.

8 Kang, K.C., and J. Lee, (2002): Feature-Oriented Product Line Engineering, IEEE Software, Vol. 19, No. 4, pp.58-65.

9 Frakes, W., R. Prieto-Díaz, and C. Fox, (1998): DARE: Domain analysis and reuse environment, *Annals of Software Engineering*, Vol. 5, pp. 125-141.

10 Rational Rose Tutorial part of *Rational Rose 2000*, Rose Enterprise Edition, Copyright © 1991-1999, Rational Software Corporation.

11 Bustard, D.W., Z. He, F.G. Wilkie, (2000): Linking soft systems and use-case modelling through scenarios, *Interacting with Computers*, Vol. 13, pp. 97-110.

12 Cheesman, John, and John Daniels, (2001): *UML Components A Simple Process for Specifying Component-Based Software*, Addison-Wesley, New Jersey, USA.

13 Quatrani, Terry, (2000): *Visual Modeling with Rational Rose 2000 and UML*, Addison-Wesley, New Jersey, USA.

14 Brooks, JR., Frederick P., (1995): *The Mythical Man-Month Essays on Software Engineering*, Anniversary Edition, Addison-Wesley Publishing Company.

15 *International Standard ISO 2788 - Documentation - Guidelines for the establishment and development of monolingual thesauri*, Second edition, 1986-11-15.

16 Rada, R., (1990): "Maintaining Thesauri and Metathesauri", *International Classification*, Vol. 17, No. 3/4, pp. 158-164.

17 Noy N. F., M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, & M. A. Musen. (2001): Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2), pp. 60-71.

18 Aitchison, J., and Alan Gilchrist, (1972): Thesaurus Construction A Practical Manual, Aslib.

19 Valente, A., T. Russ, R. MacGregor, and W. Swartout, (1999): Building and (Re)Using an Ontology of Air Campaign Planning, *IEEE Intelligent Systems*, Vol. 14, No. 1, pp. 27-36.

20 Gruninger, M., and J. Lee, (2002): Ontology Application and Design, *Communications of the ACM*, Vol. 45, No. 2, pp. 39-41.

21 Holsapple, C. W. and K.D. Joshi, (2002): A Collaborative Approach to Ontology Design, *Communications of the ACM*, Vol. 45, No. 2, pp. 43-47.