

# Modelling Requirements and Architecting Large-Scale On-Line Competence-Based Learning Systems

Geoffrey Darnton

*Bournemouth University Business School and Meta Modelling Ltd.*

*E-mail: [gdarnton@darnton.info](mailto:gdarnton@darnton.info)*

## Abstract

*This paper presents a summary of the key issues that arose while creating models and an architecture for a large-scale target information processing system (given the pseudonym LASCO). LASCO combines requirements to handle competence, learning content, on-line delivery, and learning management. LASCO is targeted at enterprises with in excess of 1,000, and typically several thousand, employees who need to participate in learning systems while selecting material from catalogues that are likely to contain in excess of 100,000 learning objects. Any subject matter could be involved. An additional requirement for the architecture and modelling was to support compliance with a range of currently emerging standards.*

*Another goal of LASCO is to maximize the use of on-line delivery and management because of the volumes involved. However, it must be capable of supporting the management of traditional forms of delivery to provide a single learning management system.*

*Tools were needed to manage the modelling activity and create new relevant languages to support the modelling and architectural work.*

*Current emerging standards may be useful for small, localized systems but they will need substantial development to support large-scale industry strength for on-line competence based learning systems.*

## 1. Background and Relevance

This paper reports summary results arising from consulting and analysis projects carried out by me in two multi-national enterprises.

Client 1 is a multinational company delivering a wide variety of training in many countries, ranging from traditional classroom situations to online delivery of computer based training (CBT).

Client 2 is a multinational company providing a managed service for the delivery of online training only. It does not develop any content and only provides a

managed service sourcing material from a variety of content developers and providers.

This paper discusses a combination of the requirements derived from both situations, each of which is very different in the underlying system requirements. When combined, what emerges is an architecture that covers a wide range of requirements likely to be found by an enterprise that wishes to provide large-scale on-line learning, combined with traditional forms of learning. Let me call this target information process system LASCO.

This practical experience is relevant to several Conference topics of interest. It gives a summary of the key issues encountered while modelling requirements and creating an associated architecture. Both Clients have the problem of delivering training to their clients who are likely to have thousands of employees. Therefore the numbers of “students” involved exceed the size of most universities. The teaching and learning strategies may need to be linked explicitly to enterprise strategies due to Client belief that continuous skills development must be successful if their clients are to survive in environments that are turbulent and subject to continual technological change.

My role has been to collect and analyze the overall requirements of these clients. The principal deliverables of both engagements have been the production of (1) detailed data and process models of systems that need to be developed to support the operations of the clients (2) the delivery of system requirements specifications (3) associated meta models. These have been combined to create LASCO (this is not as hypothetical as it may sound as both Clients commissioned the same system developers).

## 2. Key Issues

Client 1 has a broad portfolio of courses and delivery scenarios. A typical client of Client 1 wants a large scale competence based learning system for several thousand employees. Competences may range from “soft” skills such as leadership and team performance through to “hard” skills such as Network Maintenance and use of

common office products. For example, one scenario involved the competence management of 40,000 employees. Most (more than 50%) employees were located in the USA, with the balance located in several other countries around the globe. Content was provided by a variety of sources. The competences needed to be linked demonstrably to corporate and business strategies.

Client 2 deals with clients who need to provide on-line training to at least 1,000 employees. Content is sourced from multiple providers. A typical scenario would involve a client deciding what on-line training is needed in the following time period (say 1 or 2 years) and then selecting from a very extensive catalogue of available content. Client 2 is unable to provide any warranty for material from another source, but will maintain quality control procedures based on experience and feedback from clients. This client has a different large scale problem to solve in that content comes from multiple sources, and the catalogue can contain thousands of possible courses. For example one content provider offered in excess of 60,000 on-line courses.

Both Client 1 and Client 2 needed to commission the development of special software systems to handle their particular processing requirements. Off-the-shelf products were not available so bespoke systems had to be developed. Hence, they both needed software requirements specifications.

Combining the requirements of both Clients, the key issues to emerge which needed to be addressed in the resulting architectures and software requirements specifications were:

- competence management - Client 1 customers needed to express their business strategies in terms of required competencies. Training delivered had to make demonstrable contribution to those competence requirements;
- content management - both Client 1 and Client 2 obtain content from multiple sources. It is not feasible for one of their clients to read the details of 100,000+ possible courses. Therefore a classification scheme is needed to assist the identification of potentially relevant material;
- on-line delivery - both Client 1 and Client 2 needed to make on-line courses available to thousands of "students" with acceptable levels of service;
- learning management - the progress of "students" through the learning material needs tracking, possibly for several quite different reasons, but primarily it is necessary at least for the "student" to know that effective learning has taken place, and others may need to know that as well. Content providers and learning management system (LMS) service providers need to know about usage for billing, contractual, and feedback purposes;

- external links - an important consideration by the largest clients of both my Clients was connecting LASCO to external systems, particularly enterprise resource planning (ERPs). The most common scenario was building links to SAP/R3 installations. SAP/R3 could help with some training logistics and some personnel record keeping, but competence and learning management needed to be handled externally, with appropriate interfaces. Combined with other limitations in SAP/R3 this usually involved "wrapping" SAP/R3 -- shattering a common dream of what ERPs can deliver;
- meta models - the development of software requirements specifications, business process models, and architecture required an overall meta model against which completeness and consistency of the project deliverables could be assessed;
- compliance with standards - LASCO should be capable of generating output for which non-trivial claims of compliance with a range of emerging standards can be asserted and sustained. This included AICC [1], Dublin Core [2], IEEE [3], IMS [4], ISO [5], SCORM [6].

### 3. Meta Models

The production of software requirements specifications (SRSs) and architectural models needs the development of meta models to underpin required levels of completeness and consistency. I take meta models to include data, processes, and standards. This approach to meta modelling also renders any object oriented versus structured analysis debate as irrelevant, as either ideological form can be generated.

The models for these projects were developed and maintained using Problem Statement Language/Problem Statement Analyzer (PSL/PSA) [7] [8] [9]. The software requirements specifications (SRS) were generated in accordance with an IEEE Standard [10] as described by Darnton [11]. This required:

- defining the software requirements specification model to use;
- mapping each element of the SRS to an appropriate subset of PSL/PSA;
- defining required levels of completeness and consistency in the specifications;
- producing PSA database queries to check and report on completeness and consistency;
- producing a PSA database query to generate IEEE Std 830 compliant SRSs;
- using PSA to perform a canonical synthesis on the underlying data model to generate a meta data schema.

Creating a Course Description Language and a Competence Language (combined and called CDLCOMP) was achieved using tools known as Language Definition Manager (LDM) and System Encyclopedia Manager (SEM) [12] [9].

#### 4. Competence Management

What is competence? Various models are proposed or discussed by different authors (for example [13] [14] [15] [16]). There does not appear to be general agreement, or any readily available standard for this. Therefore it was necessary to identify an operational definition that could be used for modelling and architecture. The final model used was of the nature that:

competence = skills + knowledge + attitude

When competences are to be linked to strategy, further work needs to be done. An enterprise asserts what activities it must be good at doing if it is to be successful. However, it is then necessary to know what is believed about competences required to do those activities well.

From the LASCO model, high level processes are:

- express enterprise objectives and strategies in terms of activities and competences;
- identify contexts in which activities are performed;
- build and maintain a dictionary of competences;
- create an inventory of required competencies;
- assess competence of individuals and teams;
- identify gaps between required inventory and current existing competences;
- define required competence development;
- identify content for competence development;
- plan competence development;
- deliver competence development program;
- monitor competence levels (and evolving requirements).

#### 5. Content Management

So many sources of on-line content are emerging at present. Identifying suitable content for achieving identified learning outcomes has become a major issue given the number of learning objects available along with the variety of access and licensing models. The scale of the problem is now so large that it cannot be done properly without tools to assist. Some content providers charge per view. Some charge per period of time. Some charge per person who is permitted to view.

Some content providers include a registration and monitoring mechanism along with the content. Thus, a user must “register” within the overall package, and this registration information may be used for tasks such as permitting access, monitoring usage, and monitoring

progress.

These problems led to two architectural requirements: (1) the separation of content and its delivery from the management of access (2) methods for the classification of content.

The modelling work led to the creation of a specific Course Description Language because it was not possible to produce a workable mapping between the requirements and PSL. LDM was used to create the language. Use of the tool ensured that there was a set of objects and relationships expressed as a verified semantic network capable of implementation.

#### 6. On-Line Delivery

As this paper is focussed on modelling and architectural issues, on-line delivery requirements are discussed in those terms (rather than the physical implementation issues of delivering volume on-line material in operational terms).

The key initial problems are defining a “unit of delivery” and managing the whole context within which delivery takes place.

The range of dimensions of context should not be underestimated. Contextual matters that have had to be taken into account in building meta models and defining an architecture are:

- modularity
- rights of access and security schemas
- mechanisms for access
- pre-requisites, co-requisites, and post-requisites
- accreditation
- bookmarking
- content classifications
- usage statistics
- links to support services (on-line and off-line)
- progression
- relationships between content and other issues such as competences, objectives, and strategies
- relationships between content, progression, and performance (individual, group, and corporate).

The greatest difficulty with the use of internet technology and delivery over wide area networks is providing reliable estimates of delivery service performance.

#### 7. Learning Management

The need to separate content from the management of access and learning is broadly understood and represented in the emerging standards already referred to. However, for many enterprises, the issue is far wider than computer managed instruction and will need to make provision for supporting the management of all forms of instruction.

In terms of defining requirements and creating a model of tools to support an LMS, the IMS Final Specification [4] was too simple to be the model foundation. The underlying data model is stated explicitly as Person, Group, and Group Membership. Groups may consist of Persons or other Groups. Later in the document there are many other implied data objects which have clearly not yet been included in the overall data model diagram.

The model was developed from a combination of requirements from Client 1 and Client 2, and it required a substantial number of objects and relationships in addition to the IMS required objects, to produce even a simple LMS.

## 8. Architectural Observations

The production of an overall architecture for the combined functionality for these two projects required the choice of an approach to architecture. My choice was between Zachman [17] and Darnton & Giacoletto [18]. As these projects involved extensively distributed systems, along with linkages from business strategy to implemented systems, the approach to architecture adopted was that set out in Darnton & Giacoletto.

This approach requires the identification of four architectural “layers”: business; systems; technical; product.

Business and systems architecture are informed by (1) client requirements (2) emerging standards. Technical architecture requires a consideration of basic technical issues to be resolved in any target information processing system.

In modelling the requirements to handle the integration and interaction between competence, content, delivery, and learning management, many n-part relationships were identified. For example, modelling that a person can perform an activity to a particular level of competence in an identified context is a 4-part relation.

In addition, there are multiple instances of indefinitely recursive structures (analogous to bill of materials structures). For example, learning objects can consist of learning objects (and the same for learning outcomes).

These are requirements that are exceptionally difficult to manage if the choice of database management system is relational. The difficulties lie in implementing the required data model, and defining the code required to maintain integrity in the system. I have found that the most efficient approach to database management system is a hybrid involving primarily a CODASYL database management system and some multidimensional structures. In terms of commercial implementations of the various data models, the principal restriction in CODASYL DBMSs is that they implement set types with

only one owner record type, whereas the direct implementation of n-part relationships is most efficient with a DBMS that permits set types with n owner record types. This is available in the PSA DBMS used to support the modelling of the requirements. It is not available in commercially available CODASYL DBMSs (the product used for prototyping was Oracle's CODASYL DBMS [19], originally developed by Digital Equipment Corporation), and junction records are needed for n-part relationships, or multidimensional structures.

These modelling projects raise concerns about the current state of emerging standards. For example, the IEEE LTSC Learning Object Metadata Tag 1.8 concerning aggregation level only shows 4 levels (0,1,2,3) whereas in terms of meta models, that kind of structure is really an indeterminate recursive structure, and aggregation level should be computed rather than prescribed arbitrarily. The consequences can be serious for the architect. An information architect needs to define the required recursive structure, if the resulting software is to be capable of representing material from a very wide range of sources, including on-line and traditional forms. However, in order to be compliant with an emerging standard such as IEEE's Learning Object Model, it will be necessary to introduce code to perform some arbitrary mapping to the artificially and arbitrarily constrained structure in the standard. Similarly, the definition of taxons (9.2.2) is artificially constrained to 9 hierarchical levels, whereas in practice the architect needs to make provision for a network (rather than hierarchy) of indeterminately recursive structures, and then map that to whatever standards are applied to the content.

These problems can be handled more easily if attention is paid to the requirement for Technical Architecture [18]. That would require articulation of the nature of the data, relationships, and processing. The articulation would then inform a choice of product and platform.

One common approach to architecture, and perhaps the simplest, is to define major chunks of functionality with a published application programming interface (API). If this is done, it is still necessary to write something else that “calls” the API. This “something else” increases in complexity as the number of chunks increases. Therefore the architectural approach I have taken is a “bus” approach, which reduces the complexity of integrating competence, content, delivery, learning, and external interface chunks. In terms of current emerging standards, this would imply going back to the drawing board and re-defining their scope.

## 9. Concluding Comments

I am aware that there are many projects to implement the emerging standards referenced here, and that this has

resulted in much useful feedback to those standards efforts. The modelling reported here has proceeded on an entirely different footing. Commercially viable and technically feasible ways to solve the problems identified earlier were needed, with industry strength architecture. First, requirements have been obtained. Second, those requirements have been expressed as models and analyzed for completeness and consistency (in terms of generating an SRS). An additional activity was introduced to map elements of the models to elements of those emerging standards. Then logical models were produced to inform the design and choice of platform activities. The overall result is not particularly encouraging for those emerging standards. Architecting systems that satisfied the requirements meant going far beyond those emerging standards. Therefore the conclusion is that as they stand, the data models presented in those standards are technically naive and oversimplistic (for the purposes identified here). Thus future architects will have a real dilemma whether to architect an industry strength suite of tools, or an implementation to comply with emerging standards.

## 12. References

- [1] AICC – Aviation Industry Computer Based Training Committee Guidelines for File-Based CMI [Computer Managed Instruction] Systems and CBT Courseware (AGR-006) and Guidelines for Web-based CMI Systems and CBT Courseware (AGR-010). available at <http://www.aicc.org/pages/down-docs-index.htm>
- [2] Dublin Core Metadata Element Set available at: <http://dublincore.org/documents/1998/09/dces/#>.
- [3] IEEE Standard P1484 – Standard for Information Technology – Education and Training Systems – Architecture and Reference Model. Working Group Home Page: <http://ltsc.ieee.org/wg1/index.html>.
- [4] IMS Enterprise Information Model Version 1.1 Public Draft Specification, IMS Global Learning Consortium, Inc. Available at: [http://imsproject.org/enterprise/ent\\_v1p1pd/imsent\\_infop1p1pd.html](http://imsproject.org/enterprise/ent_v1p1pd/imsent_infop1p1pd.html).
- [5] ISO standards for Information Technology for Learning, Education, and Training, ISO/IEC JTC1 SC36 Home page at: <http://jtc1sc36.org/>
- [6] SCORM (Sharable Content Object Reference Model), Advanced Distributed Learning, home page at: <http://www.adlnet.org/>
- [7] Teichroew, D., and, Hershey, E.A. III, “PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems”, *IEEE Transactions on Software Engineering* Vol. 3 No. 1 January 1977, pp. 41-48.
- [8] Sayani, H.H., “PSL/PSA at the Age of Fifteen”, in Thayer, R.H., and, Dorfman, M. (eds.) *System and Software Requirements Engineering*, IEEE Computer Society Press, Los Alamitos, 1990, pp. 403-417.
- [9] PSL/PSA, *Problem Statement Language/Problem Statement Analyzer* website, [www.pslpsa.com](http://www.pslpsa.com). 2002.
- [10] IEEE Standard 830 Recommended Practice for Software Requirements Specifications, IEEE, Piscataway.
- [11] Darnton, G., *Automating the Generation of Functional Specifications*, Proceedings of CASE Conference, ISDOS, Ann Arbor. 1987.
- [12] Teichroew, D., Macasovic, D., Hershey, E.A., and, Yamamoto, Y., “Application of the entity-relationship approach to information processing systems modelling”, in *Entity-relationship Approach to System Analysis and Design*, North-Holland. 1980.
- [13] Boyatzis, R., *The Competent Manager*, John Wiley & Sons, Chichester, 1982.
- [14] Edmunds, T. & Teh, M., “Personal competence: where does it fit in?”, *Competence & Assessment*, Issue 13, 1994, pp. 6-8.
- [15] Rifkin, K., Fineman, M., & Ruhnke, C., “Developing technical managers: first you need a competency model”, *Research Technology Management*, Vol. 42 Iss. 2, 1999, pp. 53-57.
- [16] Whiddett, S. & Hollyforde, S., *The Competency Handbook*, Institute of Personnel Development, London, 1999.
- [17] Zachman, J., “A Framework for Information Systems Architecture”, *IBM Systems Journal* Vol. 26, No. 3. 1987.
- [18] Darnton, G. and Giacoletto, S., *Information in the Enterprise: it's more than technology*, Digital Press, Burlington. 1992.
- [19] Oracle CODASYL DBMS – information available at <http://otn.oracle.com/products/rdb/pdf/dbms.pdf>