Using ArchiMate to design learning environment architectures

W.G. Kraan
(Institute for Educational Cybernetics, University of Bolton, w.g.kraan@ovod.net)

Abstract: A desire to customise and personalise learning experiences, combined with the rise of a number of new tool integration technologies has led to a move away from monolithic Virtual Learning Environments (VLEs, also known as Learning Management Systems 'LMSs') towards more open and Distributed Learning Environments (DLEs). The various DLE architectures can have very different properties, however, and choosing between them can be difficult.
The Open Group's ArchiMate standard may help in that regard. It was designed to facilitate communication about architectures between all stakeholders in an organisation. It is a visual language that aims to help conversations about IT systems, business processes, organisational structure and strategy.
This paper will present a number of DLE patterns that illustrate the range of possible architectures. Both the potential of these patterns as well as the utility of the ArchiMate language in explicating them will be evaluated.
Keywords: Enterprise Architecture; VLE; DLE; LMS; learning technology
CLC number: Q948 **Document code**：A

0 Introduction

The Virtual Learning Environment (VLE, also known as a Learning Management System 'LMS') has become a dominant design in digital educational systems (Wilson et al., 2006). While the design may have strengths in the high level of control that it offers educational institutions, as well as the relative ease of deployment of its single system architecture, there are significant limitations to the learning experience it provides. The most important of those are the asymmetric relation between learner and institution, and the dominance of the course as the sole organising principle.
The Personal Learning Environment (PLE) was conceived as way to address those limitations (Wilson et al., 2006). By making use of newer, web-based technologies, a learner can compose their PLE out of a wide range of services, both from within an institution as well as outside of it. That way, the learning experience can become richer, and much more personalised in the way it adapts to the interests and preferences of the individual learner. Not least because a PLE can persist beyond formal learning while studying at an institution and thereafter. At the same time, the more balanced levels of control over the environment can make exchanges between learners, teachers and the institution more open and participatory.
As the over one hundred papers cited by Buchem, Attwell, & Torres (2011) make clear, the PLE concept is still subject to vigorous debate, and research into PLEs has become a field in its own right. At the same time, the VLE continues to be the dominant design. Part of the reason for this might be the persistent nature of dominant designs in general (Abernathy & Utterback, 1978), but other factors could include the high digital literacy required in PLE users, as well as a potential loss of the social and managerial coordinating function of a traditional VLE.
At the same time, these VLEs have started to incorporate technologies that are similar to or the same as the one that made PLEs possible in the first place. More or less formally standardised Application Programming Interfaces (APIs) such as the W3C's Widget specification (Cáceres, 2011), OpenSocial (OpenSocial and Gadgets Specification Group, 2011) and IMS Learning Technology Interoperability (IMS Global Learning Consortium, 2011a) enable VLEs to participate in the networked architectures that also characterise PLEs.
This means that a hybrid between the VLE and PLE concepts is becoming possible, and could address the shortcomings of both. Such a Distributed Learning Environment (DLE) could retain various degrees of central versus personal control or administration, while retaining degrees of personalisation and flexibility (MacNeill & Kraan, 2010). The development of DLEs has been taken forward by a number of projects in the JISC Distributed VLE programme (JISC, 2010).
A wide range of DLE architectures is possible, however, each with different system components, varying degrees of institutional control and different learning affordances. For that reason, a comprehensible and holistic view of how various DLE models could fit into a learning and teaching organisation is a pre-requisite for an informed choice between them. The field of Enterprise Architecture (EA) has pursued such a holistic approach, as is evident from one of the more widely used definitions of the term: “The structure of components, their inter-relationships, and the principles and guidelines governing their design and evolution over time” (The Open Group, 2009) One part of EA is about such methods and guidelines, as well as a wider policy framework. The other part of the approach is the description of the organisation in its key aspects; processes, people and departments, information and IT systems and the infrastructure they rely on. For the latter aspect, the ArchiMate specification (Iacob, Jonkers, Lankhorst, & Proper, 2009) provides a visual language that is designed to make an

organisation's architecture relevant and comprehensible to all stakeholders in that organisation. Experience in the JISC's Enterprise Architecture Practice Group (EAPG) suggests that the language works as expected in UK Higher and Further Education institutions (JISC Infonet, 2011).
For that reason, ArchiMate has the potential to be a good solution for the description and analysis of the various PLE models.

1 Enterprise Architecture and ArchiMate

As noted in the introduction, Enterprise Architecture is both a holistic approach to managing change in an organisation, as well as a description of the current ("as is") and desired ("to be") states of an organisation, its systems, infrastructure, processes, information and people. While the application of an EA framework to learning and teaching organisations is becoming a well established and fruitful endeavour (JISC Infonet, 2011), this paper will focus on the descriptive aspect.

The role of a modelling language is important in EA, because it needs to both record the organisation comprehensively, and – most importantly – to communicate it to all stakeholders (Lankhorst, 2009). For that reason, many EA modelling languages have been developed over the years, with varying degrees of success.

Among them, ArchiMate (Iacob et al., 2009) stands out for a number of reasons. First, it is an open specification rather than tied to a particular tool. Second, it is related to the widely used Unified Modelling Language, but greatly simplified. Third, it is explicitly designed to encompass all aspects of an organisation, not just a process or system. Fourth, it is explicitly designed for the purpose of communication, not code generation or configuration.

The language's need to be widely comprehensible clearly clashes with the need for it to be comprehensive and formally rigorous. One minor way in which this is achieved is by allowing the concepts that make up the language be represented in any way the user pleases. As long as the tool supports it, entities can be represented in their canonical diagrammatic form, as tables or as any desired icon. The major way in which comprehensibility is reconciled with comprehensiveness is with the notion of viewpoints. A viewpoint is a subset of the full organisational model that is tailored for a specific purpose, to a specific audience. That way, the model can be as comprehensive and rigorous as the analyst needs it to be, while other stakeholders are presented with excerpts that are designed to address their concerns concisely.

Within the scope of this paper, the purpose of the ArchiMate views that will illustrate the learning environment architectures is to enable comparison between them, according to the characteristics outlined in the next section. The audience will be assumed to be those generally knowledgeable in the area of e-learning, but not necessarily deeply involved in the maintenance of IT systems, nor in running the overall strategy of organisations. For that reason, details of the infrastructure on which these learning environments run have been left out, as have details about the direction of the learning and teaching institution. Likewise, the scope of these viewpoints is limited to learning environments rather than the whole organisation.

The result of these constraints is what could be termed a metamodel – a defined subset of the wider language. To maintain comparability between the learning patterns, all views will adhere to this metamodel, and they will represent the pattern examples with comparable functional requirements. That is, they will all represent the minimal high level set of entities and relations they need to function to a broadly similar level. All ArchiMate concepts will be *italicised* in the discussion, and a summary overview of the main ArchiMate concepts and relations is given in Fig. 1 (M. Lankhorst, 2004)

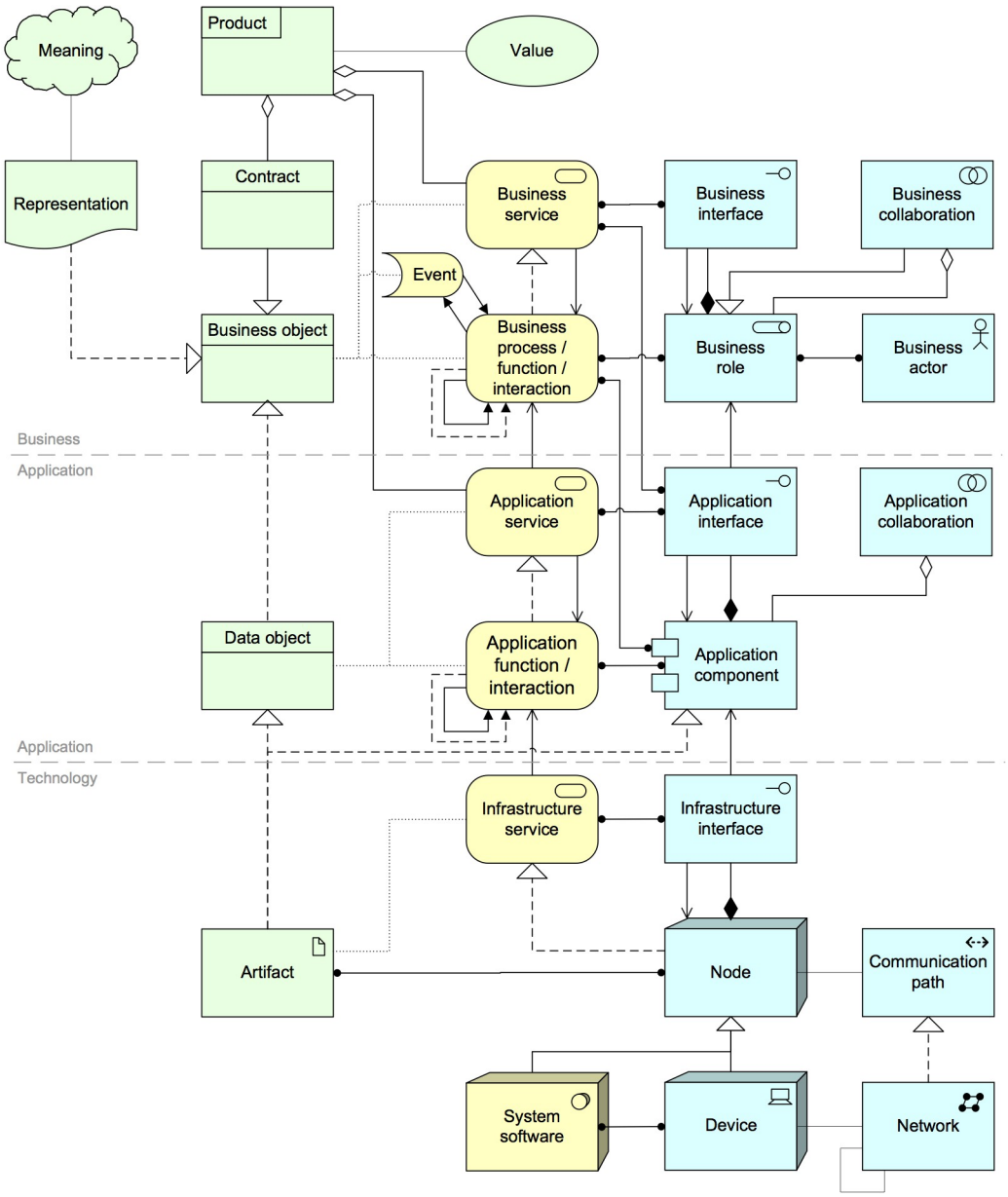

Fig. 1: Overview of the main concepts and relationships in the ArchiMate language

2 Learning environment architectures

In order to illustrate the conceptual difference between a PLE and a VLE, Wilson et al (2006) propose a series of characteristics or dimensions of learning environments:

- **Context focus / organisation**; what is taken as the primary context or organising principle in the arrangement of the learning environment
- **Relationship symmetricity**; how symmetric the relationship between learners and educators is, in terms of power to change the environment as well as their scope for participation
- **Context experience homogeneity**; how uniform or how personal the environment is for its participants
- **Nature of standards**; whether interoperability standards are education-only, or of wider application
- **Access control and rights management**; the degree to which resources and activities are open
- **Organisational scope**; the social group whom the environment is designed to support

Of these, the access control aspect overlaps heavily with both organisational scope and context focus, and will therefore not be considered separately here. The remainder captures the social and affective aspects of learning environments very well, but there is also the aspect of complexity to consider, both overall, as well as from the perspective of the learners, the educators and the institution.

Assuming that the conventional VLE will constitute one end of the spectrum of learning environments, and the PLE the other, the various DLE models should fall somewhere between the two. By keeping the diagram purpose, audience and also the entities relatively constant, any conceptual difference between the architectures should be evident.

2.1 VLE

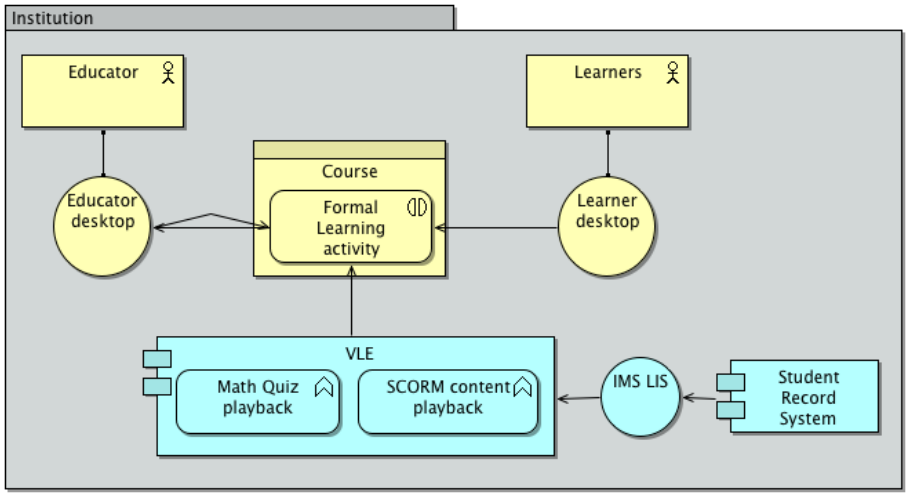

Fig. 2: Virtual Learning Environment pattern

The conventional VLE is characterised by the single central system that provides all functionality. In Fig. 2, the Educator and Learner *actors* are assigned to their desktop *business interfaces* (that is, the means by which an actor or role connects to its environment), which are *used by* a learning activity, which, in turn, *accesses* a course. The VLE *application component* is also *used by* the formal learning activity *business interaction* (a unit of behaviour performed as a collaboration between two or more business roles), and contains two illustrative *functions*. Person, membership and group data is supplied by the Student Record System *application component* via the IMS Learning Information Services (LIS) (IMS Global Learning Consortium, 2011b) *application interface*.

2.1.1 Context, focus and organisation

The main organising principle and context for a VLE is the course, which is to say that all interactions such as forum discussions and access to content are structured by the VLE to take place within the context of a course alone. Resources are often not available to colleagues in a different course or cohort, and often not available even to the learner once the course has been completed. The combination of such time and subject limitations with a desire to foster interest focussed, cross-cohort study groups has been known to drive educators to alternatives such as hosted VLEs or social networking services such as Ning (Savage & Erskine, 2009).

2.1.2 Relationship symmetricity

The reciprocal *used by* relation between the instructor desktop and the formal learning activity in Fig. 2 indicates that conventional VLEs rely on editing, configuration and uploading by educators and/or administrators rather than learners. Again, this may be convenient for organisational management purposes, but does hinder or prohibit more participatory or collaborative learning styles (Wilson et al., 2006)

2.1.3 Context experience

As Fig. 2 makes quite clear, there is no alternative to how the VLE presents resources, interactions and services. Nor is there room for augmentation of that interface with services from elsewhere. It is very homogenous.

2.1.4 Interoperability standards

The interoperability standards that a traditional VLE tends to support are limited to education-only specifications such as the use of SCORM (Advanced Distributed Learning, 2011) for the exchange of static content. This works well within the educational domain, but makes it more difficult to integrate potentially relevant resources from other domains.

2.1.5 Institutional scope

The VLE is designed to be run by and for one institution only. Even when the VLE is used as a service provided by a third party, access policies and the structure of a conventional VLE mean that potentially useful links with a variety of outside organisations are difficult or impossible to establish. From a lifelong and life-wide learning perspective, this fragmentation is not optimal because it means resources are not available beyond graduation or even beyond the academic year, and those resources are certainly not accessible to outside learners and experts (Wilson et al., 2006). The issues can be ameliorated by using a single VLE instance as a shared service in the manner of Korea's Cyber Home Learning System (Bae, Han, Lee, & Lee, 2008), but care needs to be taken to enable a degree of contextual flexibility because the contextual experience may be too uniform otherwise.

2.1.6 Complexity

Though some resources are required to acquire, install and maintain a VLE, it is quite clear from Fig. 2 that they are very simple and manageable from both an organisational and a technical perspective: there's only two software components, and they are fully controlled by the organisation. Pragmatically, that makes VLEs very attractive from an institutional point of view.

2.2 VLE with Plug-ins

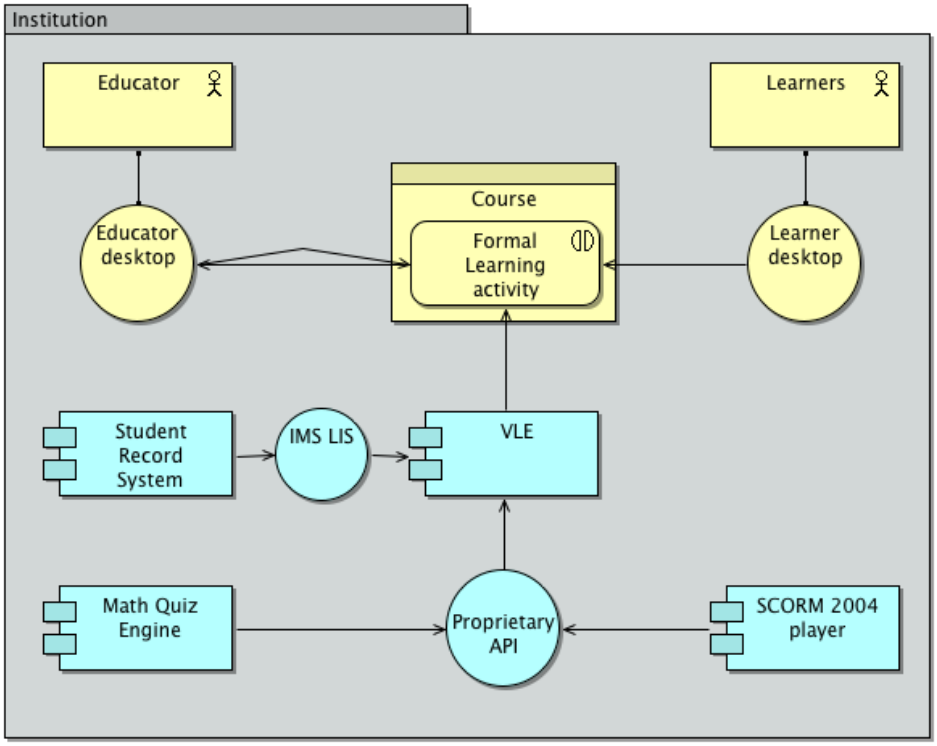

Fig. 3: VLE with plug-ins pattern

The first of the DLE models is a relatively minor, but popular departure from the dominant VLE pattern. Instead of built-in functions, specialised, independent software is integrated with the VLE to augment its features.

2.2.1 Context, focus and organisation

In terms of learning environment structure and focus, there is no difference with the dominant VLE pattern. Plug-ins are capable of spanning more than one context (Icodeon Ltd., 2009), but terminology such as 'course links' suggests that it is doubtful that many users define contexts other than the courses the VLE uses itself.

2.2.2 Relationship symmetricity

There is no difference with the VLE pattern: control over the structure and shape of the context is entirely up to the educator.

2.2.3 Context experience

The use of plug-ins have a lot of potential to customise the environment to the demands of a particular subject area, course or even learner or educator. For example, where a monolithic VLE's features are of necessity the lowest common denominator in areas such as quiz engines, plug-ins can provide specialised tests for mathematics or chemistry.

2.1.4 Interoperability standards

Much current development in this area appears to make use of the APIs defined and implemented in particular VLEs. The IMS Learning Tool Interoperability (LTI) standard aims to play a role in this area (IMS Global Learning Consortium, 2011a), and though some implementations have been demonstrated in the JISC DVLE and FSD programs (MacNeill, 2011) the specification's final form has not been determined at the time of writing. In any case, the integration of tools into learning environments is by its nature something that is limited to the education sector.

2.1.5 Institutional scope

Unlike most other DLE models, and like the conventional VLE, there is almost no scope for participation of services and people from the outside.

2.1.6 Complexity

Plug-ins, in the strict sense illustrated in Fig. 3, are installed and deployed in the same network location as the VLE- inside the institutional firewall. This adds a degree of complexity, since each plugged-in *software component* needs to be maintained, and versions on both sides of the proprietary API *application interface* kept in sync. Version changes in any of the three or more components can cause a problem. There is a degree of extra complexity for users as well, since the plug-ins may behave a little differently from the VLE itself, and the presence of several tools itself creates more complexity.

2.3 One system, many outlets

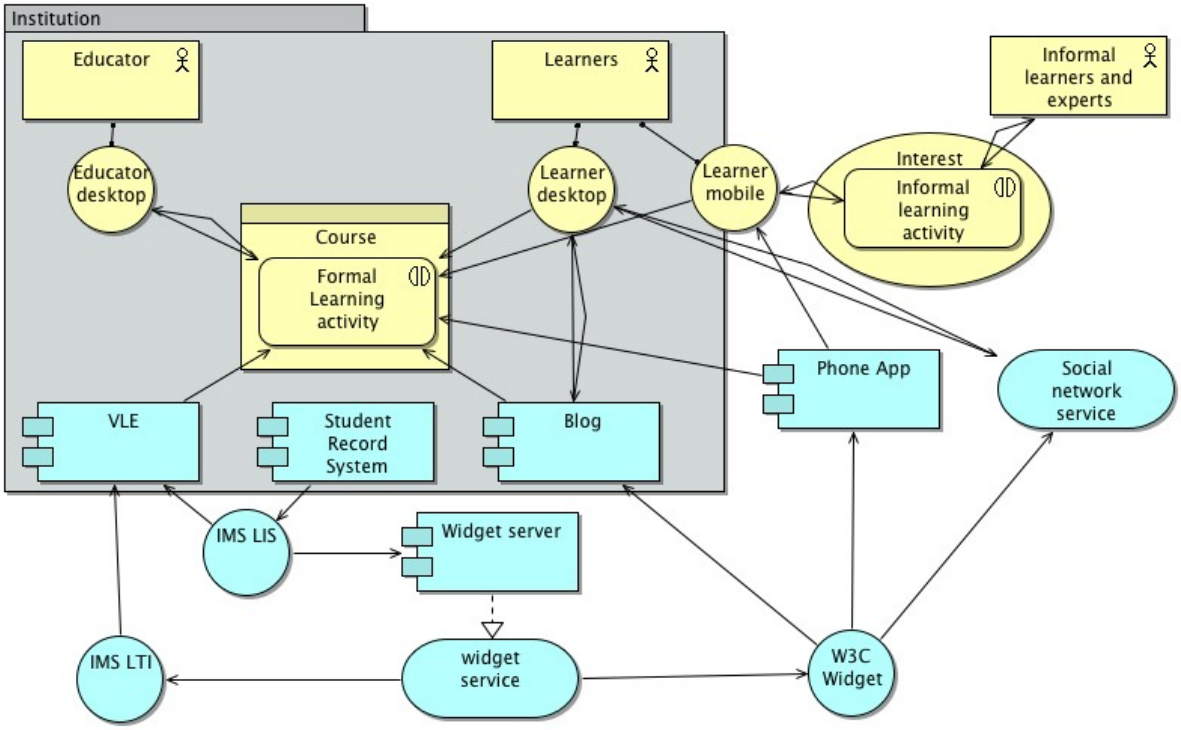

Fig. 4: One system, many outlets pattern

In the 'one system, many outlets' pattern there still is a central, source *application component* – the widget server –, but everything else is quite different from the previous patterns. The *application service* can connect via a number of *application interfaces* to a wide variety of *application components* and *services* both inside and without the institution. Fig. 4 shows the widget server outside the institution's perimeter, which is where it would be in cases such as the cloud based Plugjam platform (MacNeill, 2009), but it could also be installed inside the institution, in cases such as Apache Wookie (incubating) (The Apache Software Foundation, 2011)

2.3.1 Context, focus and organisation

In this pattern, the VLE in Fig. 4 can easily retain the course based structure it commonly has, as can the organised learning activities learners participate in. But there is no absolute need. The same resources and services have been made portable and can be integrated in *business collaborations* that are organised by interest or any other salient concept.

2.3.2 Relationship symmetricity

The case illustrated in Fig. 4 envisages a relatively passive relationship between the formal learning activity and the learner. This need not be the case – various *application components* other than a conventional VLE can accommodate a greater degree of control on the part of the learners. In either case, though, the portability of resource to the environment of the users' choosing means that there is already more of a balance in access and control.

2.3.3 Context experience

The area where the 'one system, many outlets' pattern differs most clearly from the dominant VLE pattern is the way in which users can adapt their environment to their own needs and preferences. This is a matter of degree, however; organisations can choose how much of their formal learning activities wander from the institutional VLE to the widget server. Also, the widgets the server makes available are designed to be embedded in any number of systems, but they are not designed to carry outside services into a particular institutional system.

2.3.4 Interoperability standards

An advantage of the 'one system, many outlets' pattern is that educationally oriented applications such as Wookie (The Apache Software Foundation, 2011) can be easily made to accommodate both educationally specific interoperability standards such as IMS LTI as well as more general web oriented ones such as the W3C Widget specification. Users therefore have the widest range of potential contexts to choose between. At the same time, since VLEs are web based, non-educational widget servers could be made to work as well, if not necessarily with as rich an integration (but see the case of Sakai Open Educational Environment below).

2.3.5 Institutional scope

It is possible to set up the pattern in such a way that both the widget server and all of the widget containers are controlled by the institution. For reasons of personalisability of the context and a balance of participation and control between educators, that could be less desirable, even if it is already quite some way removed from the homogeneity and imbalance of the classic VLE.

An interesting question of scope arises with regard to control over interactions that involve the placement of (institutional) widgets in external containers. If the widget integration is very shallow – via an HTML iFrame, for example – no issue arises other than the potential exposure of confidential communications by learners in, say, a chat widget. For the richer integrations that are enabled by specifications such as W3C Widget, the control over contributions from outside of the original context in which the widget was instantiated (such as a course) is a matter of the configuration of the widget server and the configuration of the container in which the widget is instantiated.

2.3.6 Complexity

The 'one system, many outlets' pattern is clearly more complex to set up and maintain than a conventional VLE. Fig. 4 has three software components, two services and many more relations to manage, for example. This need not be the sole responsibility of the institution, however. The management of the widget server can be outsourced, and any integrations users establish with systems outside the institutional context could be consider to be their responsibility.

2.4 Many providers

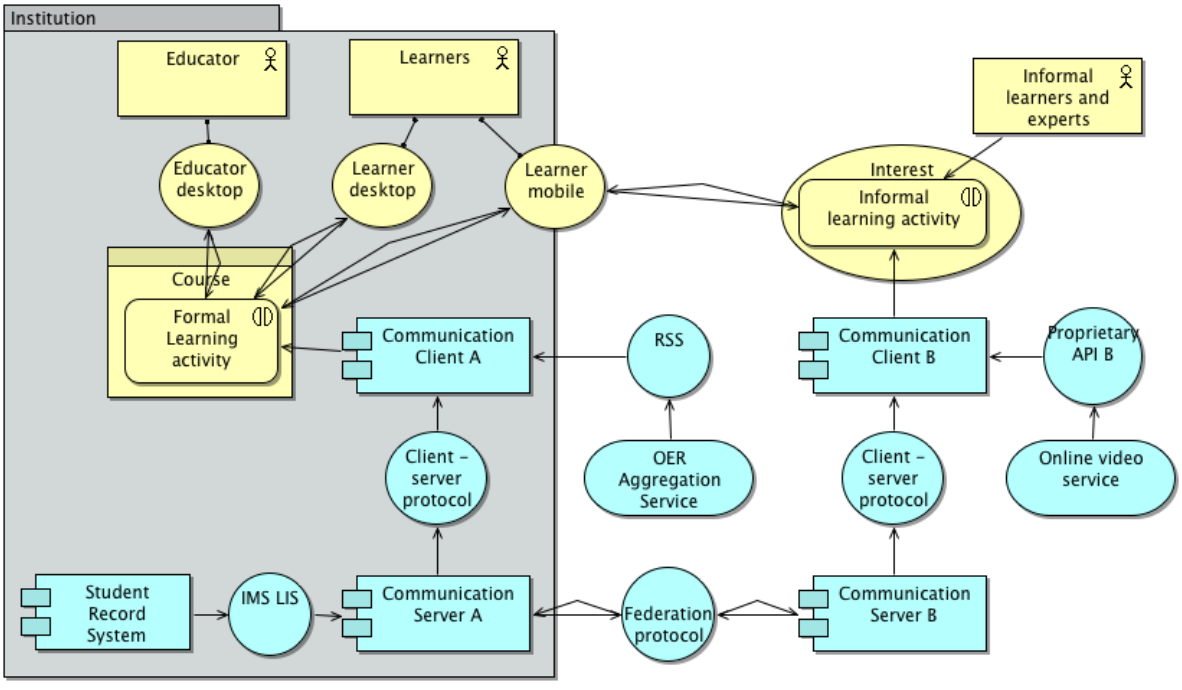

Fig. 5: The many providers learning environment pattern

This is a comparatively unusual pattern that is built around a federated client-server architecture. It's main characteristic is that sets of server and client *application components* are paired via a client-server protocol *application interface*, and that the pairs are connected to their peers via federation protocol *application interfaces*. In this sense, it operates much like email, except that other services can be integrated on the client via other *application interfaces* such as RSS. A recent example of such a system was Google Wave, which is now in the process of becoming an Apache Incubator project (Stenyak, 2011)

2.4.1 Context, focus and organisation

The federated aspect of the pattern allows a relatively fluid way of designing the context for a learning activity. Assuming that there is a conduit for institutional person and group data, like the one from the student record system in Fig. 5, it should be relatively easy to use courses as the main structuring device. The conversational nature of such federations means that adding people or resources from outside of that context can happen at any time, and by anyone.

2.4.2 Relationship symmetricity

Whether such free expansion of context can be controlled in any fashion remains to be seen. Both email and Wave have no controls to grant access controls to educators or administrators, which is not to say that future examples of the pattern can't either.

2.4.3 Context experience

The ability to connect *application services* to communication clients can provide considerable scope for tailoring the environment to specific subjects, groups or individuals. If the client-server protocol is open, users can also choose which client suits their needs best.

2.4.4 Interoperability standards

The pattern instance illustrated in Fig. 5 assumes that federated and client-server protocols are unlikely to be education specific. The overhead required to set up such protocols almost certainly needs to be spread over several domains. The IMS Learner Information Services *application interface* specification that provides people and group information to the server is educationally specific, but the same function could be taken by neutral specifications such as LDAP.

2.4.5 Institutional scope

As noted, the federated nature of the 'many providers' pattern means that a flexible balance can be struck between institutional control and external contacts. In both the case of Wave and email, communication between members remains private to the institution, even if the infrastructure allows communication with outsiders just as easily.

2.4.6 Complexity

The degree of complexity of the pattern largely depends on the reliance on integrating external services at the client level. Even if a large number are allowed, however, in disruption in their operation may well leave the main communication channel unaffected. Also, federated systems tend to be quite resilient out of their very nature: there is no single point of failure.

2.5 Both a provider and a consumer

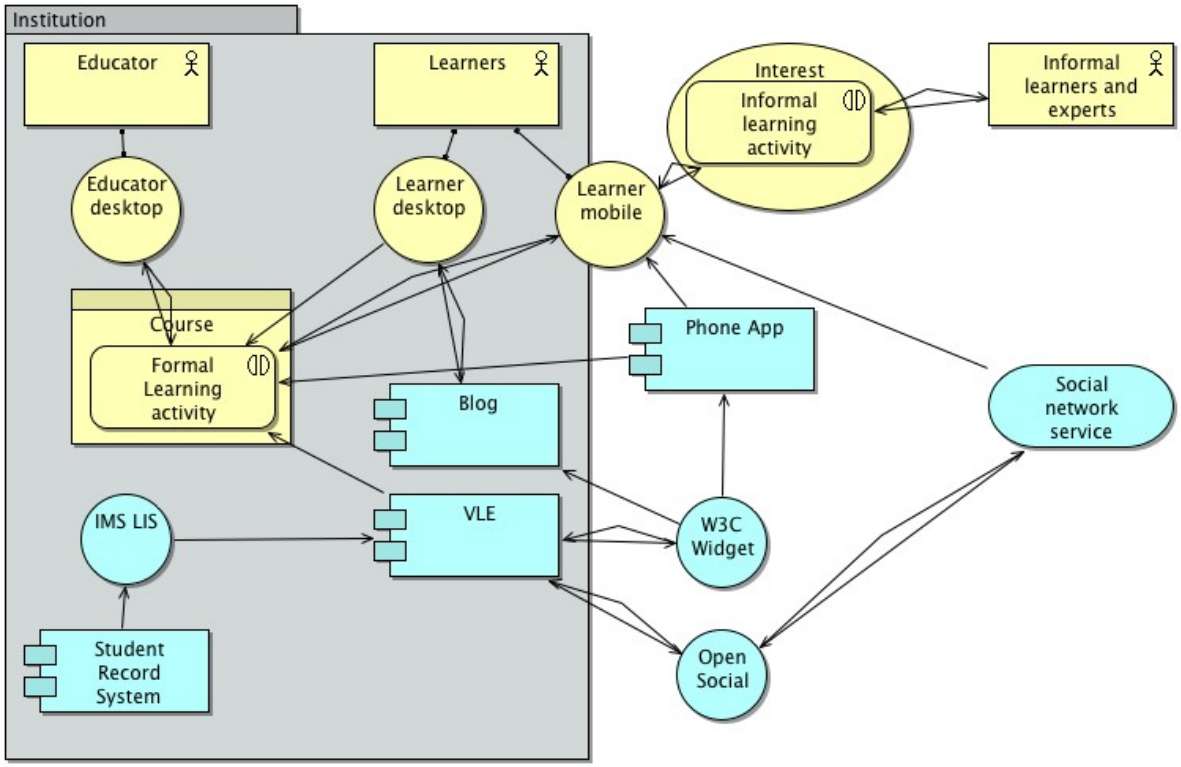

Fig. 6: Both a provider and a consumer learning environment pattern

So far, most patterns have had a clear distinction between one or more providing systems and one or more consuming applications. In this case, the VLE acts as both. A social network *application service* can be integrated into the VLE via the Open Social *application interface*, just as much as features of the VLE *application component* can be integrated into the Phone App *software component* via the W3C Widget *application interface*. A concrete example of the pattern is the developing Sakai Open Academic Environment (OAE) (Sakai Foundation, 2011)

2.5.1 Context, focus and organisation

The structure of the context depends on the conventions of the combined provider and consumer, but assuming that such a system is a VLE type of system, it is safe to assume that the default will be a course. Sakai OAE, however, aims to enable users to configure other contexts via the use of templates (Sakai Foundation, 2011). Suitably flexible generic content management systems (CMSs) such as Microsoft's SharePoint could also be adapted to support different types of context.

2.5.2 Relationship symmetricity

The dual role of the VLE in this pattern heightens the contrast between control explicitly granted, and control available. In other patterns, learners are able to participate on a more equal footing because they own, or at least control a piece of the learning environment's architecture. In this pattern, the potential for users to re-contextualise resources, services and interactions from the VLE in other contexts and vice versa is just as unlimited. But because the provision and consumption of these services and resources is determined by the VLE, administrators or suitably delegated educators are able to limit this aspect of the autonomy of learners quite considerably. For contexts such as junior schools, that may be an advantage.

2.5.3 Context experience

By its nature, the 'both a provider and a consumer' pattern allows maximal flexibility in the customisation and personalisation of the learning context. Like the 'one system, many outlets' pattern, there is the possibility to vary user interfaces independently of the source. Unlike that pattern, the 'both a provider and a consumer' pattern also allows all services and resources (including external ones) to be concentrated in a single, dedicated interface. This suggests a means of scaffolding both learners and educators while they acquire the digital skills that they will need to compose their own contexts.

2.1.4 Interoperability standards

Because the central provider / consumer is itself the dominant educational software application, the main interoperability standards it has to implement are generic ones such as OpenSocial and W3C Widget in order to

integrate with as many non-educational tools and services as possible. Though it isn't illustrated in Fig. 6 at the tool level, that emphasis on generic standards doesn't mean that educationally specific *application interfaces* are not useful as well, particularly for very educationally specific tools such as quiz engines.

2.1.5 Institutional scope

In this regard, it is interesting to note that Sakai OAE is said to be 'permeable': “Our students, teachers and researchers inhabit an academic world that extends beyond the institution. Our institutional platforms complement this fuller experience; they should not try to dominate it.” (Sakai Foundation, 2011) A lot of that promise relies on the implementation, but it is clear that the architecture of the 'both a provider and a consumer' pattern facilitates cross-institutional, interest or subject based interaction.

2.1.6 Complexity

From an institutional point of view, the combination of a source and its main destination would appear to increase flexibility relative to a classic VLE, without increasing the *application component* count too much. The example in Fig. 6 bears that out. The price, though, is a relatively high degree of complexity within the combined system itself. The VLE needs to not just provide a full feature set, but also make it portable, and substitutable by an equivalent from outside. This is further exacerbated by the number of interoperability standards that need to be supported, and the scope for version or profile mismatches that that entails.

2.6 Many widgets into one container

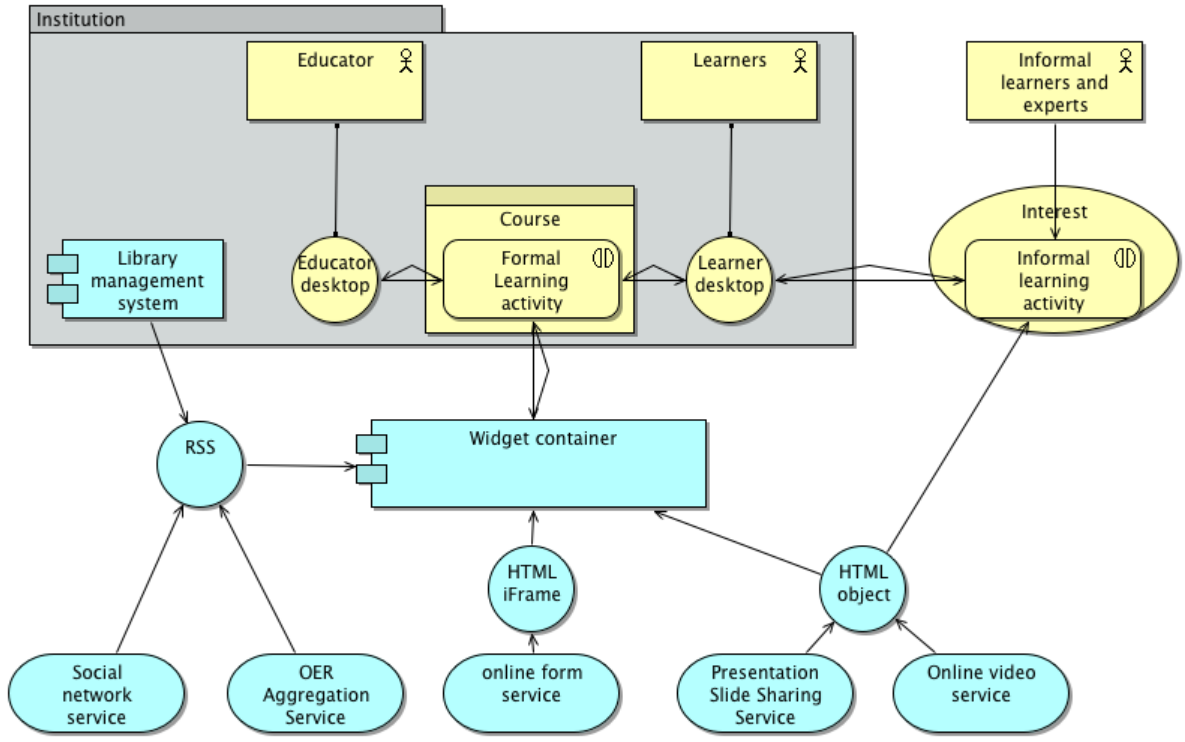

Fig. 7: The many widgets into one container learning environment pattern

In one sense, the many widgets into one container pattern completely reverses the VLE pattern: the environment is composed almost entirely of external services collected and presented in a thin, equally external container. While it ought to be possible to have a dedicated, institutionally controlled widget container *application component*, in practice only generic ones appear to exist. Toole (2009) reported on various examples of this pattern (cf. MacNeill, 2009), using services like wetpaint (wetpaint.com, inc., 2011) or ning (Ning, inc., 2011) as the widgetcontainer. General purpose social networks such as Facebook or Google + could also be suitable for this role.

2.6.1 Context, focus and organisation

Arguably the most defining characteristic of this pattern is the ease and speed with which it can be set up, and the generic flexibility of the tools. As a result, most any structuring device or social grouping can be accommodated; from the conventional course, to topics that span multiple courses and subject areas.

2.6.2 Relationship symmetricity

The ease of deployment gives all stakeholders scope to participate and control. Within the typical widget container, however, various fine grained access controls exist. An educator who sets up an environment like the one in Fig. 7 can therefore restrict the ability of learners to modify that particular context fairly precisely.

2.6.3 Context experience

The flexible, generic nature of widget containers can make it easy to set up contexts that are very finely tuned to a particular subject, group or individual, as Toole (2009) demonstrates. Resources, services and people can be quickly marshalled and added to sites dedicated to concerns that are sometimes long and persistent but can just as easily be very fleeting.

2.6.4 Interoperability standards

Since the functionality of 'thin' widget containers is quite limited, the integration technologies used tend to be quite basic, html based ones such as the *application interfaces* in Fig. 7. This can lead to problems in the integration of services or widgets that require authorisation or delegated authentication – appropriate interfaces often don't exist (Toole, 2009). Also, these containers have no way to be provisioned with people and group data from an institution, which means that a lot of the management convenience of a VLE is replaced with manual configuration.

2.6.5 Institutional scope

Since typical widget containers do not have a notion of institutions or groups, it is entirely up to the users to decide what the institutional scope of such an environment should be. This can be beneficial in its flexibility, but may further increase the administrative burden on educators.

That is not to say that there is no link with the institutional environment at all. It is often possible to embed services and resources from the institution in the widget container by using basic techniques such as RSS.

2.6.6 Complexity

From an institutional perspective, the 'many widgets, one widget container' pattern can appear to be the simplest of all: no system deployment or maintenance is necessary at all. On the other hand, the burden of maintaining both the people and widgets in a container can greatly increase the burden on educators, particularly those who don't have the requisite skills. Also, it will be difficult for institutions to control or manage what goes on inside these ad hoc environments.

2.7 PLE

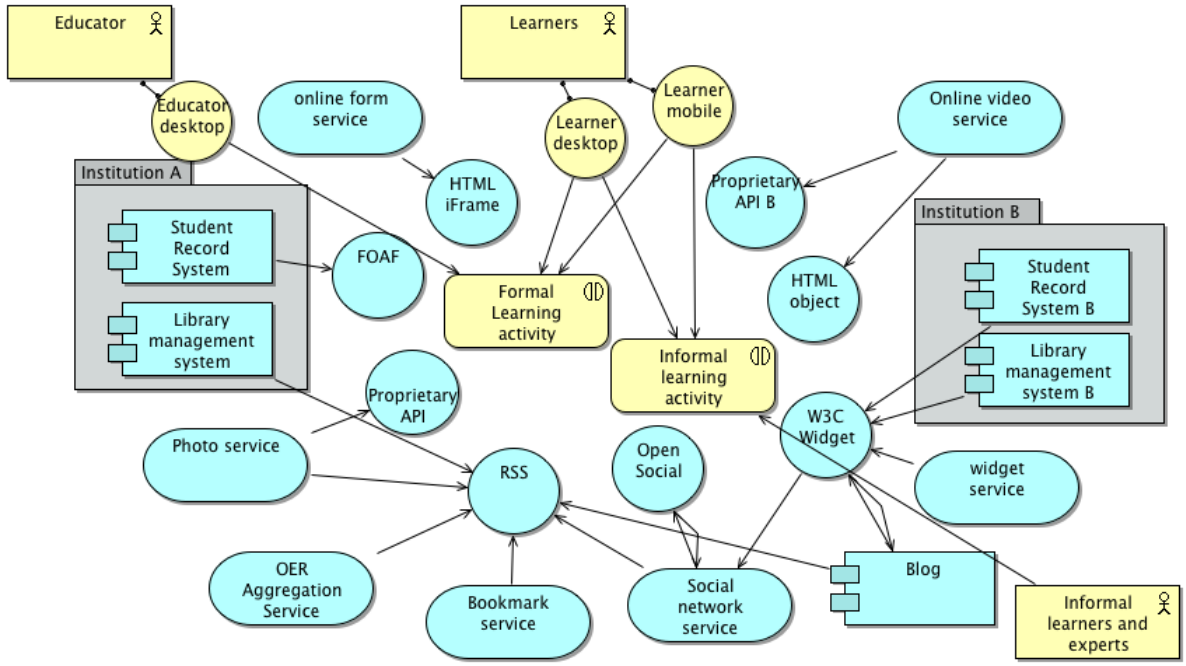

Fig. 8: The personal learning environment pattern

The PLE pattern is not very far from the previous, 'many widgets in one container' pattern. The chief difference is that the PLE pattern doesn't specify the final user interface- just the services that can be aggregated in some way by the learner.

2.7.1 Context, focus and organisation

The PLE pattern takes as its central organising principle the full length and width of the life of the learner. Courses and even institutions become just one among many sources of services and interactions. Put differently, the various entities and relations that comprise the PLE are reconfigured to accommodate any given learning activity, and the focus that it may have.

2.7.2 Relationship symmetricity

The PLE's focus on the learner's life means that the question of control is radically re-conceived (Wilson et al., 2006). In this pattern it is the learner who ultimately determines what to engage with, and under what conditions. Since control from that perspective is at a rather different level than the control offered by a conventional VLE, this need not mean that able PLE users will have unchallenged power. It may well be that a long term goal makes the trade-off of losing control over one aspect of learning in exchange for structure and accreditation worthwhile.

2.7.3 Context experience

Since the precise make-up of the user interface to all the services that make up a PLE is up to the learner, the resulting context can be highly personalised, rich, dynamic but also quite fragmented. That need not be a problem for an experiences and sophisticated learner, but others might struggle.

2.7.4 Interoperability standards

Given the lifelong and lifewide nature of the focus in a PLE, the interoperability standards used are common web ones, because they can connect to the greatest variety of generic web services as easily as they can connect to multiple formal institutions. Managing these interfaces may require a degree of technical ability on the part of the leaner, though.

2.7.5 Institutional scope

The switch to the learner perspective does not necessarily mean the irrelevance of other viewpoints. It just makes clear that learning activities – as well as the tools used to support them – are part of a much larger environment where they interact with other resources in ways that are well out of the control of an institution. The management of such a large part of the learning environment alone, without direct support from an institution, does require a relatively sophisticated learner, however.

2.7.6 Complexity

It proved difficult to stay faithful to the original PLE sketch (Wilson et al., 2006) and maintain strict comparability between the various patterns. For example, inclusion of the *business objects* that are accessed by the learning activities and thereby structure them, was clearly making Fig. 8 even more complex and difficult to read than it already is. In that regard, the diagram may well accurately reflect reality since composing and maintaining one's own learning environment out of many disparate services and devices is a potentially complex and confusing task that requires considerable skill.

3 Conclusion

By carefully composing an appropriate metamodel, ArchiMate can be useful in analysing learning environment architectures, pinpointing differences and communicating them. In the learning environment patterns examined here, each salient characteristic found a fairly consistent indicator in the ArchiMate views.

That is, the course *business object* and interest *value* to represent the major organising principles or focusses behind leaning activity *business interactions* provides a simple indicator for the context characteristic of a particular pattern. For the relationship aspect, the simple reciprocality of relationship arrows provides a similar cue. For the context experience, the variety and nature of connected external *application services* and *components* in the view is a good indicator. Interoperability standards are readily identified by *application interface* icons, and their domain specificity by what they connect to. The *group* entity indicates at-a-glance whether *application components* are within institutional boundaries or not. Finally, the complexity of the diagram itself appears to correlate closely to the complexity of an architecture in the real world.

With regard to the leaning environment architecture patterns themselves, it is clear that an increasing variety of architectures are becoming possible. Some of these patterns near the PLE end of the scale are still quite immature, but that's just one of the characteristics that stakeholders can take into consideration when choosing. This variety of patterns also means that more people are likely be able to deviate – however weakly – from the currently dominant VLE pattern, into environments that are better adapted to subject, group and individual needs and preferences, and environments that are more equatable, collaborative and open to participation by all stakeholders.

4 References

Abernathy, W. J., & Utterback, J. M. (1978). Patterns of Industrial Innovation. *Technology Review*, *80*(7), 40-47.

Advanced Distributed Learning. (2011). SCORM. *Advanced Distributed Learning Initiative*. Retrieved September 2, 2011, from http://www.adlnet.gov/capabilities/scorm

Bae, H., Han, K.-W., Lee, E., & Lee, Y. (2008). The Cyber Home Learning System for Primary and Secondary School Students in Korea: Current State and Challenges. *Society for Information Technology & Teacher Education International Conference 2008*, *2008*(1), 2887-2891.

Buchem, I., Attwell, G., & Torres, R. (2011). Understanding Personal Learning Environments: Literature review and synthesis through the Activity Theory lens. *The PLE Conference 2011* (pp. 1–33). Retrieved from http://journal.webscience.org/658/

Cáceres, M. (Ed.). (2011, August 11). Widget Packaging and XML Configuration. W3C. Retrieved from http://www.w3.org/TR/widgets/

Iacob, M., Jonkers, H., Lankhorst, M., & Proper, H. (2009). *ArchiMate 1.0 Specification*. The Open Group.

Icodeon Ltd. (2009, July). Icodeon SCORM Player Version 2 (Java Edition). Icodeon Ltd. Retrieved from http://www.icodeon.com/release/download/j2ee/release_2p0p4/docs/index.html#d4e269

IMS Global Learning Consortium. (2011a, January 3). Tools Interoperability Guidelines. *IMS Global Learning Consortium*. Retrieved a from http://www.imsglobal.org/ti/index.html

IMS Global Learning Consortium. (2011b, March 1). Learning Information Servces. *IMS Global Learning Consortium*. Standards document, . Retrieved August 30, 2011, b from http://www.imsglobal.org/lis/index.html

JISC. (2010, July 1). Distributed VLE programme. *JISC; inspiring innovation*. Retrieved August 29, 2011, from http://www.jisc.ac.uk/whatwedo/programmes/elearning/distributedvle.aspx

JISC Infonet. (2011). Enterprise Architecture Practice Group;Doing Enterprise Architecture - together. *JISC Infonet; good practice and innovation*. Retrieved from http://www.jiscinfonet.ac.uk/flexible-service-delivery/ea/eapg

Lankhorst, M. (Ed.). (2004). ArchiMate Language Primer–Introduction to the ArchiMate Modelling Language for Enterprise Architecture. Version 1.0. Telematica Instituut. Retrieved from https://doc.telin.nl/dscgi/ds.py/Get/File-43840/

MacNeill, S. (2009, November 3). The headless VLE (and other approaches to composing learning environments). *Sheila's work blog*. Retrieved August 30, 2011, from

http://blogs.cetis.ac.uk/sheilamacneill/2009/11/13/the-headless-vle-and-other-approaches-to-composing-learning-environments/

MacNeill, S. (2011, July 11). “Win, win, win” Are we really there with IMS LTI and LIS? *Sheila’s work blog*. Retrieved August 29, 2011, from http://blogs.cetis.ac.uk/sheilamacneill/2011/07/08/win-win-win-are-we-really-there-with-ims-lti-and-lis/

MacNeill, S., & Kraan, W. (2010, February 1). Distributed Learning Environments briefing paper. JISC - CETIS. Retrieved from http://wiki.cetis.ac.uk/images/6/6c/Distributed_Learning.pdf

Ning, inc. (2011). Create Your Own Social Website. *The World’s Largest Platform for Creating Social Websites™*. Social networking, . Retrieved August 30, 2011, from http://www.ning.com/

OpenSocial and Gadgets Specification Group. (2011, August). OpenSocial Specification 2.0. googlecode.com. Retrieved from http://opensocial-resources.googlecode.com/svn/spec/2.0/OpenSocial-Specification.xml

Sakai Foundation. (2011). About Sakai OAE. *Sakai Project*. Project homepage, . Retrieved August 30, 2011, from http://www.sakaiproject.org/node/2239

Savage, M., & Erskine, J. (2009, June 1). Presented at the Staffordshire University Learning & Teaching Conference, Staffordshire University. Retrieved from http://www.slideshare.net/jwe1/su-l-t-conference-june-2009-final

Stenyak, B. (2011, April 27). Crash course in Wave and Apache Wave. *porque a veces uno se aburre… stenyak.com*. Blog, . Retrieved August 30, 2011, from http://www.stenyak.com/archives/887/crash-course-in-wave-and-apache-wave/

The Apache Software Foundation. (2011). Apache Wookie (Incubating). *Wookie*. Project homepage, . Retrieved August 30, 2011, from http://incubator.apache.org/wookie/

The Open Group. (2009). *TOGAF Version 9*. Van Haren Pub.

Toole, T. (2009, November 3). *Jisc Cetis Conference09*. Presented at the JISC CETIS Conference 09, Birmingham. Retrieved from http://www.slideshare.net/ttoole/jisc-cetis-conference09

wetpaint.com, inc. (2011). Wetpaint Central Help and Support. *Wetpaint central*. Social networking, . Retrieved August 30, 2011, from http://www.wetpaintcentral.com/

Wilson, S., Liber, O., Johnson, M., Beauvoir, P., Sharples, P., Milligan, C., & others. (2006). Personal Learning Environments: Challenging the dominant design of educational systems. *on Professional Learning, Competence Development and Knowledge Management-LOKMOL and L3NCD*, 67.