Journal of Universal Computer Science, vol. 18, no. 11 (2012), 1432-1453
submitted: 30/11/11, accepted: 15/5/12, appeared: 1/6/12 © J.UCS

The Wookie Widget Server: a Case Study of Piecemeal Integration of Tools and Services

David Griffiths
(The Institute for Educational Cybernetics, University of Bolton, Bolton, UK
d.e.griffiths@bolton.ac.uk)

Mark Johnson
(The Institute for Educational Cybernetics, University of Bolton, Bolton, UK
M.W.Johnson@bolton.ac.uk)

Kris Popat
(The Institute for Educational Cybernetics, University of Bolton, Bolton, UK
K.J.Popat@bolton.ac.uk)

Paul Sharples
(The Institute for Educational Cybernetics, University of Bolton, Bolton, UK
P.Sharple@bolton.ac.uk)

Scott Wilson
(The Institute for Educational Cybernetics, University of Bolton, Bolton, UK
S.Wilson@bolton.ac.uk)

Abstract: Apache Wookie (incubating) has generated considerable interest within the context of Technology Enhanced Learning where it was developed, as well as in mobile applications. The origins of the system in providing services for IMS Learning Design are described, together with an introduction to the system's design and functionality. However, the areas where it has had success are distinct from the application area for which it was designed and developed. The implications of this for understanding user needs is analysed by using ideas drawn from sociology. The complexity of the relationship between the context of use and user needs, and the feedback loops between them is discussed, and the role of technological interventions as an element in a discourse is considered. It is proposed that this understanding of users needs, together with the experience of the development and use of Wookie, argues in favour of an interoperability strategy which focuses on relatively small sets of functional requirements, and avoidance where possible of specifications developed for particular application domains: an approach which may be characterised as piecemeal rather than Utopian.

Keywords: Interoperability, Wookie, Widget, Specification, W3C, Apache, IMS Learning Design, Mobile, Mash-up, personal learning environment
Categories: H.1.2, H.3.3, H.4.1, H.4.2

1 Introduction

This paper tells a story which appears to be simple: the development of a server for W3C widgets which gained sufficient traction to be accepted into the Apache

Foundation incubator [Apache Foundation, 2011]. Nevertheless it has not been an easy paper to write. The development of Wookie has generated many different perspectives stimulating new and contrasting applications, which are remote from the use cases which led to its development. Some of these aspects have been described [see Wilson 2009, 2011] in earlier papers (although many have not) but to date we have not documented or analysed the whole process. Nor have we drawn conclusions from this successful development project which can inform development strategy.

In this paper we seek to make good this lack, with a focus on the technical challenges, and the solutions developed in response to them. We do not discuss in detail the degree to which the systems met the needs foreseen for the users, a topic which we hope to address in a future paper. We do, however, show how the development of the technological solution to the social problem of orchestrating learning activities has changed our understanding of that problem, and brought about new lines of research and the development of new practices. These are not restricted to the initial problem within Technology Enhanced Learning (TEL), but extend to the wider area of delivering and combining services across platforms and applications, including mobile phones. We draw on the work of Luhmann to understand the limits on the prediction of the use of technology. We show how the use of, and contribution to, open specifications, together with the development of open source reference implementations, were not simply a practical measure to improve the effectiveness of our technological intervention, but were rather an enabling component of the interaction between the technological and the social which constitute the design space for applications.

2 The origins of Wookie in the provision of services for IMS Learning Design

In this section of the paper we describe the problem which Wookie was initially designed to resolve: the provision of services for IMS Learning Design (LD) [I M S Global Learning Inc, 2003], and the system which was developed to meet this need.

2.1 The goals of IMS Learning Design

A common sense view of learning tells us that some learning activities are better than others, and it seems reasonable that we should be able to identify those activities which are effective, describe them, and provide guidance to instructional designers which will enable them to create optimal courses. The approach is set out by Koper in [Koper, 2005a]. Building on Reigeluth [Reigeluth, 1999] he describes how "...learning design knowledge consists of a set of prescriptive rules with the following basic structure: if learning situation S, then use learning method M, with probability P." Koper led development of Educational Modeling Language (EML) at the Open University of the Netherlands (OUNL) [Koper, 2004a] in which to express learning designs and so make it possible to propose explicit and reproducible learning design rules. The language was formal, and expressed in XML, so that it could be made use of by computer systems in order to orchestrate the interactions of teachers and learners. In order to promote adoption of the specification, and the exchange of Units of Learning, the Educational Modeling Language was later adapted to create IMS LD.

An open source application, Coppercore [Vogten, 2007] was developed to interpret Units of Learning, and to orchestrate the interactions of users. At this point it became clear that interoperable learning designs presented a problem: An implemented rule has to be both context free (abstracted, so it can be run repeatedly) and context specific (so that it can make use of the services which are available to each individual user). How could this be achieved?

2.2 Approaches to service integration in IMS LD

While Koper's exposition of learning design rules is not informed by a particular approach to computer support, the way in which it was instantiated in IMS LD bears the hallmarks of the period in which it appeared. EML, and more explicitly IMS LD, were interoperability specifications, and the reasonably enough the assumption seems to have been made that the systems which they were making interoperable were Virtual Learning Environments (VLEs). Indeed, given the TEL technology of the day, what other position could have been taken? Thus the Edubox player, commissioned by OUNL to implement EML, conformed quite closely to the pattern of a VLE [Tattersall, 2005]. When IMS LD was published there was a concerted effort by the Valkenburg Group to publish a reference architecture [Koper, 2005b], and to provide the components which it specified. The CopperCore engine, mentioned above, was the reference implementation for IMS LD runtime, instantiating Units of Learning (UOLs), and keeping track of the progress of learners through them.

The first attempts at resolving the problem of providing services for LD runtime stayed close to the tight integration of specific services which is the dominant model in VLEs. The Valkenburg Architecture envisaged the integration of services through the CopperCore Service Integration (CCSI) layer [Vogten, 2006], and this was the point of departure for the work carried out by the team at the Institute for Educational Cybernetics (IEC), University of Bolton, which is forms the narrative of this paper. Early work was carried out by the Open University in the UK, using CCSI to make available a Moodle forum though CopperCore, and by OUNL to integrate a QTI player. The authors contribution to this effort was an integration of the Reload SCORM player [Griffiths, 2007] within the TENCompetence project. This clarified the problems and complexity posed by connecting a new service to an IMS LD runtime system, and we concluded that:

> The CopperCore Service Interface layer provides the necessary framework, but actually implementing a new service requires knowledge of this specific API, and not a defined and agreed standard. The framework also is quite open, allowing the developer to write his/her own calls between LD Engine and new service. While this allows a large amount of freedom to the developer, it is also very abstract. … The effort involved, however, suggests that it is not a solution which can meet the need for the agile integration of a large number of services into IMS LD. [Griffiths, 2008]

This report continues to describe how IMS LTI was rejected as an alternative because it required a range of SOAP Web Services, and so did not resolve the problem of agile integration. Instead it was decided to extend the emerging W3C Widget Protocol. This supported a much simpler and more lightweight set of requirements for collaboration services, and also enabled existing widgets to be ported to operate within an LD environment with relative ease. The result was the development of the

Wookie widget server, which thus had its origins as a technical solution to an implementation problem.

2.3 The widget based approach to services in IMS LD

The new widget based approach to services for IMS LD was outlined in the paper *Distributing education services to personal and institutional systems using Widgets* [Wilson, 2008]. As the title indicates, the decision was taken at this early stage in development to seek a generic solution to the specific problem of service deliver in IMS LD. This paper documents the initial architecture of the system, and describes how *container applications* must support the Widget Configuration API, and be capable of rendering a view that loads the resulting URL. For each container application a *widget plugin* is required, enabling a widget to be placed within the context of the container application, and implementing the widget configuration API. The widget plugin collects the relevant configuration information via the container application's authoring system, and instantiates a widget using the getWidget() method.

The Wookie Widget Server was developed as a new standalone cross-platform application which supplied widgets to third parties as they were requested (such as CopperCore in this case). The container application provides the *context* for use, and determines who is authorised to access the widget instance. The server has a defined API for communication between client and server, using a combination of Javascript and AJAX16 This API was based on the W3C Widget API draft specification [W3C, 2011b], and in a later phase of work integration between Wookie and Apache Shindig enabled Wookie to serve OpenSocial Applications. Wookie provides an administration interface with which to manage the publication of widgets, and gallery for selection of widgets for inclusion in the platform which consumes the service.

Wookie provided a means of delivering generic services, but to resolve the problem of delivering services to IMS LD it was necessary to define a means of specifying the services which were to be called by a particular UOL. The IMS Learning Design specification describes only the following four services (as well as an additional Learning Object type): Conference; Send Mail; Monitor; Index Search. These services, therefore, need to be extended in some way. A solution to this was formulated where existing elements and attributes of the IMS Learning XML binding could be used to support these specific services (such as chat for example). Within the XML binding of each service there is the ability to allow a "parameter" value to be specified. This can be any text an author may wish, but it is usually used in the form of name-value-pairs. To use one of the widgets made available from the widget server, a parameter is added to an existing service element within an environment. The name-value pair string to enter takes the following syntax, **widget=<type of widget>**. So for example, to use the default chat widget service, one would enter **widget=chat**. Similarly to use the default forum widget, one would enter **widget=forum**.

In order for that widget to be used by an IMS LD player, or other third party software, it must be associated with a label of some sort. In the example of the threaded discussion forum widget, one might associate the label *forum* with that widget. One also might want to associate the label *discussion* with it. The Widget Server allows a user to associate several contexts to one particular widget. When the

IMS Learning Design is authored, one can enter the widget=forum, or widget=discussion within the parameter attribute of the service, in order to associate the service with the specific widget residing within the Widget Server. A Widget *default* was defined, because more then one widget in the system may have the same context or label. In this way if the *environment* requires a chat widget but does not specify a label which matches any of those available on Wookie, the server can resolve the uncertainty by providing its default chat widget.

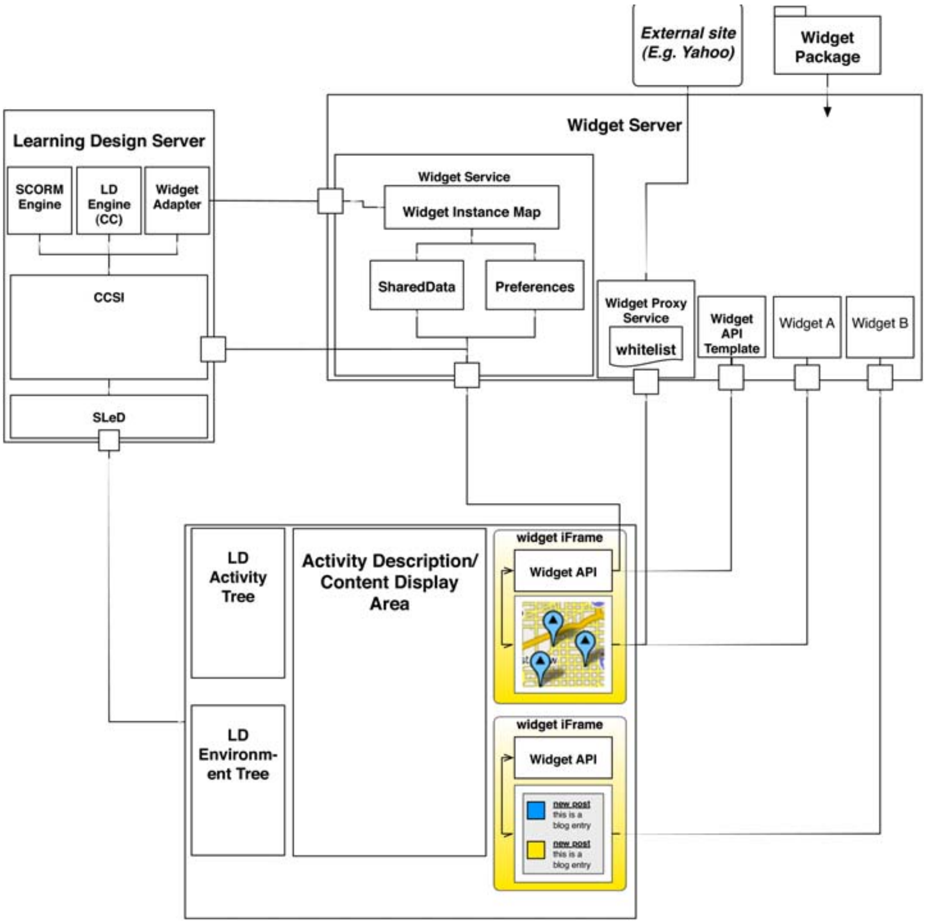

Figure 1: Integration of Wookie in the wider TENCompetence IMS LD infrastructure (taken from TENCompetnece deliverable D6.1[Griffiths, 2008])

The W3C Widget API allows a widget to interact with the Widget Server, using Javascript. The widget is able to set and get values, such as preferences. More importantly, it is also has access to shared data. An example of this is a chat log between chat users, which also raises the issue that when a message is sent by one user, the other users in the chat session must also see that message. Functionality such as this created the need to include callback events.

The user navigates in the Unit of Learning by using a player (in the case of the TENCompetence demonstrator this was SleD [McAndrew, 2005]). When a service is encountered SLeD obtains the environment information from the CopperCore engine, and parses each service entry found, to see if it contains any widget=context entries. If one is found, then the SLeD player builds a query. The query contains information which is specific to the run, environment, service and user who is requesting the widget; from within the unit of learning. The Widget Server uses these parameters to either return an existing widget instance (an instance which has been used before) or create a new widget instance. The CCSI widget module processes the query between the Widget Server and SLeD. Once SLeD has the response it can translate this information into the user interface. For example, returned values contain the URL of where the widget can be found and the widgets height and width to be displayed. Ultimately, when the user clicks on the widget link in the browser, a pop up window appears containing the widget content. Collaborative widgets were initially supported through an internal specification shared states, but this was later replaced with an implementation of the Google Wave Gadget API [Wilson, 2009a].

2.4 Defining the specific services to be provided to learners

As described above, an implemented learning design rule has to be both context free (abstracted, so it can be run repeatedly) and context specific (so that it can make use of the services which are available to each individual user). There are many trade-offs which can be made between these two imperatives, and the authoring process for Wookie services in UOLs illustrates the choices made in the Wookie architecture. A plug-in was developed for the ReCourse Learning Design Editor [Griffiths, 2009] which enabled the author to incorporate widgets into a UOL. This interrogates a specific Wookie server, which returns a list of the widgets it provides. These are shown on a palette in the *environments* editor of ReCourse, making them available for inclusion as services in the UOL. When a widget is specified in a UOL, the “parameters” attribute of an IMS LD service is allocated a string such as “widget=chat”. This indicates to the server at runtime that the given service requires an instantiation of a particular widget.

The Wookie server is distributed together with a default set of services and bundled with CopperCore. The administrator of a server can add new services, and designate new defaults. Taking the example of “widget=chat” above, authors can be confident that a Wookie server will have a chat widget of some sort available, because there is a default chat widget included in the distribution. They cannot know, however, if the administrator of the server will have changed the default chat widget for another one which they consider to be superior. Similarly there is no guarantee that the administrator will not have deleted the default services. Thus there is a responsibility on the deployer of the UOL to check that the required services are indeed available on the designated Wookie server. It is, however, an easy matter to upload any missing widgets to a Wookie server, or to change the address of the designated Wookie server for a UOL so that it is directed to a server where all the required widgets are available. Thus, if an author makes use of an entirely new widget service, they will have to indicate where the administrator of a Wookie server can find the widget so that it can be deployed. The result is a system that enables authors to make available selected tools to selected groups of users in a complex flow, and to provision these tools simply by assigning users to roles in the UOL. The lack of this functionality had previously meant that most UOLs focused on adaptive delivery of content, but it was now possible to author UOLs where content took second place to services (for an example see the debate template in [Griffiths, 2008]).

3 The difficulties inherent in planning technological interventions

In section 2 we have described two technological interventions. One, IMS LD and its associated software, was developed as the result of sustained theoretical intervention and received funding which within the context of TEL was relatively substantial. The resulting technology has not been adopted. The other, Wookie, was a technical solution to an implementation problem with no explicit justificatory theory, which was funded by a single strand of work within the TENCompetence project. It has achieved a degree of adoption which is unusual within TEL research. In section 3.1 we summarise the evidence suggesting that the implementation work done on IMS LD was not substandard. In section 3.2 we briefly outline the ideas from sociology which have provided us with a way of understanding our contrasting experience in working on these two areas.

3.1 Limited use of Wookie in IMS LD

The IMS Learning Design specification was developed by leading experts in the field of TEL, following extensive prior work at the Open University of the Netherlands in the design and testing of its forerunner, the EML specification. A concerted effort was made to develop tools to design and run UOLs, with development effort being provided by a number of institutions, much of it funded by the European Commission. IMS LD was developed with distance education in mind, but it excited considerable interest in the wider educational community [Burgos, 2005], [Griffiths, 2005]. Because IMS LD could describe activities carried out by people in roles, rather than simply deliver content, it seemed a much richer pedagogical environment to work in than that which was offered by other interoperability specifications or VLEs. This led to efforts to implement specification in face-to-face and blended environments.

The systems developed were trialled with promising results, including those for the systems described here [Hazlewood, 2008], and [Georgiev, 2009]. Participative design methodologies were used in the design of authoring tools [Griffiths, 2009], [Hernández-Leo, 2006], and [Neumann, 2009]. We have also shown that the underlying conceptual structure of IMS LD is not an insurmountable barrier for teachers [Derntl, 2012]. Nevertheless, it is now clear that IMS LD has not fulfilled the hopes which were placed in it. While it remains the focus of interesting research activities, it has failed in its primary goal of providing an interoperability specification for learning activities which would enable them to be transferred between Learning Management Systems. One might expect that Wookie would have met the same fate, and remained interesting but largely unused code. We now consider why this was not the case, and the lessons which can be drawn from this.

3.2 Problems in the design of technology

The lack of adoption of IMS LD is not exceptional in technological development, especially technology which seeks to establish a new paradigm. For example work on Google Wave, which was launched to great fanfare in 2009, was abandoned by Google in 2010 because it "has not seen the user adoption we would have liked" [Barnett, 2010]. Clearly the design of applications which fall outside the mainstream is a risky and unpredictable business. Indeed in some ways it seems to have more in common with fashion design than it does with engineering, as witnessed by the importance given by Apple to design factors and to the "coolness" of their highly successful consumer technology. We now consider some perspectives on the nature of technology development which can help throw light on this unpredictability, and on the design and adoption of Wookie.

It is usual to describe the evolution of technical systems as a process of gathering user needs, followed by analysis of requirements and implementation, see for example [Maguire, 2002], sometimes with iterations of the process. This methodology assumes that users needs exist beyond the technical environment in which they are manifested. In contrast, our experience has been that the technology itself has an historically determined logic which constrains the design space of viable new designs. Moreover, the introduction of a new system presents affordances for use which stimulate users to imagine new possible ways of acting with technology. This then leads to the identification of new needs which would not have existed without the presence of the technology. Collaborative design methodologies have been developed in order to empower users (or, more strongly, humanity) to control the technological environment in which they find themselves, rather than being forced to adapt to it. For example

> Democratizing design is necessary because users' needs are highly heterogeneous in many fields and therefore cannot be anticipated by designers; users' expertise and talent also is widely distributed. … we need to create socio-technical environments … that allow people to acquire the technical knowledge and skills necessary to use them and adapt them to their needs [Fischer, 2011].

We sympathise with this position, and indeed make use of iterative participative design methodologies in our own development work. However, while we do not question its usefulness as a design tool, this approach has had limited success in resolving the social problem which it seeks to address, i.e. that systems are produced which are not adopted by the people for whom they are designed.

In understanding the relationship of users to technology and the society in which they belong we find Luhmann's analysis useful (see [Qvortrup, 2005] for an introduction). He proposes that in order to make sense of the world, we should view societies and their institutions not as conglomerates of human organisms, but as systems of communications, which may be mediated by technology. From this perspective there is no opposition between a society, and the technology (including both methods and artefacts) which it makes use of. Rather a technological intervention in communicative processes is itself a social intervention. Viewed in this light computer application development is neither technology led, nor user led. Instead, technological artefacts may be seen as interventions in the wider social process which leads to the identification of user needs, among many other consequences. For example, the launch of Google Wave and Wookie stimulated discussion of the functionality available in Moodle [Moodle, 2009]. While this may help us understand more clearly the processes involved when we introduce new technologies, it does nothing to help us control them. Rather it does the reverse, because it precludes us from trying to resolve an equation with a human group on the one side and the technology to be controlled on the other. Luhmann himself warns of the complexity involved when he discusses the concept of risk as it applies to high technology, pointing to:

> ...the growth in causal complexity, i.e., the multiplicity and variety of causes and side-effects woven into the actually desired events. This includes not least of all 'human factors' with their notorious unreliability. What actually happens is then determined by selection processes, which accept situational conditions that become relevant in ways that can neither be built into a model nor be predicted. [Luhmann, 2008] p.81.

Kelly provides a less rigorous but simpler formulation of the situation in his description of a *technium* composed of artefacts, methods and concepts (although we can imagine that Luhmann would balk at the implied ascription of agency to technology): “At some point in its evolution our system of tools and machines became so dense in feedback loops and complex interactions that it spawned a bit of independence. It began to exercise some autonomy” [Kelly, 2010].

The panorama is complicated further still when one takes into consideration individual practice. Within sociology the relationship between individual practice and social context (or social structure) has long been discussed, with descriptions ranging from Giddens’s Structuration theory [Giddens, 1984] to Bhaskar’s Transformational Model of Social Activity [Bhaskar, 1975]. These models make clear that social structure conditions individual practice, but is in turn transformed by that practice as it evolves. The complex feedback loops which this interaction creates explain why it is hard to make predictions about either individual practice or social structure, and provides an additional source of uncertainty for the designers of technology. This analsysis does not in itself help us identify the specific factors which prevented adoption of IMS LD, or promoted the use of Wookie. It does, however, indicate a place where we could place our attention: as argued in [Johnson, 2009], if a new technological practice does not transform the social context, then the practice tends not to be sustained. We now consider how the development of Wookie constituted an intervention in the context of Technology Enhanced Learning where it was developed.

4 Wookie as an intervention in the learning design discourse

In the light of the warnings in the previous section we do not here attempt the major task of passing judgement on the causes of the limited adoption of IMS LD, although we have previously discussed this through the lens of positioning theory [Johnson, 2009] [Johnson, 2011] and are continuing efforts to clarify the question. However, it is worth discussing one aspect which is relevant to Wookie, identified in [Neumann, 2010]: the linear structure of the IMS LD behavioural model, which leads to a lack of runtime flexibility. This is a particular problem with regard to face-to-face and blended learning, where the teacher often needs to respond to the emerging dynamics of the class by adjusting the learning activities being carried out and the services which they use. Work has been carried out to try to address some of these problems, for example [de la Fuente, 2007]. More radical alternative approaches to implementing IMS LD are also possible, as Koper states “Whether this model is implemented in XML, RDFSchema, OWL,Topic Maps etc. depends which tools and technologies are used at any moment in time” [Koper, 2004b]. Nevertheless, the fact remains that all existing IMS LD players provide and activity tree through which teachers and learners can navigate, and in teacher led environments this inevitably leads on occasion to a conflict between the teacher's desire to mould emerging interactions and the limited set of possible routes available.

It was in this context that Wookie was introduced as an element of the available IMS LD infrastructure. While it was, as we have discussed, motivated by technical requirements, it also took on significance by being an example of another possible approach to supporting teachers. This sparked off conversations about technology and its use in education and in wider society, and provided an on-going focus for those conversations. [Rennison, 1995] proposes a system whereby a progression through spaces in a designed data visualisation system constitutes a visual discourse, citing Lakoff “Via this metaphor, redirecting the mind from one topic to a more important one is understood in terms of pointing from the current locus of visual attention to another, more important visual locus.” [Lakoff, 1989]. Similarly Manovich [Manovich, 2009] describes how media are exchanged in social contexts, becoming tokens used to initiate or maintain a conversation. In these terms, rather than presenting a series of graphic spaces, the demonstration of technologies such as Wookie constitutes a discourse which redirects the locus of attention. The provision of the ability to demonstrate an idea enacted in technology, is a powerful intervention, and all the more so if, like Wookie, it is sufficiently flexible to be demonstrated in a a number of the various contexts where discourse is taking place. What then was the meaning of Wookie in this technological discourse?

The declared aims of IMS LD were not to achieve a change in practice or discourse, but were pedagogically neutral, specifically to support “pedagogical diversity and innovation, while promoting the exchange and interoperability of e-learning materials” [I M S Global Learning Inc., 2003] We hypothesise that enthusiasts for IMS LD (amongst whom the authors have to varying degrees been counted) were strongly focused on the potential benefits to be gained in moving from a content based description of the educational process to the more explicit representation of pedagogical activities which the specification offered, and the new opportunities for experimentation and facilitation which this opened up. The technological artefact which is IMS LD had (as Lakoff suggests) redirected our minds from the current locus of attention to another seemingly more important locus.

However, Luhmann's analysis implies a warning that this insight is often obtained by a shift in view rather than an expansion of our field of vision, as summarised by Kallinikos:

> The predictable forms by which technology often (but not always) operates are precisely due to the construction of simplified or planned causalities, whose recurrent unfolding is ensured through the exclusion (or the attempt to such an exclusion) of any possible factor that could impinge on and disturb such a functionally simplified order. [Kallinikos, 2005]

From this perspective we can think of content based adaptive learning of the type supported by SCORM [ADL Technical Team, 2004] as a planned causality which excludes the teacher as a contributor to the outcomes of the learning process. On the other hand the learning activity centred view stimulated by IMS LD (and the wider learning design movement), generates planned causalities centred on the design of the activities. This either excludes the causality of the learning materials, or places them in a highly complex relationship with the activity structures, and while it recognises the role of the teacher as a causal factor, it excludes the teachers response to the emerging contingencies of the teaching and learning process. The development of Wookie provided a solution to the technical problem which confronted the IMS LD reference implementations, but it did so with highly flexible services, which could be readily switched at runtime and could be deployed in a wide range of other environments. This drew attention to the area outside the planned causality of IMS LD, and redirected redirected the minds of some researchers (including the present authors) "from the current locus of attention to another seemingly more important locus".

This discussion is of great interest to those involved in the development of these particular technologies, as they wish to understand their own experience, but the reader may find them inconsequential, if not a council of despair to developers. In the following section, however, we move on to consider how an interoperability strategy can improve the prospects for the use of software, by encouraging it to be used in a wide range of contexts, and so to form part of the on-going technological discourse.

5 Interoperability strategy

In the previous sections we have seen how Luhmann proposes a view of technology as a "causal closure of an operational area" [Luhmann, 2008], whose results cannot be modelled or predicted, while [Kelly, 2010] points out that technology is composed of feedback loops and complex interactions of such complexity that their configuration determines the scope of technological developments available to us. In the light of this, what strategy should technologists adopt to give their creations the best chance of flourishing?

The IEC which developed Wookie is the home of the Centre for Educational Technology and Interoperability Standards (CETIS), and so naturally our default choice is to use interoperability specifications wherever possible in our development work. The movement towards interoperability specifications was driven by a desire for social inclusion and collaboration (interoperability enables all users and institutions to access all resources), and effectiveness of development (code developed in one institution could be used to fulfil the same purpose in another institution). This latter aspect reached its apogee of ambition in the eFramework [Wilson, 2003], which aspired to defining a set of domain descriptions for different areas of education, and a set of APIs which would enable components from various sources to be integrated. However, both the complexity of the problem and the scale of the implementation presented major challenges, and the eFramework, failed to obtain traction and has been shelved.

As we have discussed, Wookie was built in order to provide services for IMS LD runtime, and the user experienced the services as completely integrated into the Learning Design Player. However, informed by the focus on interoperability in the IEC, from its inception the development team saw the problem as a specific instance of a more general problem of service interoperability. This strongly informed the choice of of an open specification as the basis of the solution, W3C Widget Packaging and XML Configuration [W3C, 2011]. This formed the basis of a wider strategy to promote interoperability and potential reuse of Wookie which rested on three planks. Firstly, in line with the policy established by the TENCompetence project which funded Wookie, all development work was made public on SourceForge. Secondly, The development team contributed actively to the W3C standards process [W3C, 2011b], including participation in mailing lists, presentations at the W3C face-to-face working group meeting at Orange/France Telecom, and submitting a joint white paper with the PALETTE project (see http://palette.ercim.org/). As a result of this, Scott Wilson from the IEC team was invited to become a member of the W3C Web & Hypertext Applications Group (see http://www.w3.org/community/whatwg/) and to contribute formally to the development of the specifications. Thirdly, a focus of development work was to ensure that Wookie was fully compliant with the W3C Widget specification, even if this functionality was not immediately required. In December 2009, at the close of project funding, it was announced that it had passed all 166 W3C Widgets conformance tests (Scott Wilson, 2009) a status which it still maintains at the date of writing [W3C, 2011b]

The fact that the solution which we developed for the IMS LD provides services through a generic widget infrastructure proved valuable from the perspective of effective development. It reduced the amount of LD-specific code that needed to be maintained as part of an LD runtime platform, and made existing widgets available for use in IMS LD. Conversely, it was hoped that developing LD in a direction that converges with developments in the wider web (including both web 2.0 and mobile applications) would enhance the ability to deploy LD in more environments.

In September 2009, Wookie's acceptance into the Apache Incubator (see http://incubator.apache.org) was formally confirmed, following a process described in [OSS Watch, 2011] The incubator provides a mechanism for developing communities around open source software; once these become sustainable the project graduates to become one of the core Apache Software Foundation (ASF) projects. Since entering the incubator, a number of enhancements have been made to Wookie as part of its transition to a sustainable community project. These include a much improved approach to management of the software code itself, including dynamic management of dependencies, automated build and deploy. Today Wookie makes use of the ASF services for code management, issue tracking, mailing lists, wiki and website support. This significantly reduces the overhead in continuing the project. More significantly, a team of experienced ASF members are mentoring the project to ensure its future success.

When the Wookie development team asked if we could credit the TENCompetence project and the EU funding, we received this response from Dan Brickley at W3C:

> I'd love to see it included. The structure of academia tends to reward scholarly paper-publishing but doesn't really know what to do with software and data work. European projects also tend towards producing deliverables that are mostly likely to be giant PDFs rather than running re-usable code. So when we do finally get useful outputs from European research funding that enrich the open standards / open source scene, please let's not be shy in celebrating that! Maybe others will follow the great example, and start thinking more seriously about open source life-after-funding for their codebases, rather than taking a "throw the code over the wall and hope for the best" approach. Millions of euros get spend every year on these EU research projects, that's a lot of lines of code that

could be going into the common pool..." (Personal communication, cited in [Griffiths, 2010], p.46)

While this is only the view of one person, it is a highly informed view, which suggests that the path taken by Wookie into an Open Source foundation is the exception rather than the rule. One factor which contributes to this may be that developers on projects often do not have the expertise to know where or how to submit their code for inclusion in an open-source foundation. For example, one of the key requirements of an Apache project is ensuring that all the code is available under compatible open source licenses. Thus one of our first tasks in preparing the Wookie code for submission was to check the ownership of every file used and obtain appropriate licenses and/or permissions. This required soliciting Contributor License Agreements (CLA) from all contributors, which included developers at the University of Bolton, Logica, and several external independent developers whose code was reused in Wookie. In addition, as the primary developers, the University of Bolton had to supply ASF with a corporate CLA. This was no small task, and in itself might well be enough to deter potential contributors. In this respect the support of OSS Watch (see http://www.oss-watch.ac.uk) was invaluable in mentoring the development team through the submission process, as was the sponsorship of Ross Gardler of OSS Watch in proposing Wookie as an Apache incubation project.

In section 5 we outlined the difficulties which confront anyone who is planning a technological intervention in being able to predict the results which will be achieved in practice. The interoperability approach which we have outlined in this section serves to mitigate this difficulty. Firstly, it enables the technology to be included in more of the feedback loops which are generated between technology and users, and in the discourse around them. Secondly, the more contexts which the software is able to operate in, the greater the possiblility that it will still be useful when the social understanding and use of technology has shifted (partly in response to its own introduction). In the following sections we discuss how this occurred for Wookie as regards the concept of a Personal Learning Environment (PLE).

5.1 The implications of Wookie for the design of Personal Learning Environments

The concept of a Personal Learning Environment was introduced in [Olivier, 2001), and further elaborated in [Johnson, 2008]. The concept resonated with many TEL researchers, and has inspired conferences, workshops and development projects. The core observation was, firstly, that learners have many tasks to carry out, using technology to manage many of them, and secondly that educational institutions often control information needed by learners by insisting that learners visit institutional systems before they can access or act on their tasks. The result is that learners need to learn how to access a wide variety of systems in carrying out their role as students and citizens. This not only creates cognitive load and wastes time, it also positions learners so that they have to adapt themselves to the institutions convenience, rather than vice versa. The PLE proposed "putting the learner centre stage" and empowering them "to customize and even construct their own personal learning environments" [Wilson, 2009a]. Various degrees of accommodation between the learner and the institution have been proposed, and the term 'institutional PLE' (iPLE) has been coined, and defined by [Millard, 2011] as being a system that provides a personalised interface to University data and services and at the same time exposes that data and services to a student's personal tools. Once the functionality provided by Wookie became available, it became clear to the team at IEC that it had a potential role to play in experimentation with the idea of a PLE. We plan in a future paper to examine in detail the degree to which Wookie addresses the requirements which were foreseen for the Personal Learning Environment, and here we restrict ourselves to a brief discussion of the technical implications.

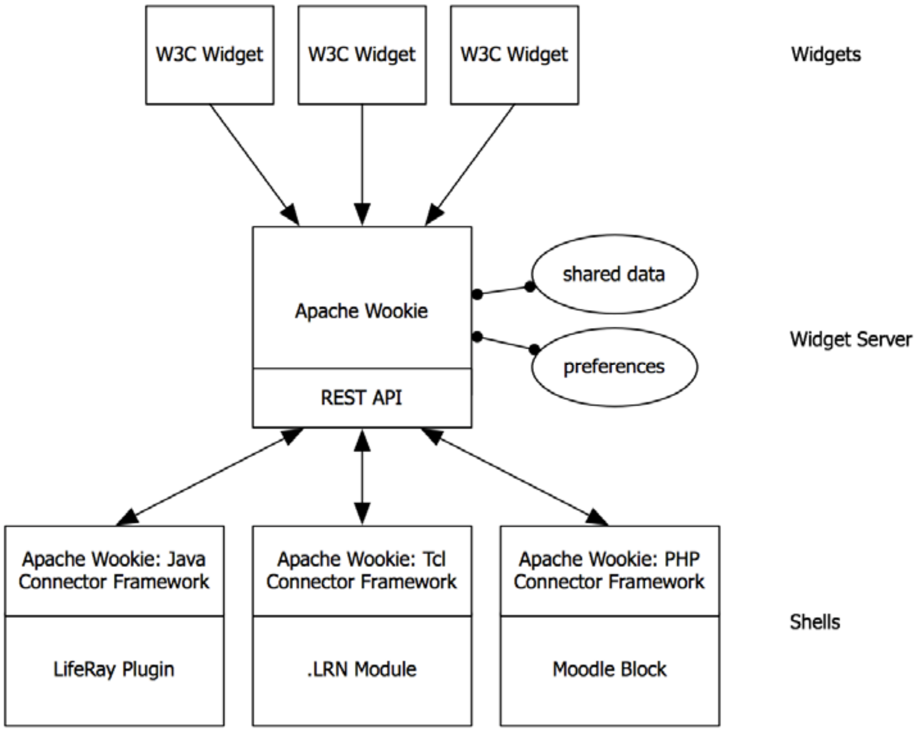

Figure 2: Use of Wookie in the Connector Framework of the iTEC project [see Griffiths et al 2011]

VLEs and Content Management Systems have plug-in systems which enable developers to add new features. If a developer creates a connector plug-in which can communicate with Wookie, then the host application immediately becomes able to select and make use of any widget which is available on the selected Wookie server. This is not a major development effort, and, for example, plug-ins have been developed for Moodle and Wordpress in a matter of days. Development of a Wookie plug-in for the Java portlet environment of LifeRay proved more challenging, but by no means prohibitively difficult. Moreover the widget will often have certain advantages over the native functionality. A Wookie widget will provide immediate updates without a page refresh, supporting the creation of real-time data from databases, simulations or social applications. A still more mould-breaking feature was that, if the plug-in provides context information, the same instance of a widget in multiple platforms could share and manipulate the same data.

A Moodle course was made public which consisted almost entirely of content delivered through Wookie widgets [Wilson, 2011]. Demonstrations were also made of widgets running in both CopperCore and Moodle with shared data which could be edited from either environment. These demonstrations depended directly on the functionality which W3C Widgets could provide within a browser, but another benefit also followed from the use of an interoperability specification which was less direct. The IMS Learning Tools Interoperability (LTI) specification, like Wookie, included support W3C Widgets, and consequently it was possible to include Wookie as an interoperable tool, and indirectly achieve integration between Wookie and any system which supports LTI. Chuck Severence learnt about this functionality and entirely independently of the Wookie team demonstrated a Wookie widget running in Blackboard, Moodle and WebCT [Severance, 2009].

The aim which the PLE discourse had inspired of enabling learners to consume and configure services from a number of institutional sources within their own favoured applications, had hitherto been more of an aspiration than a practical goal. The interoperability strategy adopted by Wookie, however, transformed the situation by providing an easily implemented mechanism whereby this could achieved with relatively little effort. As may be imagined, this sparked great interest among those researching PLEs. A number of projects working on PLEs have made use of Wookie or are doing so at the time of writing, for example the European funded ROLE and iCAMP, and the UK JISC funded WC2 and WIDE.

6 The implications of Wookie for Learning Design infrastructure

Once a user has included a widget component in an application (for example by inserting a block into a Moodle page which has “Wookie” selected as the activity) they are able to select any of the widgets on that Wookie server, and thereby share them with the learners who have access to that space. This is a very easy process to carry out, and the results are immediately available to the learners making use of the course [Wilson, 2011]. The combination of Wookie and a VLE, can therefore provide teachers with a much larger range of services and resources than a single VLE can support. It also enables the teacher to switch between them in response to the emerging circumstances of the class (and, if institutional policy and configuration of the learning environment so permits, this choice can also be placed in the hands of the learner). This functionality, as we mentioned above, drew attention to the limited ability of IMS LD systems to respond to emerging classroom circumstances, and raised the possibility of using Wookie to build a system which would seek to support precisely this purpose. The IEC team were approached to participate in the iTEC project, because the coordinators had also recognised this potential. iTEC (see http://itec.eun.org/) is a major European Integrated Project, with the goal of stimulating use of Information Technologies in classrooms around Europe. iTEC scenarios are supported by collections of widgets, delivered to *shells* (currently Moodle, .LRN and Liferay). From an IMS LD centric view, this looks like an *environment* with an *activity instruction.* However, there are two major differences. Firstly, the teacher takes responsibility for orchestration of activities, rather than the computer. Secondly, the teacher can switch the services and resources in the environment in response to the emerging dynamics of the classroom. In this way the capabilities of Wookie which we have described in earlier sections have proved to be useful in a learning design context other than IMS LD.

More interestingly, another aspect of Wookie which had previously passed relatively unnoticed has come to prominence in iTEC. A mechanism is provided whereby the user can inspect a gallery showing the widgets available on the server, and select the one to be displayed. In discussions in iTEC it became clear that this functionality could provide a curated collection of widgets which could be made available to teachers to choose from, and that this could therefore provide the basis of a simple real time authoring environment for teachers. The power of this functionality became clear to the project because another quite separate technology had directed the attention of participants to another locus of attention. This was the App Store available on smart phones, which enabled users to install new applications on their phone extremely easily, and (in some environments) with the support of recommender systems. After extended discussion of the various possibilities, the iTEC project decided to build its architecture around an App Store, with extensive social functions to provide recommendations.

It was also necessary to describe the functionality of widgets so that the appropriate tools and resources could be found. Other parts of the project make use of controlled vocabularies for similar purposes, but this was problematic for widgets for two reasons. Firstly, the functionalities of widgets cover large parts of the domain of computer based tools, and their description could not be exhaustive, and secondly, the functionalities of tools overlap. This impasse was resolved, again, by drawing attention to another technology, in this case an affordance based classification method, which had been developed in another project, iCAMP [Valjataga, 2007], for the purposes of classifying tools in a PLE. Here we can see a recursive re-entry in which a domain (PLE research) whose development had been informed by Wookie, generated functionality which in turn influenced the development of Wookie. Given the new importance of the selection by users of the widget to be displayed, the widget gallery is being deprecated from the Wookie server, and will be replaced by an API. The iTEC App Store is currently being developed as a separate server application, and the hope is that other developers will develop other applications using the API, perhaps creating functionalities which are at present unforeseen.

Wookie is a tool which has come out of a TEL project, but because of its interoperability and open source policies it has attracted the attention of technologists in other domains. This has led to the IEC team working on Wookie in the context of the Omelette project which is researching "the development, management, governance, execution and conception of converged services with a specific focus on the telco domain" [Omelette, 2010]. In doing this it is extending the engagement of Wookie with interoperability specifications, using emerging specifications such as W3C Device API and Policy (see http://www.w3.org/2009/dap) and Bondi [OMTP, 2010] so that device-specific capabilities (e.g. GPS, SMS, addressbook/PIM) can be incorporated into mashups. This will enable, for example, a Wookie widget embedded in Moodle to send an SMS or make a phone call to a learners mobile phone. It will be of interest to see what the impact of this technology will be on educational users understanding of the VLE and the possible uses of technology in education.

7 Conclusions

In this paper we have described the process through which a successful approach to the integration of services was developed. This is worth doing in its own right, providing documentation of a story which may be of value to other developers and to researchers into the design and use of technology. Our main purpose, however, has been to frame this story as a case study which illustrates the difficulty of planning technological interventions in social systems, and identifies interoperability as a key strategy in making these as responsive as possible to emerging social dynamics. In doing this we made use of ideas drawn from sociology which indicate that technological interventions:

a) carry with them an enclosure of causality (and so may be considered to imply statements about the way the world works)
b) redirect the attention of potential users towards new locus (and so may be considered to constitute statements in a discourse)
c) generate complex feedback loops with the context in which they take place (and so their results are unpredictable, and may transform the environment which generates the needs which they are intended to address)

This implies that a view of user needs which sees them as independent of technology is at best misleading, and at worst may lead to sub-optimal use of technology. We ascribe some of the problems experienced by IMS LD to this. The specification was developed as the result of an intensive needs analysis process, and led to a single very extensive specification which was then implemented. In contrast Wookie was developed as a response to an immediate problem (how do we deliver services in an IMS LD environment), using the technical W3C widget specification, and with little direct contact with end users. Nevertheless Wookie has achieved greater adoption. Our conclusion is that this is due to the interoperability strategy adopted. Rather than creating their own specification for the interoperability of learning tools, the team chose to treat the task as a specific case of a much wider problem, and built on the ideas emerging around the generic technology of widgets, themselves embedded in the wider interoperability framework of the Web. The resulting system provided a means whereby demonstration and experimentation could be carried out to see in practice how users requirements and patterns of use developed in response to the technological intervention. An example of this is the demonstration of Wookie in Moodle, [Moodle, 2009] and [Wilson, 2011]. The fact that Wookie implemented a relatively small set of functional requirements and defined a simple set of APIs meant that it could be integrated easily with a wide range of existing and emerging applications and platforms. In contrast, IMS LD was a very large specification, which was necessarily implemented with extensive and complex systems. Consequently, while the technology led to extensive experimentation on the lines described by Koper above, it was hard to adapt it to address some of the questions which emerged from the on-going research. In this we find that we are echoing Popper, who argued that in seeking to transform society "The piecemeal engineer will ... adopt the method of searching for, and fighting against, the greatest and most urgent evils of society, rather than searching for, and fighting for, its greatest ultimate good" [Popper, 1966], p. 158. He opposes this to the Utopian approach, in which "we must determine our ultimate political aim, or the Ideal State, before taking any action".

We have described how 'piecemeal' approach, together with an interoperability strategy, led to Wookie becoming an intervention in the technology and discourse around the PLE, and in IMS LD and the wider field of learning design. In this paper we have only briefly identified the technical aspects of this process, and there is a great deal more to be said about the affordances which Wookie offers for users in these two fields, which we will address in a future publication.

In the light of this, we ascribe the success of Wookie to

a) focusing on relatively small sets of functional requirements addressing a problem at hand, which can be reused in multiple contexts
b) placing interoperability at the heart of the design of the system.
c) avoidance of domain specific specifications wherever possible
d) open-source code, and open development practices

As well as giving the technological intervention a greater chance of success (because its goals are more limited), this also optimises the agility its redeployment and adoption in unforeseen contexts. In this way the software is better positioned to play a continuing role in the on-going transformation of the technological landscape which emerges from the recursive interactions of users and technology.

However, we do not conclude from this analysis that developers should ignore users. As technologists we have an unavoidable professional and ethical obligation to try to understand the likely consequences of our interventions. Just as we have drawn on sociology in understanding the problem, we believe that methodologies drawn from the same field can help us in resolving it. Specifically, drawing on Realistic Evaluation [Pawson, 1997] our recommendation is that the designers of technologies hypothesise about their products as interventions in a context in which the outcomes are determined by a specified social mechanism. In effect, this means being aware of and explicit about the causality which they ascribe to the system which they are creating (the intervention which they are planning, and the outcomes which they hope for) and about the mechanisms which they believe will bring this about, including the feedback loops resulting from the influence of the system on its context. It is to be expected that the initial formulation of the mechanism will not perfectly describe the outcomes, and that these will also shift as the context evolves. Thus this hypothesising should not only inform the design, but also be modified according to the emerging results of experimentation with the resulting technology. Such an approach can be adopted more easily with technologies which are sufficiently agile and interoperable to be used in a number of different contexts. In the case study which we have presented, In the case study which we have presented, it is this which was enabled by the way in which interoperability specifications and open source development processes were used in development of Wookie (although this is not the only benefit of interoperability specifications).

Google chief executive Eric Schmidt explained his company's development policy in simpler terms in his comments on Wave: "Our policy is, we try things." [Barnett, 2010]. In this paper we have set out what it means for us to “try things” with technology, how this can be a powerful strategy even for a relatively small development initiative when it is allied with interoperability specifications and open methodologies, and how this has supported the success of Wookie as a system promoting interoperability of services.

Acknowledgements

We would like to acknowledge the support provided by the European Commission for the development of Wookie through the TENCompetence, iTEC, and Omelette projects. We would also like OSS Watch for their facilitation of the process of applying for Wookie's admission to the Apache Incubator.

References

[ADL Technical Team, 2004] ADL Technical Team: "Sharable Content Object Reference Model (SCORM) 2004 2nd Edition" (2004). Retrieved from http://www.adlnet.org/downloads/70.cfm

[Apache Foundation, 2011] Apache Foundation: "Apache Wookie (Incubating)" (2011). Retrieved July 14, 2011, from http://incubator.apache.org/wookie/

[Barnett, 2010] Barnett, R.: "Google Waves Goodbye To Messaging App"; Online Sky News, Thursday August 5th 2010. Retrieved from http://news.sky.com/home/technology/article/15677447

[Bhaskar, 1975] Bhaskar, R.: "A realist theory of science"; Sage Publications (1975).

[Burgos, 2005] Burgos, D., Griffiths, D.: "The UNFOLD Project: Understanding and using Learning Design"; Open University of the Netherlands (2005). Retrieved from http://hdl.handle.net/1820/548

[De La Fuente, 2007] De La Fuente, L.: "Integration in Generic Tool Learning Design to Support Complex Learning Methodologies"; Proc. EC-TEL 2007 PROLEARN Doctoral Consortium, Crete, Greece. CEUR Workshop Proceedings 288 (2007).

[Derntl, 2012] Derntl, M., Neumann-Heyer, S., Griffiths, D., Oberhuemer, P.: "The Conceptual Structure of IMS Learning Design Does Not Impede Its Use for Authoring"; IEEE Transactions on Learning Technology, 5, 1 (2012) 74-86

[Fischer, 2011] Fischer, G., & Shipman, F.: "Collaborative Design Rationale and Social Creativity in Cultures of Participation"; Human Technology, 7, 3 (2011)

[Georgiev, 2009] Georgiev, A., Griffiths, D., Hernández-Leo, D., Melero, J., Stefanov, K., Perez, M., Sharples, P., et al.: "TENCompetence Deliverable D6.2 annex 1 - Validation testing and usability results" (2009). Retrieved from http://hdl.handle.net/1820/1730

[Giddens, 1984] Giddens, A.: "The Constitution Of Society, outline of the theory of structuration"; Cambridge, UK (1984)

[Griffiths, 2005] Griffiths, D., Blat, J.: "The Role of Teachers in Editing and Authoring Units of Learning using IMS Learing Design"; International Journal on Advanced Technology for Learning, International Journal on Advanced Technology for Learning, Special Session on "Designing Learning Activities: From Content-based to Context-based Learning Services, 4, 2 (2005)

[Griffiths, 2007] Griffiths, David, Koper, R., Liber, O., Sharples, P.: "Integrating IMS Learning Design and ADL SCORM using CopperCore Service Integration"; Proc. Service Oriented Approaches and Lifelong Competence Development Infrastructures, 2nd TENCompetence Open Workshop, Institute for Educational Cybernetics, Bolton, 8-91. Retrieved from: http://hdl.handle.net/1820/1023. Retrieved from http://hdl.handle.net/1820/1023

[Griffiths, 2008] Griffiths, D., Franklin, T., Sharples, P., Beauvoir, P., Wilson, S., Miao, Y., Tattersall, C., et al.: “D6.1 Report with summary of WP6 outputs of month 1-18”; TENCompetence (2008) Retrieved from http://hdl.handle.net/1820/1148

[Griffiths, 2009] Griffiths, David, Beauvoir, P., Liber, O., Baxendale, M. B.: “From Reload to ReCourse : learning from IMS Learning Design implementations”; Distance Education, 30, 2 (2009)

[Griffiths, 2010] Griffiths, David, Franklin, T., Sharples, P., Beauvoir, P., Wilson, S., Miao, Y., Tattersall, C., et al.: “TENCompetence deliverable D6.4. Evaluation” (2010). Retrieved from http://hdl.handle.net/1820/2298

[Hazlewood, 2008] Hazlewood, P., Oddie, A., Barrett-baxendale, M.: “Using IMS Learning Design in Educational Situations”; In L. Lockyer, S. Bennett, S. Agostinho, & B. Harper (Eds.), Handbook of Research on Learning Design and Learning Objects: Issues, Applications, and Technologies, IGI Global (2008) 282-294.

[Hernández-Leo, 2006] Hernández-Leo, D., Villasclaras-Fernandez, E. D., Asensio-Perez, J. I., Dimitriadis, Y., Jorrin-Abellan, Ivan M.; Ruiz-Requies, I., & Rubia-Avi, B.: “COLLAGE: A Collaborative Learning Design Editor Based on Patterns”; Educational Technology & Society, 9, 1 (2006), 58-71.

[I M S Global Learning Inc, 2003] I M S Global Learning Inc. “Learning Design Specification Version 1”; (2003). Retrieved from http://www.imsglobal.org/learningdesign/

[Johnson, 2008] Johnson, M., Liber, O.: “The Personal Learning Environment and the human condition: from theory to teaching practice”; Interactive Learning Environments, 16, 1 (2008) 3-15.

[Johnson, 2009] Johnson, M., Griffiths, D., & Hanslot, Z.: “Positioning Learning Design: Learner Experience and the challenges of transforming teaching practice”; Proc. TENCompetence Open Workshop, Rethinking Learning and Employment at a Time of Economic Uncertainty (2009) 11-21. Retrieved from http://hdl.handle.net/1820/3191

[Johnson, 2011] Johnson, M., Griffiths, D., & Wang, M.: “Positioning Theory, Roles and the Design and Implementation of Learning Technology”; Journal of Universal Computer Science, 17, 9 (2011), 1329-1346

[Kallinikos, 2005] Kallinikos, J.: “The order of technology: Complexity and control in a connected world”; Information and Organization 15, 3 (2005) 185-202

[Kelly, 2010] Kelly, K.: “What Technology Wants”; Viking Books, New York (2010)

[Koper, 2004a] Koper, R., Manderveld, J.: “Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning”; British Journal of Educational Technology, 35, 5 (2004) 537-551

[Koper, 2004b] Koper, R.: “Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher’s Workload”; JIME Special Issue on the Educational Semantic Web (2004). Retrieved from http://jime.open.ac.uk/jime/article/viewArticle/2004-6-koper/188

[Koper, 2005a] Koper, Rob, & Tattersall, C.: “An Introduction to Learning Design. Learning Design: modelling and implementing network-based education & training”. Berlin: Springer Verlag (2005), 3-20

[Koper, 2005b] Koper, Rob, Tattersall, C., Wilson, S.: “Architectures to Support Authoring and Content Management with Learning Design”; in Learning Design: modelling and implementing network-based education & training, Springer Verlag (2005), 41-62

[Lakoff, 1989] Lakoff, G., & Turner, M.: “More Than Cool Reason, a Field Guide to Poetic Metaphor”; The University of Chicago Press (1989)

[Luhmann, 2008] Luhmann, N. “Risk”, Aldine Transaction (2008)

[Maguire, 2002] Maguire, M., & Bevan, N. “User requirements analysis A review of supporting methods”; Proc IFIP 17th World Computer Congress, Usability: Gaining a Competitive edge, Kluwer (2002); 133-148

[Manovich, 2009] Manovich, L. “The Practice of Everyday (Media) Life: From Mass Consumption to Mass Cultural Production?”; Critical Inquiry, 35, 2 (2009)

[McAndrew, 2005] McAndrew, P., Nadolski, R., Little, A.: “Developing an approach for Learning Design Players”; Journal of Interactive Media in Education, 14 (2005) 1-12

[Milllard, 2011] Millard, D.E., Davis, H. C., Howard, Y., McSweeney, P., Yorke, C., Solheim, H., Morris, D. “Towards an institutional PLE”; in Proc. Perosnalised Learning Environment Conference 2011, Southampton, GB (2011)

[Moodle, 2009] “Google Wave discussion thread” (2009). Retrieved December 15, 2011, from http://moodle.org/mod/forum/discuss.php?d=124670

[Neumann, 2009] Neumann, S., Oberhuemer, P., Derntl, M.: “Visualizing Learning Designs Using IMS Learning Design: The Position of the Graphical Learning Modeller”; ICALT 2009 Ninth IEEE International Conference on Advanced Learning Technologies (2009) 732-733

[Neumann, 2010] Neumann, S., Klebl, M., Griffiths, D., Hernández Leo, D., de la Fuente Valentín, L., Hummel, H., Brouns, F., et al.: “Report of the Results of an IMS Learning Design Expert Workshop”. International Journal of Emerging Technologies, 5, 1 (2010) 58-72

[Olivier, 2001] Olivier, B., Liber, O.: “Lifelong Learning: The Need for Portable Personal Learning Environments and Supporting Interoperability Standards”; CETIS Working Paper, Retrieved from http://wiki.cetis.ac.uk/images/6/67/Olivierandliber2001.doc

[Omelette, 2010] Omelette: “Project Objectives”. Retrieved December 15, 2011, from http://www.ict-omelette.eu/objectives

[OMTP, 2010] OMTP: “BONDI 1.1 Approved Release” (2010). Retrieved from http://bondi.omtp.org/1.1/index.html

[OSS Watch, 2011] OSS Watch: “Wookie : a case study in sustainability”. Retrieved December 15, 2011, from http://www.oss-watch.ac.uk/resources/cs-wookie.xml

[Pawson, 1997] Pawson, R., Tilley, N.: “Realistic Evaluation” Sage Publications Inc. (1997)

[Popper, 1966] Popper, K. R., “The Open Society and its Enemies Volume I, The Spell of Plato”; Routledge and Kegan Paul, London (1966)

[Qvortrup, 2005] Qvortrup, L.; “Society’s Educational System - An introduction to Niklas Luhmann’s pedagogical theory”; Seminar.net - International journal of media, technology and lifelong learning, 1, 1 (2005), 1-21

[Reigeluth, 1999] Reigeluth, C. M. “The elaboration theory: Guidance for scope and sequence decisions”; in C. M. Reigeluth (Ed.), Instructional design theories and models: A new paradigm of instructional theory, Lawrence Erlbaum Associates, Hillsdale, NJ (1999), 5-29

[Rennison, 1995] Rennison, E. “The Mind’s Eye: An Approach to Understanding Large Complex Information-Bases Through Visual Discourse”; Massachussetts Institute of Technology (1995). Retrieved from http://mf.media.mit.edu/pubs/thesis/rennisonMS.PDF

[Severance, 2009] Severance, C.; "Apache Wookie and IMS Basic Learning Tools Interoperability"; Dr. Chucks Blog (2009), Retrieved December 12, 2011, from http://www.dr-chuck.com/csev-blog/2009/11/apache-wookie-and-ims-basic-learning-tools-interoperability/

[Tattersall, 2005] Tattersall, C., Vogten, H., Hermans, H.: "The Edubox Learning Design player"; in Rob Koper & C. Tattersall (Eds.), Learning Design, A Handbook on Modelling and Delivering Networked Education and Training, Springer, Berlin (2005), 303-310.

[Valjataga, 2007] Valjataga, T., Pata, K., Laanpere, M., Kaipainen, M.: "Theoretical Framework of the iCampFolio - New Approach to Comparison and Selection of Systems and Tools for Learning Purposes. Creating new learning experiences on a global scale"; LNCS, Volume 4753, Springer, Berlin/Heidelberg (2007), 349-363

[Vogten, 2006] Vogten, Hubert, Martens, H., Tattersall, C., Van Rosamalen, P., Nadolski, R., & Koper, R.: "CopperCore Service Integration; Integrating IMS Learning Design and IMS Question and Test Interoperability"; Proc. ICALT 2006, Kerkrade, The Netherlands. Retrieved from http://hdl.handle.net/1820/570

[Vogten, 2007] Vogten, H., Martens, H., Nadolski, R., Tattersall, C., van Rosmalen, P., Koper, R.: "CopperCore Service Integration"; Interactive Learning Environments, 15, 2 (2007) 171-180

[W3C, 2011a] W3C.: "Implementation Report for Widget Packaging and XML Configuration" (2011). Retrieved December 15, 2011, from http://dev.w3.org/2006/waf/widgets/imp-report/

[W3C, 2011b] W3C.: "Widget Packaging and XML Configuration, W3C Recommendation 27 September 2011"; (2011). Retrieved from http://www.w3.org/TR/widgets/

[Wilson, 2003] Wilson, S, Olivier, B., Jeyes, S., & Powell, A.: "A Technical Framework to Support e-Learning"; JISC Working Paper, (2003)

[Wilson, 2008] Wilson, S, Sharples, P., Griffiths, D.: "Distributing education services to personal and institutional systems using Widgets"; In F. Wild, M. Kalz, & M. Palmér (Eds.), Mash-Up Personal Learning Environments (MUPPLE'08), Maastricht, (2008), CEUR Workshop Proceedings. (2008) 5-32. Retrieved from http://ceur-ws.org/Vol-388/

[Wilson, 2009a] Wilson, S, Sharples, P., Griffiths, D., Popat, K. "Moodle Wave: Reinventing the VLE using Widget technologies"; MUPPLE'09: Mash-Up Personal Learning Environments, (2009) 47-58. Retrieved from http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-506/

[Wilson, 2009b] Wilson, S., "Apache Wookie passes W3C Widgets conformance". CETIS, (2009). Retrieved December 15, 2011, from http://zope.cetis.ac.uk/members/scott/blogview?entry=20091218112722

[Wilson, 2011] Wilson, S, Sharples, P., Griffiths, D., Popat, K.; "Augmenting the VLE using widget technologies". International Journal of Technology Enhanced Learning, 3, 1 (2011), 4-20.