Toolkits and Demonstrators Review

Wilbert Kraan and Scott Wilson, JISC CETIS
March 2007

Table of Contents

Overview and Terms of Reference

The review was requested by JISC as one of the activities of the e-Learning Framework support project, and was undertaken between September 2006 and March 2007 alongside the Toolkits and Demonstrators programme. The review is a technical report into the outcomes of the programme, and not a formal evaluation of the programme itself. In February a set of interim recommendations were made to JISC to inform the development of new funding calls.

While a range of work has been conducted within the wider brief of the e-Framework and development activities, this review limits itself to the development activity within the Toolkits and Demonstrator strand alone. It evaluates the functional coverage of the toolkits that have been developed to date and identifies any gaps that still need to be addressed. This report also reviews the approach as a whole.

The motivation for the review was to provide recommendations for the suitability of the toolkit and demonstrator approach for re-employment in future and, if it is suitable, what kind of changes would make it better. Both assessments are based on common issues and outcomes of the present set of toolkits and demonstrators. At a slightly lower level, the review identifies those toolkits that are of strategic value. That is, those toolkits that might need to be developed further and for which a sustainability model needs to be elaborated.

All of these findings are based on the outcomes of desk research, with the survey based on an examination of web resources.

Introduction

The toolkits and demonstrators programme came out of a desire to explore the possibility of a programme that was less focussed on developing complete, new and monolithic e-learning applications, and more on the service-enabling of existing software. One motivation for exploring such a direction was the rise of service oriented approaches (soa) in other sectors, with its promise of better resource utilisation, and isolation of changeable systems from the rest of the network. Another motivation was a desire to improve return on investment (ROI) by spreading risk across many, small, high return projects rather than a handful of long-running projects.

The soa aspect was one of the motivations behind the establishment of, first, the e-learning framework, and, later, the e-framework. These frameworks supplied much of the technical and conceptual structure behind the toolkits and demonstrators programme strand. Return on investment was addressed more directly in the organisation of the strand itself, and is examined in more detail in this report.

Aims of the programme

By switching the focus from applications to the services applications can provide, the toolkits and demonstrators' overarching programme intended to enable a wider range of pedagogical models, and to support a greater diversity of institutional types. Soa was used to help achieve the required flexibility by breaking down functionality into smaller units, and making it possible to recombine such services in a greater range of combinations. By specifying interoperability standards for these services, and developing a shared terminology, another aim was to improve sharing of practice and to build a common conceptual model in e-learning.

While these aims were mostly addressed in the development of the framework, they were made concrete in the intended outcomes of the toolkits and demonstrators programme: open source toolkits and interfaces that facilitated implementation at an institutional level.

Once these toolkits and demonstrators were deployed, a further aim was to develop new, service oriented models for e-learning systems at institutional level. Institutional pilots were then to be established to evaluate the generic technical framework and the implementation tools.

Finally, the programme was to facilitate the development of practical guidelines and toolkits to enable institutions to install and implement the technical framework as the basic structure of an e-learning system.
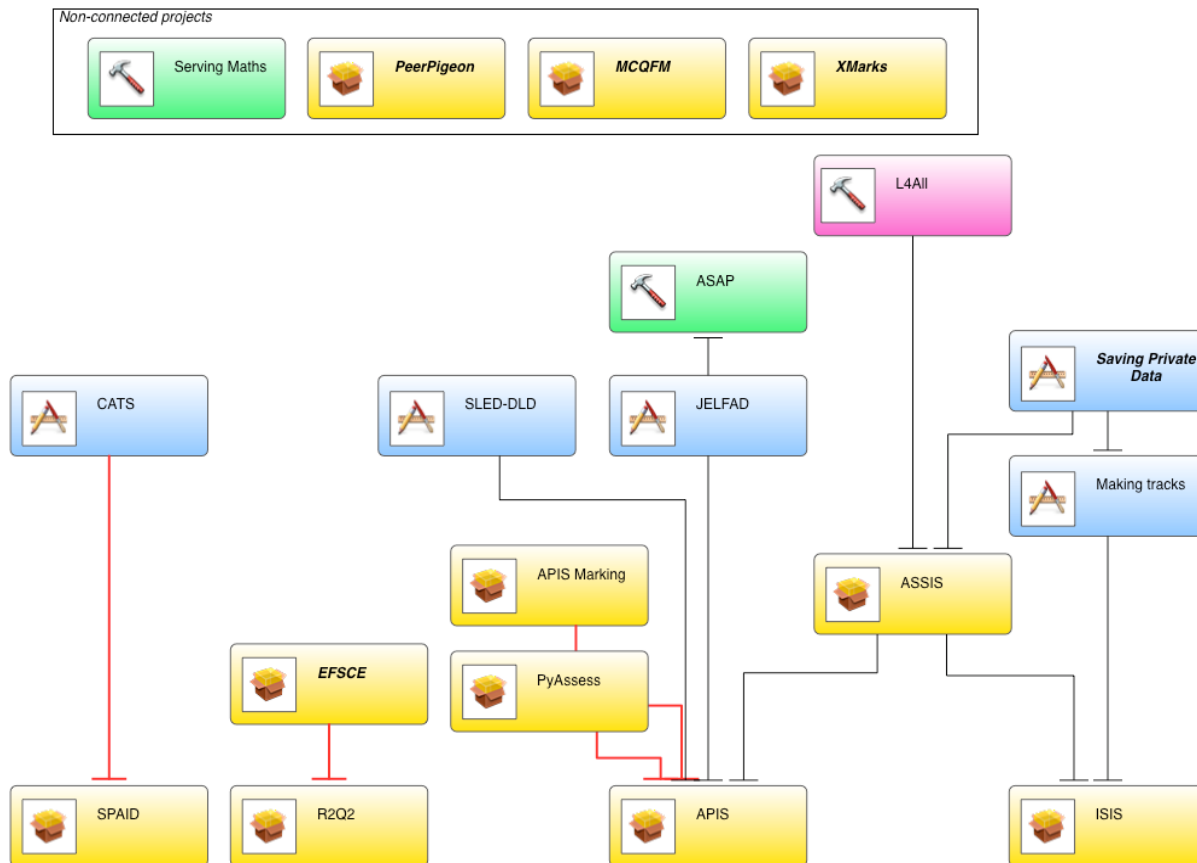
Analysis

Impact and value

We analyzed several project clusters, using the Toolkit, Demonstrator, DEL Tools I & II, and DEL Regional Pilot projects as the evidence base. We developed a family tree of projects, linking

together those projects that had a direct influence on other projects.

Assessment cluster

The assessment cluster of projects demonstrates a high level of reuse of project outputs, with APIS and ISIS standing out as success stories here. APIS was used in two demonstrators, three other toolkits, a DEL tool, and a DEL regional pilot. While the original developer of APIS no longer contributes to the work, the toolkit has continued to develop, with input from the University Pompeu Fabra in Barcelona moving the toolkit to support QTI v2.0, demonstrating the clear demand for this work.

In the second phase of the programme, Southampton developed the R2Q2 toolkit. Originally intended to develop upon APIS, the toolkit was redeveloped from scratch. So, while APIS has provided a very useful role in this domain in the early years, the R2Q2 toolkit effectively replaces it.

Overall, it would seem reasonable to assert that the APIS toolkit has played a part in making it easier for projects to use standards-based online assessment, and enabled a range of other projects to achieve QTI capability more readily and cost-effectively than would have otherwise been the case. The recent capital call will build upon the work in this space and should move the sector forwards in terms of available open-source infrastructure for online assessment.

The strand of work concerning services for the sequencing of learning objects (ISIS, ASSIS) was

technically very accomplished, but rather than support a path to adoption for others it instead highlighted the lack or readiness in the sector (particularly in terms of instructional design skills) for this level of sophisticated adaptive assessment. This finding may have saved considerable resources being spent in this area before it was ready.

The majority of projects in this space show strong interlinking and coordination, which is related to a very active JISC-CETIS SIG community in this area which has assisted the interaction of projects. It may be the case that the social networking afforded by the SIG was the primary success factor involved, bringing together projects and promoting the reuse of existing toolkits.

The toolkit approach has worked well in this space, enabling projects to bite off small chunks of a very complex space one at a time; the experience gained from this can now be applied in the Capital projects to unite these chunks into a more widely usable foundation utilising the most mature and well-tested components.

Some gaps do exist in this space, particularly support for marking and grading; however, these are the subject of new toolkit projects such as XMarks. Assessment Item banks has also been a gap for some time, but is being addressed in the Capital call, building upon initial work by SPAID and CATS.

Enterprise cluster
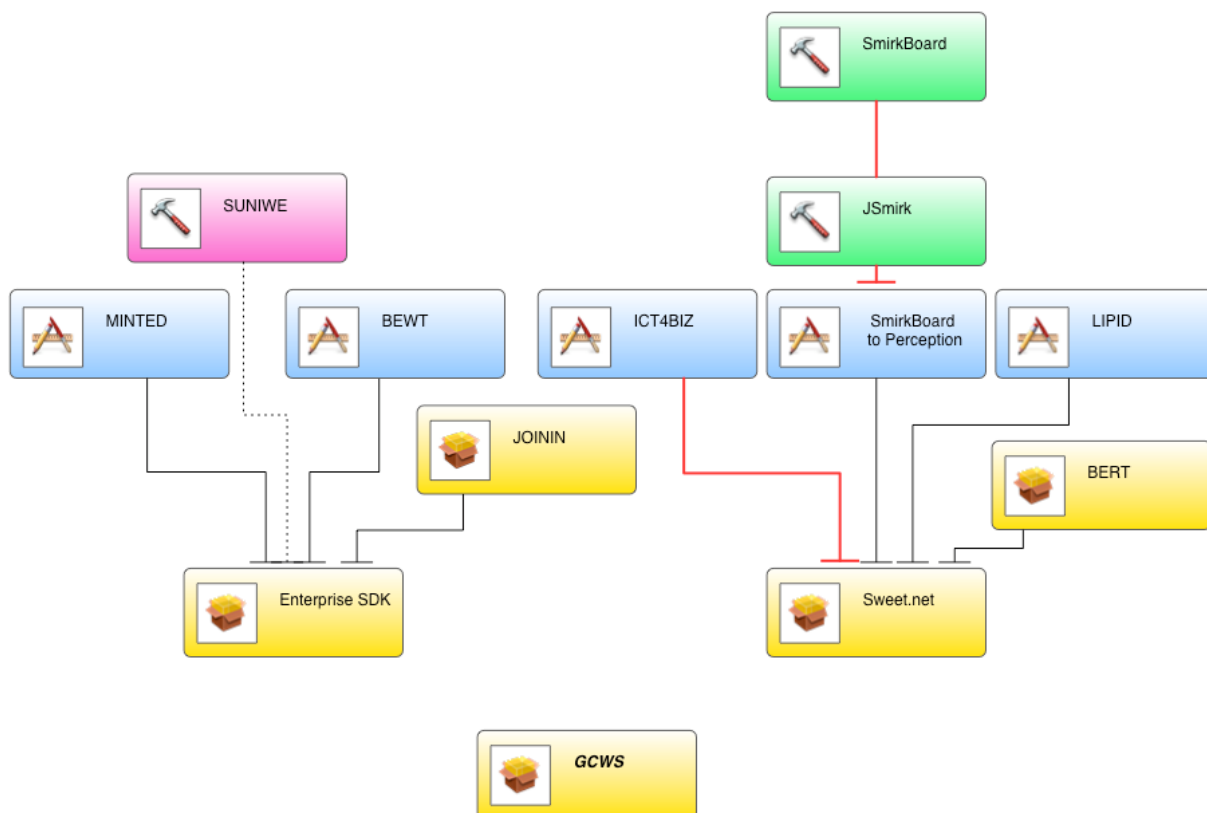
The Enterprise cluster has a relatively simple structure. Two versions of the IMS Enterprise

Services SDK (Java and .Net) have been the basis for five demonstrators and three DEL projects, and have spawned two further toolkits - JoinIn and BERT. Two additional toolkits have been recently commissioned - GCWS (calendars) and XMarks (assessment results).

The Sweet.net and EnterpriseSDK toolkits have clearly enabled projects working with the IMS specification to get up to speed much more quickly than would otherwise be the case. The JOININ and MINTED projects used toolkits to enable the rapid development and deployment of a specific service, without having to go too much into depth on the WS infrastructure. In contrast, the SUNIWE project looked at SOA broadly, with Enterprise as only one of a set of services. The SUNIWE project found implementing the specification from scratch without a toolkit quite time-consuming. These two approaches also indicate two different strategies towards institutional SOA adoption, in the "bottom-up" JOININ and MINTED approach the use of SOA spreads from small projects using working services, in the "top down" SUNIWE approach organisational capability is realised through implementation from first principles. It is too early to say which approach is better in any given institutional context.

Both toolkits, being rather technical, required additional support from the original developers with these projects; feedback from demonstrators has improved the documentation and packaging of these toolkits, but there is still room for improvement.

There has been interest from some commercial providers in using the toolkits, particularly Sweet.net, however we have been unable to discover if the code made it into any products. However, they have been used in several open-source projects as a direct result of JISC follow-on projects, including Boddington, LAMS and Moodle.

Overall in this cluster the two primary toolkits have clearly made an impact at least within the efficiency of other JISC-funded projects in the same space. The investment in both these toolkits was rather modest (we estimate a total, including additional sustainability funding, of £16,000 for the Java version; around £30,000 for .net) and easily recovered in cost savings made in the 10 related projects.

All projects in this cluster have had a strong engagement with the JISC-CETIS Enterprise SIG, and this has contributed to the overall cohesion of the projects in this space and amplified the value of the toolkits. As the toolkits have required significant personal interactions between demonstrators, tools and pilots with the original toolkits developers, the SIG has performed a critical role in bringing people together and sharing information.

With a new version of the IMS Enterprise Services specification due in 2007, it is likely that there will be a demand for updated versions of both toolkits to support new initiatives and projects.

Content cluster
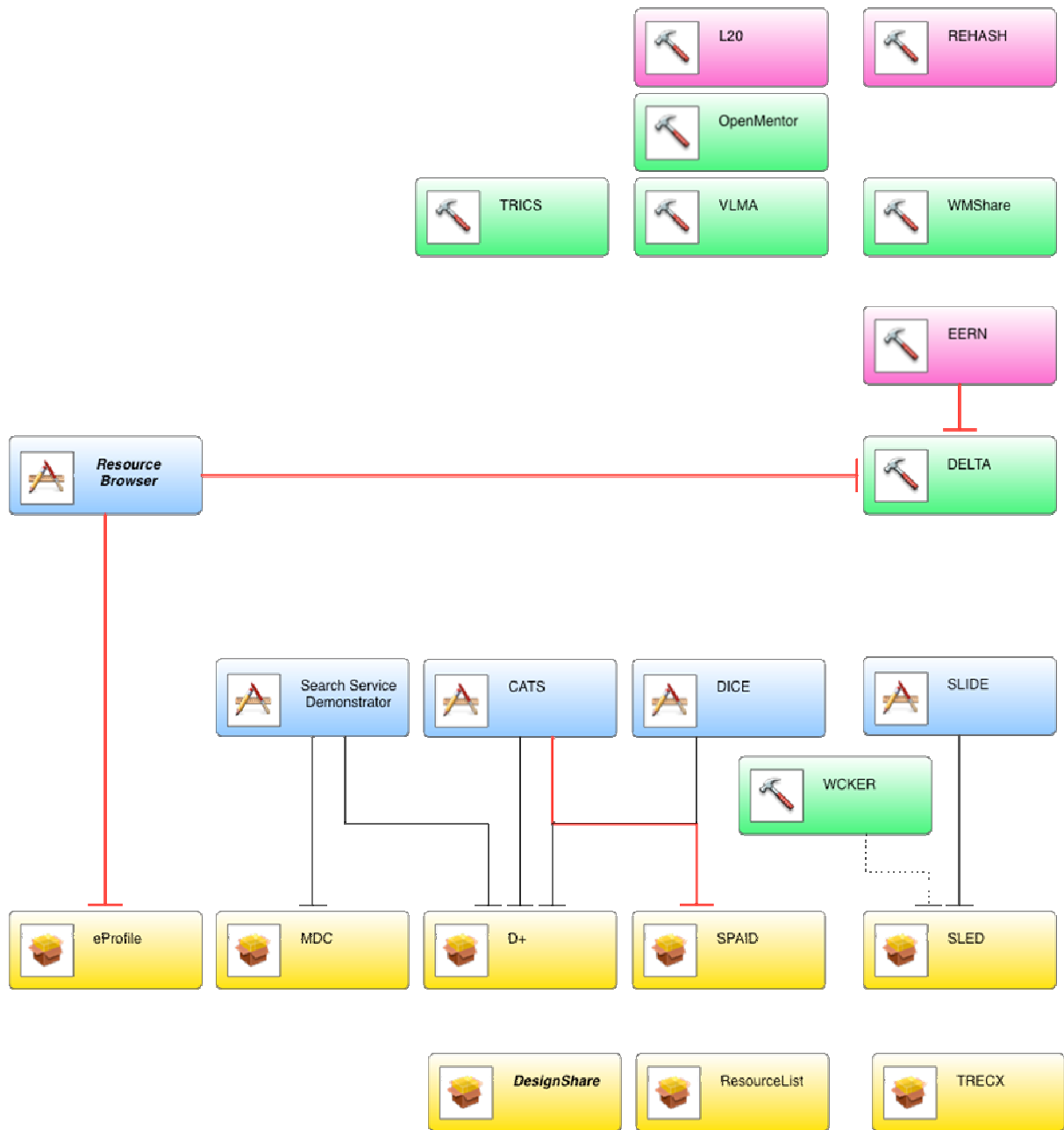
The content cluster actually spans several areas of interest: using learning designs, managing assessment items, and locating and using learning objects.

Within this space there has been considerable development and reuse of the SLeD toolkit, which provides the web runtime environment for learning designs, which has been used in two demonstrators (SLIDE, SLED-DLD) and a tool (WCKER). Within the LD space, the SLeD toolkit has been used as much for experimentation and learning about the technology as about broadening uptake or moving into implementation.

In the learning objects space, the D+ toolkit has proved quite popular, with three demonstrators. D+

itself was built upon an earlier JISC 'toolkit', the JAFER work by Oxford in 2000-2003 [http://www.jafer.org/], one of the few examples of toolkit reuse extending to "before" the programme started. While D+ has seen some adoption, JAFER has remained relatively static, while other non-JISC toolkits such as JZKit have been more actively maintained.

Within the assessment space, SPAID and CATS were early forays into item banks; the discoveries made in these projects were incorporated into the specification for a generic item bank component issued in the Capital Call in 2006.

Portfolio cluster
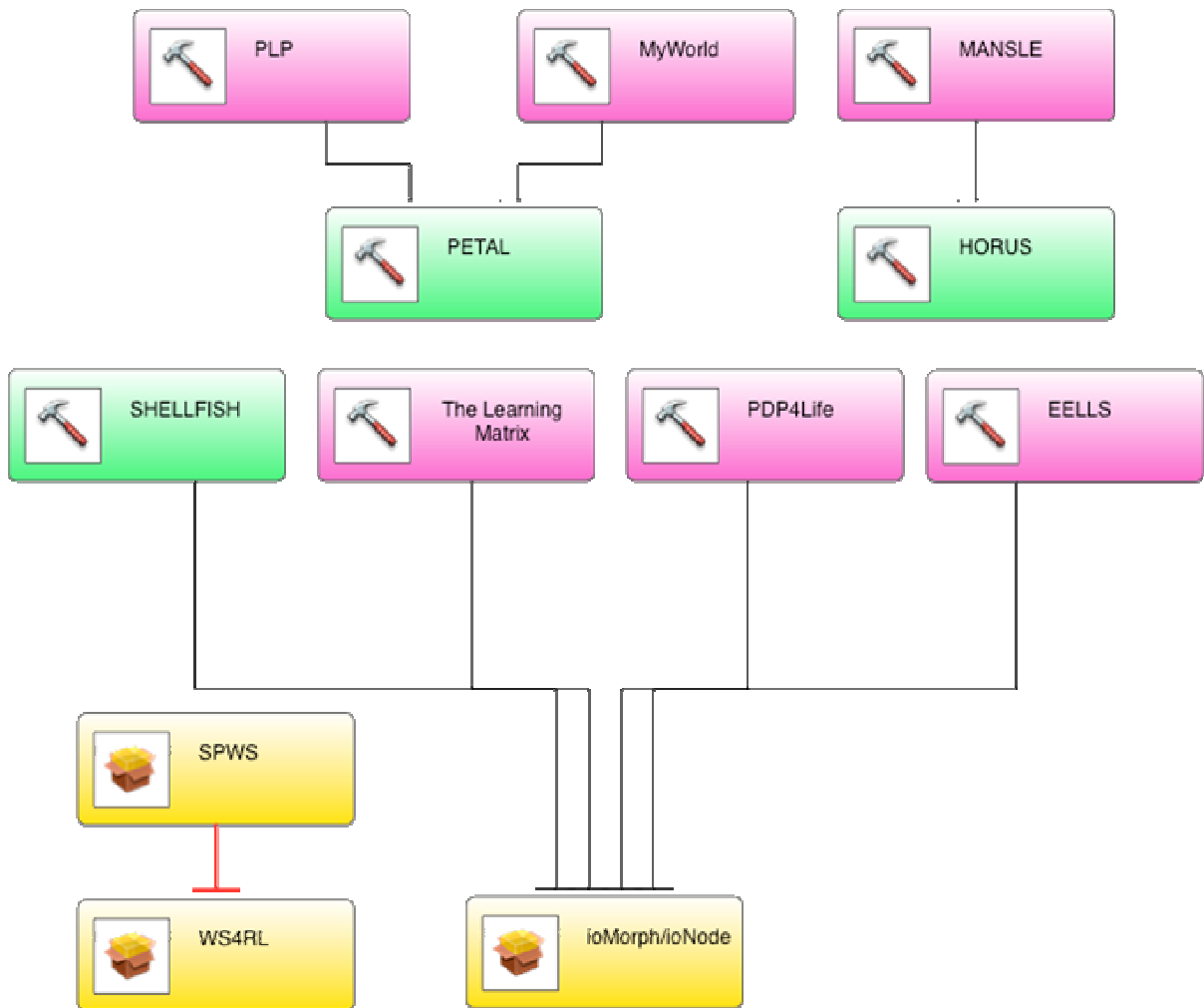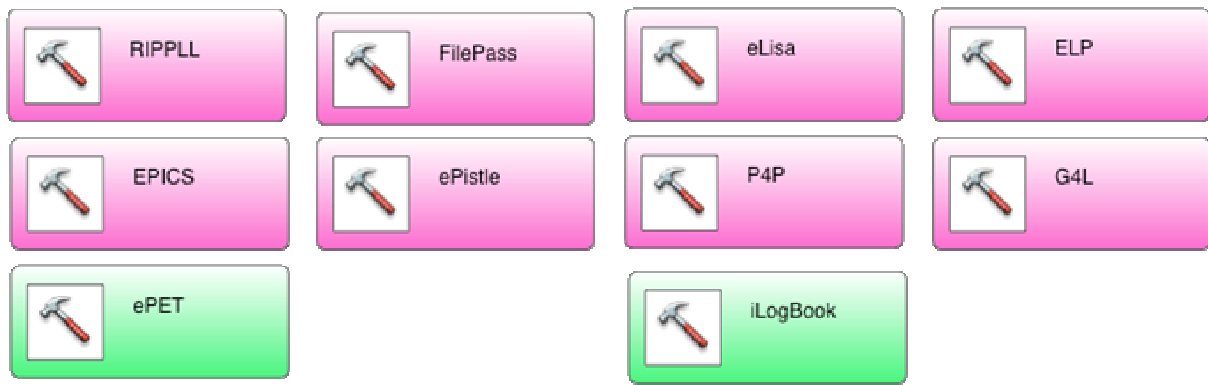
The portfolio cluster of projects is dominated by regional pilots and DEL tools, and show less cohesion than the Assessment and Enterprise clusters.

Toolkits can be leveraged when they help developers to solve a recurring issue; in the ePortfolio cluster, projects were largely concentrating on general IT issues (data integration) and pedagogic

issues (PDP) and had not identified any specific areas requiring technology intervention. There is therefore a general lack of maturity in the conceptual models used in this cluster.
The main success story in this space has been the ioNode set of infrastructure components for assisting the data integration between organisations. However, there is nothing specific about portfolios here, ioNode simply provides an efficient way of managing data transfer and transformation irrespective of the kind of 'payload'.

Current efforts focussed on admissions and assessment may identify common components that might be a subject of future toolkit development. Since user needs are key to portfolio adoption effort maybe best spent at the user agent level and in the areas of identity and personal information management

Collaboration cluster

The Collaboration cluster is very patchy, and has shown no real instances of reuse. While it would appear there is some demand for the ability to reuse collaboration components such as forums, chat and other basic functions, the current deployment frameworks make it difficult to do so in practice. Also, interoperability in this space engages much more at the user interface level than at the services and data level, which makes it difficult to provide toolkits that provide functionality which can fit in a wide range of configurations. Another issue related to this problem is that components at the user level need to be integrated into the general identity and permissions framework; in other toolkits it has been possible to restrict identity to the identity of the requesting "server", rather than individuals - this is not something that is possible at the level of forum, chat and related services.
Future work in this space is therefore dependent on several issues being tackled:

AcademicTalk

RAMBLE

GroupLog

FlowTalk

FROCKLE

- general authentication and authorization, distributed as well as federated

- interoperability standardization at the user interface component level, e.g. WSRP, IMS Tools Interoperability, and W3C Widget specifications in use by the major deployment frameworks such as Moodle and Sakai

Others

Several toolkits have been funded which don't fit into any obvious category. These include PSE, the Portlet Service Embedder, Socket, the generic WS deployment system, and the recently funded MathTran project.

Impact and value of demonstrators

It is easy to measure the impact of toolkits in terms of adoption and reuse. The impact of Toolkit Demonstrator projects is more difficult to assess. Their value has primarily been in terms of the value they have added to toolkits, either in terms of validation of the overall approach, improved toolkit documentation, or improved integration and usability.

Demonstrators have varied in terms of the amount of development work, in some cases **the line between a toolkit and a demonstrator has been very blurred**; e.g. the CATS project, while building on D+ and SPAID toolkits, was essentially a development project in its own right.

Demonstrators were expressly commissioned without any expectation that they would be embedded in production at their own institution; where demonstrators have been engaged in embedding activities these have had very mixed results. While the SLIDE demonstrator eventually resulted in the production use of SLED at Liverpool Hope, at Brockenhurst College the ICT4BIZ demonstrator was halted as a result of encountering organisational resistance. Overall, **including an element of institutional embedding in a demonstrator project incurs extra risk**. Demonstrators should remain detached from institutional requirements to manage project risks and management overheads.

However, demonstrators can provide value in other ways, for example through **integration of toolkits with common platforms** in use in organisations. For example, learning platforms such as Moodle and Sakai, common application frameworks such as Ruby on Rails, uPortal and Zope, and common web applications such as WordPress, MediaWiki, and PhpBB.

We believe that projects can have a significant impact when they **engage with major Open Source projects**. JISC development services such as OSSWatch, JISC-CETIS, and UKOLN can help projects become involved in these developments.

Success Factors

Success Factors for Toolkits

The role of SIGs and social networks
Where high levels of reuse of toolkits has been observed, this has been strongly correlated with a strong social network of developers operating within the JISC-CETIS SIGs. While toolkits are intended to be capable of being picked up and reused on the basis of their code and documentation alone, it is quite clear in practice that a more personal connection between the developers of projects and the developers of the toolkits increases the likelihood of successful reuse.

Sustainability funding
Beginning with phase 2 of the programme, projects were allocated a fund of up to £10,000 for the original toolkit developers to help sustain the codebase. This funding enabled the original developers to be paid to do additional work, such as documentation, training and advice.

In the case of the two Enterprise toolkits the use of this funding was '"considered very successful by the demonstrator projects"' concerned, who felt they had obtained a very good level of engagement and added value from the original developers. From the toolkits side, the funding enabled the

developers - now working on other projects, and in one case another employer - to justify spending time on improving the toolkits.

Overall, this is a low-cost way of ensuring that development resources are available for toolkits, and appears to be successful in practice.

Support from toolkit developers
A major factor where toolkits have been successfully adopted has been **access to, and support from, the original developers**. This has also been a weak point, and problems have occurred when developers have left the sector after the end of the project (e.g. D+, APIS). To some extent these issues can be mitigated by providing improved documentation. However in the wider open-source world unless there is very large-scale adoption of the work, then support from the original developers cannot be replaced by documentation and mutual self-support.

Demonstrator feedback
Demonstrator projects have had a clear impact on toolkit quality over the course of the programme, providing high-quality feedback, and in some cases code contributions to the toolkits themselves. Using a toolkit in the context of a different development team provides much better feedback and quality assurance than is the case where toolkits are reused by the same project team in a different context or project.

Success Factors For Demonstrators

Support from the toolkit developers
Communication with toolkit developers has been an essential requirement for an effective demonstrator. A good working relationship is required to enable the outputs of demonstrators to be fed into toolkit development.

Commonly-used platforms
Using a widely-used platform (e.g. Moodle) as the basis for a demonstrator ensures that the outputs of the demonstrator are easier to understand by the wider community in terms of the evaluating the value added through integrating the toolkit.

Assessing the right level of use
While it can be a fairly simple task to implement a toolkit, its important to pitch the implementation at such a level that it can identify opportunities to improve the toolkit, for example by making use of it for new types of functionality. Both MINTED and JOININ, for example, extended the ways in which Enterprise services were used, which had an impact both on the Enterprise SDK and in engagement with IMS about the specification itself.

Community engagement
Demonstrator projects often encounter integration issues that have been tackled by the wider community before, for example, directory integration and authentication. Projects that engage in the community via SIGs have been able to tap into that expertise and avoid common pitfalls. Within the programme several demonstrators have been successful in operating across two or more SIGs (e.g. SLED-DLD, CATS).

Gaps

As things currently stand, the coverage of **e-framework service genres is very comprehensive**, and very few gaps can be identified in terms of toolkits developed or currently in-development, particularly as **where gaps have been identified these have been filled by projects funded in other programmes**.

Clearly not all toolkits have achieved significant take up within their area of coverage; however, there are a number of factors responsible for this including maturity of the sector as well as the way the toolkit was presented and marketed. There also appears to be a factor in terms of **platforms supported**; the work to date has been overwhelmingly focussed on SOAP services developed in Java, and uptake may be increased by widening the supported platform base for toolkits, specifically supporting REST-based services using languages such as Ruby and PHP.

Overall, **the 'hit rate' of the initial programme was very high** and the coverage very broad; the next phase should be more focussed as **the most obvious service genres have already been targeted** for development effort.

Rather than fill gaps in coverage, the most practical near-term strategy may be to **build upon previously successful toolkits**, particularly adding multi-language support and platform integration, and to continue to **fund demonstrators, particularly those that feed into major open-source platforms**.

It may be the case that with some toolkits that its simply a matter of time; for example it took several years after initial development for the Enterprise toolkits to reach peak usage.

Issues

A number of issues have arisen over the life of the programme:

- There has been the recurring issue of **documentation quality** for toolkits. However a process of peer review and the demonstrator projects have improved the quality of documentation.

- The **funding and proposal management lifecycle** has been an issue, in that the amount of work required for project management has been disproportionately heavy in relation to the levels of funding. This is one of the disadvantages of having much shorter six month projects.

- There is an issue of **balancing short-term intensive work and longer-term commitment**: toolkits require a fairly short development time followed by an unknown period of low-intensity support and maintenance. Mechanisms for better supporting this need to be investigated. So far the use of sustainability funding has been useful, but it difficult to always predict the levels of developer engagement required.

- There have been problems **retaining personnel and expertise** within the sector for particular toolkits. It suggests that where possible risk should be spread by trying to widen the pool of people working on toolkits beyond the initial developer.

- In some cases the adoption of toolkits has suffered as a result of **lack of awareness**. The sheer volume of projects funded through the e-learning programme has made it difficult for development projects to be aware of each others work and therefore be able to identify parallel or complementary work. The SIGs could play a **pivotal role of SIGs in facilitating cross project awareness.**

- There has clearly been fairly patchy success in terms of **assessment of community readiness** for uptake of a new toolkit. However, the overall hit rate has been high and this suggests the degree of risk-taking in the programme is about right.

Evaluation of the overall approach

Overall, a simple review of reuse patterns within JISC projects shows that of the 22 toolkits funded under the first two phases of the programme (that is, excluding newly-funded projects):

- 5 Toolkits were reused by at least 3 other projects
- 5 Toolkits were reused by at least 1 other project from a different institution
- 2 Toolkits were only reused in other projects from the same institution
- 10 Toolkits were not reused by other projects

Overall this shows a 'hit rate' of 50%, which we believe is a good return at the level of investment supported. It means that around 20 projects were able to make direct use of prior work in the form of running code.

This is a high level of reuse when you consider that the programme was intended to fund development which that would not otherwise take place as its too risky in production.

If we were to assume that the cost of repeating the work is comparable to the effort in developing it as a toolkit (e.g. 6 months developer time) then the overall cost saving would be approximately 10 FTE (about the same level of resource cost as the complete range of projects funded), *'purely in terms of savings made within the projects themselves*, and not taking into account savings made in the wider sector by non-project usage or added value in terms of quality and capability.

Additionally, the 'top 5' toolkits continue to provide a return as they continue to be reused in new projects. It is also likely that they will at some point become incorporated into common platforms such as Moodle and Sakai.

Recommendations

General points from an evaluation of the approach

The toolkits and demonstrators programme has shown some real benefits.

- *Facilitating standards uptake*: This has taken place for Enterprise, SRU/W and QTI, and to a lesser degree Learning Design.

- *Sharing and reusing development outputs*: There was clear evidence of this amongst JISC-funded work, with **25% of toolkits being reused by 3 or more projects**. In terms of

external projects and products, **considerable value has been added to existing OSS systems** (Moodle, Boddington and Sakai). This was probably a better value investment than direct funding of those OSS projects as development effort has targeted e-Framework priorities.

- *Reduced cost of implementation*: There has been **evidence of cost reduction for JISC-funded projects** as a result of reusing toolkits.

- *Spreading risk*: The programme has **identified high risk areas of insufficient maturity at relatively low-cost**.

- *Enabling innovation*: **Toolkits have enabled institutions to undertake more advanced work than would otherwise be feasible in short projects**. For example, the use of Learning Design in production use at Liverpool Hope was only possible using SLED. The Making Tracks project was only possible due to the previous development of ISIS and ASSIS. The Enterprise SDK enabled experimentation with and enabling of dynamic and ad hoc grouping for pedagogic purposes at the Open University by solving the immediate issue of service operation.

- *Enabling a service-oriented approach*: E-learning **SOA development capacity has been established** in the UK through this programme, and has seeded a number of first production web services in institutions.

- *Piloting a development approach that works with existing systems and vendors through separation of integration, implementation and user experience*: Developers have adapted quickly to a new development approach that separates concerns; development has **offered added value to existing platforms rather than created competing, probably unsustainable, solutions**.