Acknowledgements

The CTREP project is a JISC Repositories Enhancement Project. The project team at CARET would like to acknowledge the contributions of partners and collaborators, both formal and informal: Cambridge University Library, University of Highlands and Islands, Teaching for Learning Network, Learning Landscapes Project, Catherine Howell, Helen Burchmore, Ingrid Birchell-Hughes, Katy Cherry, and various members and fellows of the Sakai project, particularly Nico Matthjis and Ian Boston. Finally, we also wish to thank Andrew McGregor, JISC programme manager for his assistance in the course of the project.

Executive Summary

CTREP created a connector between an Institutional VRE and an Institutional Repository. It is designed to be reusable in a number of different institutions where policy on deposit varies by means of a flexible deposit configuration system. In the process of executing the project:

- the various stakeholders came to understand institutional cultural differences and address them in such a way that recent projects with a strong Repository and research
dissemination/visualisation aspect have been more joined up than would previously have been possible

- we developed an approach to policy expression designed both to avoid creating unnecessary tension within the institution during its development, and also to be authorable by a wide range of individuals
- we have sought to record and capture lessons learnt (based, in part on case studies) for future institutionalisation projects
- we developed a number of techniques which allowed apparent barriers to integration to be overcome by technical-architectural tools
- we open-sourced the integration
- we modified our approach to metadata/data binding in light of community feedback and developed a spreadsheet-based automated approach with which contributors felt comfortable, but which required a number of technical obstacles to be overcome through the use of creative programming techniques.

We retained management and liaison staff who communicated, educated and supported Institutional Repository adopters

The institutions involved are committed to continuing to expand upon the work of the project.

**Background**

CTREP sought to establish and enhance good-practices in the management of the Institutional Repositories of the partner institutions, through various technical and non-technical means. The principle aims of the project were to increase the quantity of ingested material, and general satisfaction of users with the Institutional Repository, through improving ease-of-use, and increasing integration with other institutional systems, including established Virtual Research Environments within the institution, (each based around Sakai).

The principle barriers were anticipated to be cultural. As a recent development, VREs tend to be operated and managed within University innovation units with a strong focus on rapid development and experimental practice. Digital repositories, and therefore the software which facilitates their operation, operate within a library context where an important focus is upon preservation and sustainability. Further, academic culture is again distinct from these environments, and varies both between and within disciplines.

With such a wide variety of culture and priorities, the project sought to implement a number of technical interventions within the institutional infrastructure of the partners, in order to investigate these issues, and to establish practices, both technical and non-technical which sought to overcome these barriers. To ensure that the experience so gained was not lost, the project defined, as a deliverable, a document to communicate these lessons.

**Aims and Objectives**

In broadest terms, the aim of CTREP was to increase the institutionalisation of the institutional repositories at the partner institutions, in order to better preserve and disseminate the digital material created by those institutions.
In narrowest terms, the project aimed to develop a technical integration, in the form of software and machine-readable description, which connected an institutional VRE based upon Sakai to the Institutional Repository to ease the repository document ingest process.

Naturally, the majority of the enterprise would lie between these two extremes:

Before the software could be developed, a quantity of enabling work was required. For example, before policies could be examined for machine reading, they had to be fully written, and before user-input could be used to enhance the software, careful nurturing of institutional relationships was required to ensure that such associations could be productive.

As the software was developed, and iterations undertaken, the team needed to engage in-depth with academics undertaking relevant research projects, becoming familiar with their material and practices.

Alongside the software development, CTREP employed management and liaison staff who communicated, educated and supported Institutional Repository adopters.

This engagement was to reveal opportunities for institutionalisation which would refine and extend the initial remit. An important goal of the project was to be that these 'soft' lessons learnt in the project were not lost.

**Methodology**

**General Approach**

A key steer throughout the project was that the CTREP team were to remember that CTREP was not primarily a software project, but one of institutionalisation: embedding an Institutional Repository within an institution, including its institutional IT provision, with a primary focus on the Institutional VRE.

Camtools, the institutional VRE of the University of Cambridge, incorporates research data, and is a focus for the daily work of collaborative research projects. Through developing a common VRE/IR offering, we could strengthen both the Institutional Repository and VRE and embed each more fully within the institution.

The challenges were not primarily technical in the sense that they did not focus on the difficulty in moving information from A to B, as the protocols and standards to perform these tasks largely pre-existed, and 'only' adaptor work was required.

Instead, the challenges focussed around building collaborative relationships across working cultures within the partner institutions and, in the longer term, allowing any outputs to be applicable to other institutions where there will be further differences in policy and culture.

In order to help overcome these barriers, a number of technical tools were developed, described elsewhere in this document, to help bridge these gaps. However our focus was to consider these developments, where they proved to be appropriate, as a means to an end.

**Tactics**

First, issues around the size of skill-set pools in the wider community and employment market cannot be ignored when considering developing an institutionalised system. Whilst the team had broad experience in a wide variety of tools, including specialist repository experience, it was seen
as important to acknowledge that a sustainable offering should probably base itself around skills which are widely available in the general IT and repository community. Therefore, specialist, more closely-fitted technologies were rejected in favour of more ubiquitous ones.

Secondly, We identified a risk, early on in the project, that if a technical solution was developed early and then offered to users or repository staff, who were then asked “what is wrong with it?”, “how might you use it?”, or “please use this tool to express such-and-such”, the technical team ran the risk of driving our investigation of institutional procedures according to pre-conceived ideas. It is well known that institutional IT offerings are often worked-around or ignored because they fail to represent institutional processes accurately.

To avoid, this we adopted the blank page principle, whereby we did not offer the investigating user any input on the appropriate tool to use to achieve a task. We took their offerings and then tried our best to represent their chosen tools and techniques in the software. This is evidenced in a number of place in the project, and steered it in directions which, we strongly feel, would otherwise have been neglected.

For example, the technical team did not provide an initial structure for the natural language embodiment of policies by the Institutional Repository team. In doing so, the finally delivered policies were revealed to be too complex for our initial hypothesis as to an approach for technical embodiment. If we had given the Institutional Repository team a template, we believe they would likely have been delivered a “dumbed down” version of the policies, appropriate to the scheme offered, so as to avoid the possibility of failure. Again, this would be working around IT, not genuine integration to the institution.

Similarly, when we asked a team to develop and deliver to the technical team a summary of the metadata within a research set, we did not specify a tool or schema. What was delivered, a curiously structured Excel spreadsheet, led to a discussion as to reasons for decisions, and drove our technical development down a novel route which is currently in the early stages of being rolled out more widely across the University.

Thirdly, it was seen as important to curtail discussions which could otherwise consume the whole project. Members of the project team have seen various other projects sunk by interminable discussions of relatively few ‘hot’ subject areas. We proposed that we would investigate technical means of curtailing these discussions, through features embodied in the software, in order to keep the project on track whilst keeping collaborators content. We described such features as “firebreaks”.

For example, the issue of metadata crosswalking can be guaranteed to lead to long discussions in almost any circumstances. Our naïve initial take on this issue was that a large number of use-cases could be solved by adopting a simple, lowest common denominator mapping to a ubiquitous metadata standard, such as Dublin Core.

However, we soon realised, even if this was the case (which we did not determine), that this would not solve the issue of long crosswalking discussions. Instead we added a “firebreak” which allowed customizers to alter fields and to crosswalk their metadata. This addressed the issue of the perceived need, curtailed any unenlightening discussions which might otherwise have occurred, and the question of true need became moot. The users have the ability to crosswalk metadata in an extremely extensive way (with all the power of a full programming language) and may use it if they desire.

Fourthly, it was clear from the early stages of the project that the VRE and IR groups should present, wherever practicable, a joint offering to the institution, hiding the artificial barriers of
different software applications and management structures.

**Dissemination**

The lessons which we were to learn during the progress of the project would be valuable to other institutionalisation projects in the broader community. However, it was clear to us that the communication of those lessons would be challenging. We therefore retained a professional writer to summarize the experience and to produce a document for a general audience, separate from this formal reporting process. This document is written and currently in pre-press.

**Implementation**

**VRE/Repository Integration**

**Designing to Address Cultural Concerns I: Configuration and Control**

Repository managers are rightfully concerned with oversight of the ingest process, particularly in gathering appropriate metadata for long-term preservation and broad dissemination. Because the mandate and material of each institution is unique, it is very likely that the information required for appropriate implementation of ingest policy decisions will vary between institutions.

The technical challenge arising from this observation becomes: how can an institutional repository maintain control over this ingest process when external methods are used to collect and ingest the data?

This externalisation process created significant worry within the partner institutions, but the project determined that this concern would be massively amplified when developing a technical solution which could be deployed on multiple sites, or survive policy changes which are bound to occur within any organisation in the long-term.

To this end, configurability was made a focus of the technical development. A service was integrated into the software whereby the user-interface code, when rendering, could consult a remote schema, under the control of the Institutional Repository, which determines the information to be gathered, messages to display, and so forth.

In addition, a scripting language was introduced which allowed extremely fine-grained control over the behind-the-scenes progress of the ingest process, allowing per-user customised interfaces, routing of data and messages on the basis of supplied metadata, and so on. This allows a repository to maintain fine-grained control of the ingest process without high-level, hard-to-acquire development skills, and allows their policies to be embedded in the operations of the ingest process. These concerns are anticipated to increase as the wider prevalence of diverse ingest methods increases the variety of material submitted for deposit.

**Why Scripting, and How?**

A number of successful applications use scripting to achieve configurability, though the terminology used varies widely. Some applications, such as Excel, refer to scripting as *macros*, and these scripts are typically very simple, though macros exist to solve a wide variety of complex tasks. Other applications, such as the graphics package Processing refer to themselves as *development environments* and describe the process as being akin to programming, albeit with
programming primitives more commensurable to the task in hand.

Within this range of scripting environments, a broad variety of languages are used (and at one time it seemed almost mandatory that each significant application developed its own).

Mail Transfer Agents (MTAs) make a good exemplar of the task in hand for designing a customizable ingest process as the concerns of their configuration, metadata, routing, accepting and rejecting, and so on, are a good model, in an information-science sense, for the embodiment of policy in an ingest process.

Unfortunately, common MTA products do not address this situation satisfactorily, and their baroque configuration process is a common concern of MTA administrators. For many of these applications, script-based configuration evolved out of increasingly convoluted corner cases applied to regular (declarative) configuration until, almost by accident, the configuration became akin to a programming language. A syntax 'evolved' in this way is almost always hard to comprehend and manipulate. Such examples have therefore given scripting a bad name in such contexts.

In order to address these concerns, the project determined to use an existing, popular scripting language and whilst not hiding its essentially programatic nature, lowering the bar as much as possible.

We wished barriers to entry for configuration to be sufficiently low to allow an important subset of software tinkerers and amateur programmers to meaningfully engage with the language: those with expertise, and professional credentials in a relevant domain.

Someone who is an expert in information science, with an aptitude for things computational, can achieve a lot which might otherwise be lost in the communication process between domain-professional and professional developer. Even if the product so-created is not perfect, it serves as an excellent expression of intent, or specification, for a professional programmer.

In order to achieve this end, the choice of language was crucial. The project ultimately settled on javascript.

The pool of javascript programmers is massive, and increasing, including many who would not consider themselves to be programmers. The language itself is broadly conventional, and so would be familiar to almost anyone with relevant programming experience, and there is a large quantity of documentation available for the javascript, both in print and on the web. In terms of functionality, javascript is a moderate fit with the task in hand, though not as perfect as you might expect for a domain-specific language, and good interpreters are available for it in multiple languages (the project used the Mozilla Foundation's Rhino for Java).

It is important to emphasises two things. First, the language was chosen almost entirely because of community issues relating to popularity, etc., and would have been low down a list ordered by inherent suitability of language features. This focus on the social aspects of the decision matched the high-level focus of CTREP on the cultural aspects of institutionalisation.

Secondly, whereas javascript is familiar as something which executes within a web-browser to accomplish visual and other effects, it is important to emphasise that this is not how it is being used in this case. This javascript is run in the server, and is not based around manipulating an HTML document, nor the use of web-protocols.

**Designing to Address Cultural Concerns II: Ease of Collection**

Having addressed a primary concern of the repository community through configuration and
scripting, our attention in later iterations of the project turned to metadata collation methods.

But this aspect of the project came as somewhat of a surprise, and challenged our initial assumptions, following engagement with user communities.

It is the orthodoxy of the repository community that an ingest must be as transparent and simple as possible for the creator because of a lack of time and/or inclination to engage with the deposit process. A large number of studies have confirmed this finding, which is, no doubt, prominent and important for the vast majority of depositors.

However, we have discovered areas where this assumption should be challenged.

Because of time and budgetary constraints on the project, we focused on areas where we felt we could 'win' a large quantity of data for little outlay in terms of community building, and so forth.

Social science data-collection and analysis research projects were ideal for this purpose: they generated large quantities of information but, unlike the typical case for scientific datasets, this data bore a close resemblance to the data which might be expected of ad hoc depositors: academic papers, posters, transcripts, instruction guides, and so forth.

In this context, we engaged particularly with the Learning Landscapes Project and Teaching for Learning Network, by developing existing relationships to coax the team into extensive participation with CTREP. Initially we suggested a number of technical methods by which we could arrange for the software to induce information concerning the authorship, title, type, provenance, rights, and so on for the document, supposing that this would lead to a valuable time saving.

However, what we discovered from observing the researchers was a genuine engagement with discovering and cataloguing the information corresponding to each of the pieces of data within the project, so as to produce a catalogue, as an intellectual output of the project in its own right.

Therefore our focus shifted away from software which attempts to automatically induce metadata from data, to an examination of the methods of organisation which came naturally to identified pilot projects. To this end, we worked closely with Helen Burchmore of TFLN, and consulted on the experiences gained from a number of completed projects (such as QUADS) who have undertaken this cataloguing task in other contexts.

**Excel-Based Collection**

For the projects selected, a natural early step in data collection is to prepare an Excel spreadsheet describing their data.

Typically some considerable initial start-up time is spent by the project describing this spreadsheet, and the columns it might contain, the constraints upon its values, and so on, in consultation with a number of project partners. After this definitional task has been completed, the spreadsheet is let lose on cataloguers and researchers.

A number of software solutions exist which might seem, in some ways, to be superior to Excel. Management Systems exist for Documents, Collections, Content, and so on, which allow rich interaction between the user and the data, and fewer opportunities for errors than with a spreadsheet. However, in terms of user-comfort, community size, and ubiquity, Excel (and alternatives, such as the OpenOffice spreadsheet application) have no rival.

This choice creates considerable challenges for interpreters. The excel file format is proprietary and complex, and researchers commonly convey information using features (such as cell-merging) which are not well-embodied in simpler, open equivalents (such as CSV files). Also, without
constrained vocabulary, typographical and formatting errors are common in Excel files. However, in accordance with the general tactical methodology of CTREP, we decided to “bite the bullet”, and spent a considerable amount of time and effort in interpreting excel data without requiring a change in practice.

First, we identified and implemented an algorithm to interpret cell merges and blank cells in the context of the other fields in the spreadsheet, to determine the meaning of blank cells (principally distinguishing “ditto” from “none”).

Secondly, we used a technique which we have named “cheat-based reasoning” (see below) to use the information which is available to us in the source file-system (within the VRE) to handle formatting errors in identifiers.

Configuration for these algorithms, and associated metadata are encoded in a second sheet in the workbook with a predefined name (“Archive”). The data stored there (such as mapping of columns to Dublin Core) is easily enterable at the extensive spreadsheet design stage by non-experts.

Ongoing development includes defining, with all parties, a spreadsheet which would be a usable default, across the institution, for common cases.

Cheat-Based Reasoning

A major barrier to using an excel-based metadata collection method for automatic ingest from a VRE was the issue of identifier mismatch in coupling metadata to data.

We addressed these issues using a technique which we named “cheat-based reasoning”.

The identification of data to ingest will, in the general case, include a number of mistakes and formatting issues.

Our initial approach to resolving this issue involved attempting to define, as with the cell-merging case, an algorithm to apply appropriate corrections to the data, based upon observation of practice (such as the fluid interchange of hyphens and underscores during transcription). However, the mistakes and changes were too wide-ranging and variable for this technique to work.

It is a well known principle that it is simpler for an algorithm correctly choose from a small set of alternatives than a large one, as the proportion of the configuration space which maps to each alternative will be larger if the number of choices is smaller.

Therefore, we developed “cheat-based reasoning”. At each stage of determining an identifier, the VRE was contacted prior to the determination, and asked to return a list of available options, so as to reduce the available choices. Then a matching algorithm was repeatedly applied with increasing degrees of laxness until one match succeeded. By downloading the “cheat-sheet” we turned something once equivalent a “short answer” question into something closer to a “multiple choice” problem.

Scripting Plugins

Despite our observations, it seems likely to the project team that, when engagement broadens and deepens sufficiently, the issue of deducing metadata from data will reappear. Relatedly, it also seems clear that the spreadsheet system developed for Teaching for Learning Network will want to be extensively customised by other deploying institutions.

To address both of these issues, we used the scripting configuration principle outlined in earlier
sections, and extended it to include the idea of a plugin. Rather than apply the spreadsheet process as a built-in or fixed-configurable part of the ingest process, the procedure was wrapped in a simple API and made available to script writers as a javascript library. This allowed the authors of ingest-procedure scripts to determine when the technique was or was not appropriate to use on the basis of context, and how it should be applied.

Automatic metadata extraction will be presented as another such plugin, which will be applied, where appropriate, to documents in contexts for which it makes sense.

All this could be driven, for example, by a user selecting from a new dropdown in the UI widget (made possible through our schema mechanism) between “Academic Paper”, which might undergo automatic extraction, “Excel-Controlled Research Project”, which would undergo spreadsheet extraction, or “Video”, which would undergo neither.

Early Experimentation with Policy Expression

We entered the project with the working hypothesis that the embodiment of repository policy would be essential component of successful institutionalisation. To this end we worked with the PLEDGE project at MIT, which aimed to use a semantic representation of policy to facilitate, amongst other things, federated storage beyond institutional boundaries.

The starting point for both projects were RDF-like languages and reasoning systems based upon these languages, such as RDFS and OWL. The PLEDGE project eventually opted to use the REI policy expression language.

In line with the methodology of the CTREP project, we asked that the members of the team responsible for developing repository policy do so in isolation from technical concerns, in order that they were not unduly constrained by technical choices. This, naturally, had the satisfying spin-off benefit for the institution that such policies were written and made available to the University through the Institutional Repository website.

A technical member of the project team then endeavoured to represent those policies, as best as possible, in machine-readable form. The project instantly hit problems which fundamentally questioned our use of the declarative approach, and which, it was determined, change of the choice of policy language could not hope to resolve. Principle amongst these issues were

- Whilst policy expression languages are often vaunted as being “not programming”, with the implication that less training or experience is required to author or read them, we found the opposite was the case. Beyond the baroque syntax of these languages, it became clear to the technical team that an embodiment of a complex policy in a machine-readable form required a level of familiarity with formal logic and machine reasoning which lay beyond the capabilities of the vast majority of those who were asked to interact with these languages. In fact, it seemed highly likely that considerable programming ability (perhaps Computer Science to at least Masters level) would be seen as a prerequisite to writing complex policy embodiments.

- The extreme complexity of organisations leads to definitional issues in using a logic-based approach. How can a complex institution, such as an ancient University composed, as it is, of many departments, colleges, and international collaborations, be efficiently represented in terms of predicates and rules?

- The very nature of the organisation is often a subject of deep political debate, or else it is moot, to avoid such debate. Many attempts to describe roles precisely, without the
comforting vagueness available in natural language, risked reinvigorating a large number of points of contention, each of which which remained unresolved despite the attentions of great minds over many centuries. In essence, this process would stir up a hornet's nest.

**Investigating Policy Expression Issues**

The PLEDGE project is widely seen as successful, and we evaluated its methods and outputs similarly. Therefore, the question was raised as to how PLEDGE managed to avoid these issues which seemed insurmountable to CTREP.

First, our projects differed considerably in scope and purpose. Whilst both addressed issues of machine-readable policy embodiment, the purpose of such embodiment differed greatly. For CTREP the priority was institutionalisation of the institutional repository; for PLEDGE the focus was on large scale federated repository infrastructure, and the issues surrounding custodianship of another institution's data.

One approach which we considered, and which we believe proved valuable for PLEDGE was to “duck-type” our definitions. “Duck typing” is terminology used in programming language communities to describe an approach to resolving the issue of what “type of value” some data has.

The “duck type” approach recalls the adage “If it walks like a duck and quacks like a duck, I would call it a duck”: i.e. it focuses on the properties of the data in the context of its use rather than attempting a more abstract, ontological approach to its definition. In the context of policy expression, we might define a type of institution as an “able to ingest institution”, or declare “legal deposit interest” for a user without reference to the terminology of the University used to define such individuals and institutions.

Such duck-typing was ideal for the PLEDGE project, as the focus was around operationalising rules at a remote institution on the basis of local policy. In the case of the first integration for PLEDGE, the rules were operationalised in iRODS at the SRB of the San Diego Supercomputing Centre.

However, CTREP’s focus differed: the transmission of policy between federated institutions was not as important as the apparently simpler task of developing tools to express and embody policy within a single broadly-coherent institution.

We had a clear choice

- **either** develop infrastructure to allow the easy expression and discussion of machine-readable policy in a manner directly analogous to the natural language policies developed by the team
- **or** investigate operationalised, imperative policy representations directly.

As the former option seemed unrealistic in the current state of the art, we opted for the latter. Fortunately, in investigating the iRODS language into which the PLEDGE REI is compiled, we discovered it to be close to an event-driven imperative language.

At this point it became clear that such an imperative approach might be more appropriate to CTREP because

- though “it is programming”, it is programming in the sense that many of “non-programmers” undertake everyday, writing scripts and macros to help configure large office applications.
- a fully operationalised approach would allow us to focus on “what happens”, not “what it
is”. As the former is a matter of fact, rather than speculation, and as “what it is” in corporates a large quantity of contentious “what should be” questions, we anticipated that sticking to “what happens” would tend to present fewer conflicts.

Rather than develop our own policy language, or use a niche language from a specialist domain, to maintain the force of the first of the bullet points above we chose a common, event-driven imperative language, javascript, as our means of embodiment.

**Prototype Implementation**

In contrast to the broader architectural issues described above, the prototype integration was developed as a platform for experimentation (for example on policy expression) and discovering and overcoming technical integration issues around protocols, and such-like. This prototype, developed around the Content Hosting Handler system in Sakai, faced a number of issues.

The approach was, in some sense, almost too transparent in that it did not provide ample opportunity to interact with the user to provide information necessary for successful deposit. Also, during experimentation, we discovered that the only options available for metadata entry using Content Hosting Handlers, the Sakai metadata system, is poorly accepted by users, and generally neglected.

However, this prototype system did provide a number of successes. First, it allowed us to investigate a declarative approach to policy expression (which was rejected in favour of an imperative approach, as described above). Secondly, it allowed the project to investigate wire protocols, such as SWORD and LNI to submit patches for some issues within their implementations, and begin the negotiation process of Institutional Repository connection.

**Primary Implementation I: Server/Client/Server**

One of the major technical and organisational issues with developing software for institutional integration in a rapid development cycle seems to be creating an environment in which experimentation and update can take place simply and quickly, but without destabilising the core system.

[The following description involves some technical depth, but we believe describes a valuable and broadly-applicable technique].

In the case of this project, this issue is particularly evident in its VRE integration, as it is the VRE which includes a user-facing UI component. But, as well as customising its user-interface to present ingest options to a user, the developed technology must also query the VRE for identity information, for file contents and for directory listings for “cheat-based reasoning” (described earlier).

Fortunately, Sakai is moving in a direction which helps provide a solution to these stability-disparity issues. In adopting a rendering model which is more focused on the client-side, the Sakai server-side now provides a number of “feeds” of raw information in JSON format, and also makes raw data blobs available to the client through a REST-like interface. This allows UI components to be developed and installed on the client side without destabilising the underlying server-side data architecture.

In our case, however, we were not executing on the client side. Current wire protocols to connect to repositories are largely too complex to implement exclusively on the client side, and the issue would anyway remain as to the appropriate place to store and execute scripting and schema
Instead, to connect to a repository, the server-side adaptor was placed on a non-enterprise but managed server to which the project had access, and for which a hole was punched in the VRE url space using Apache's mod_proxy. This allowed the server to 'appear' at a URL which seemed to the client to be a part of the server-space of the VRE. The client could then communicate with the repository, and be served with schema information, as if a the server were a service of the VRE, whilst maintaining minimal alteration to the VRE to avoid destabilisation (requiring only the addition of the proxy rule). This risk of destabilisation occurred as a repeating concern of the VRE team when discussing integration in the early stages of CTREP.

However, file and identity information were needed by the adaptor from the VRE, but such information was not easily available to the client (though the client could have been made to work as a proxy this would have been highly inefficient, made the client highly complex, and subjected the process to security vulnerabilities). Fortunately, because the adaptor server appears as if in the URL space, the client will always send session information to the adaptor server, which the adaptor can then use to retrieve information from the VRE as if a client.

Therefore, experimental services can be integrated with a production VRE, without destabilising it, which give access to all data feeds available to a client.

Illustration 1: Server/Client/Server architecture to address stability concerns.

Primary Implementation II: Parallel Repositories

A parallel institutionalisation issue arose when integrating the adaptor with the institutional repository. Because of stability and scalability concerns, our repository managers were understandably wary about integrating with a new ingest protocol. LNI proved unstable, and SWORD was immature at the start of the project.

With user-acceptance of the project's ingest method unproven, this challenges the update policy of the Institutional Repository.

After discussions, this was resolved by use of a shadow staging service. The shadow service is an installation of the institutional repository which resembles the Institutional Repository, but with the
necessary additional features installed (SWORD ingest, in this case).

Rather than being an alternative to the Institutional Repository, however, the Shadow Repository alerts members of the team when content has been ingested. This data is then transferred, through a manual process, to the Institutional Repository by members of the IR team.

The migration path for this setup then becomes

- initially the Shadow Repository is hosted by the project team
- then the Shadow Repository is moved to being hosted by the Institutional Repository team, once uptake reaches a predefined level,
- finally the functionality of the Shadow Repository is integrated with the Institutional Repository itself when update schedules allow, and the Shadow Repository is decommissioned.

**Illustration 2: Using shadow repositories to ease deployment concerns.**

**Primary Implementation III: Embodiment**

Internally, the code for the adaptor is structured to allow maximum flexibility going forward, despite the possibility of integration targets changing. In particular,

- the VRE integration code has been separated from the body of the application behind an API, to allow code to be easily developed to target another VRE technology.
• the repository integration code is similarly separated from the application body by an API, again to allow code to be easily developed to target another repository technology. In this case, this switchability has already been exploited: there are both Fedora and DSpace embodiments of this API.

• The target Repository and VRE implementations to be used are easily configured from the configuration system. In addition, test implementations of each API allow the system to be developed and tested without connection to complex repository VRE/Repository systems, which helps isolate the component being tested.

• Complex interventions, such as spreadsheet data extraction, are separated from the body of the code by means of a primitive plugin system, which allows their use to be carefully configured in an institutionally appropriate manner by system configurers.

• The ubiquitous Rhino javascript engine has a number of well-known flaws, and competitors are emerging to challenge its dominance. Therefore, we abstracted away the Rhino API behind a generic javascript API to allow easy replacement at the appropriate time.

A number of convergences simplified our implementation. For example, the use of JSON in data-feeds is compatible with the data representation used by javascript (being derived from it), minimising the translation steps required within the application. The convergence of “Web 2.0” technologies over recent years to common standards and methods of operation (such as the use of JSON, client-side rendering, AJAX, and so on) was a major simplifying trend in the low-level execution of this project.

However, in places differences between protocol approaches made abstraction behind a common API complex.

For example, the dominant means of transferring data to Fedora via FoXML is to make this data available at a URL, whereas DSpace data is presented within a packaging format. If the data for a FoXML ingest is not open-access at the time of deposit, this can become an issue. Therefore, in order to hide this protocol difference behind a common API, we developed a temporary, security-controlled store which allowed data to be stored whilst the Fedora ingest process was underway.

Such issues of maintaining uniform interfaces to inherently distinct processes became one of the major low-level technical challenges to the project.

Enhancements and Changes

As described in earlier sections, a number of enhancements and changes to the originally envisaged product were incorporated during the progress of the project, responding to feedback:

• An implicit approach to deposit user-interface was replaced by an explicit (widget based) one, as this allowed the repository to maintain more input and control over the ingest process.

• A visually-rudimentary (but powerful) administration interface was created to allow the repository to maintain control over the ingest processes and policy embodiments.

• Scripting was introduced to allow fine-grained control over the operation of ingest and to embody policy execution.

• As described above, after careful examination, we dropped the initial goal of declarative policy embodiment, and replaced it by imperative embodiment, as a simple, powerful
approach.

- Following discussions with researchers, we replaced our intention to support automatic extraction of metadata from data, with support for processes which allowed originators to easily explicitly tabulate the information. We envisage that automatic extraction will re-emerge as a priority going forward.

- As UHI found themselves unable to contribute to the final stages of CTREP, initially support for Fedora was dropped. However, it soon became clear that Fedora is of increasing interest at the remaining institution. Details of the new relationship of CTREP to Fedora are given in the following section.

CTREP and Fedora

As mentioned elsewhere, CTREP maintained an interest in Fedora despite the partner institution which used Fedora as an Institutional Repository regrettably having to withdraw from the project. The retained interest was a function of the long-term goals and prospects of the remaining institution.

First, DSpace 2, a natural successor to DSpace, will support a number of switchable content stores, of which the Fedora store may be amongst the most compelling: Fedora has shown itself to be stable and scalable to large content stores and has a flexible associative model for complex datasets.

Secondly, a number of research groups with complex data within University of Cambridge are already using Fedora to maintain their project data, separately to the primary Institutional Repository. In addition to its richer expression of associations, Fedora also has stronger APIs than DSpace 1 which allow these research teams to more easily “mash-up” the data they contain, and to present and analyse it in novel and informative ways.

The maintenance of these separate per-project repositories, whilst compelling in their own terms, represents a risk to an institution's archiving policies. These project repositories are often installed in an unmanaged and ad hoc manner, which do not represent a sustainable archive strategy for the data.

Some support for connecting to a Fedora-based repository, or similar technology, was therefore seen as an important goal for the CTREP project, so such a connector was implemented. A future focus is to extend the configuration system to include policy-based switching of destination repository, based upon input and context.

We would be interested in hearing from groups which have institutional Fedora deployments who wish to co-operate on developing this Fedora Integration further.

Proposed Enhancements and Changes

The outputs and processes begun under the auspices of the CTREP project continue at University of Cambridge. We have secured informal commitments from the major partners to continue with developing the project deliverables either from existing funds, or else from further bidding. In particular we are investigating:

- How to further improve the visual appearance of the integration both to the user and to the administrator.

- Adding further hooks into the scripting system to allow further opportunities for customization and reconfiguration.
• Determining what further testing and refinement (if any) is needed to allow marketing at various levels within University communications without overburdening support infrastructure.

• Ensuring all technical knowledge gained in the project is embodied in documents, not merely “in the heads” of the project team.

• Developing institution-wide spreadsheets, based on the spreadsheet ingest method, to allow an off-the-shelf solution for the ingest of social science datasets.

Developing Common Practices

Privacy Policies

One set of policies addressed by the project were those concerning privacy. Both the VRE and Institutional Repository have privacy policies around such issues as identifying information, unsolicited email, and so on. Each was developed independently and, (largely through accident and strategically irrelevant differences), seemed to prevent data sharing.

Through a series of rather intense meetings and email correspondence, the project eventually managed to develop a means of co-operating which led to a more “joined-up” approach to privacy at the institution.

This had unintended positive side-effects described in the sections below.

Metrics

The institutional VRE actually operates in the context of a combined VLE/VRE offering without formal distinction between roles, as is seen to be appropriate at a research-led University. Further, the use of the VRE varies greatly in terms of its role within any given research project. It became clear to the project that it was vital that we developed metrics to distinguish between these uses, to be able to focus on those uses with strong repository use-cases.

Through meetings with the VLE/VRE team, and enabled by the privacy policy discussions describe above, the project developed heuristic metrics to determine the use to which a VLE/VRE site was being put based upon its state and events, in order to focus our efforts upon relevant sites.

This had the side-effect for the institution of being useful in VLE/VRE planning and statistics gathering. It also led to further side-effects which have proved valuable to the project, described below.

Joint Marketing and Operations

As a further step in the chain, the project realised that one of the impacts of a common understanding of privacy policies, and algorithms for metrics within the VLE/VRE, was that email marketing of the University's Institutional Repository offering could be achieved with minimum “ruffling of feathers” amongst recipients.

The VRE includes contact information for site users, identified by role, and our metrics allowed us to focus our mailings on those who were likely to be genuinely interested, and so reducing “collateral” recipients. Our common understanding of privacy issues allowed us to email participants without stepping beyond our assurances, and guided us in appropriate and inappropriate
topics and styles of approach.

Open Access
Throughout the project, the most fraught issues with engaging academic collaborators have been focused in the area of Open Access and Dark Archiving.

With the imperative switching facilities provided by our adaptor software, this is not a major technical problem: an institution can choose, if it wishes, to divert information to a dark collection, on the basis of user input.

However, there was understandable resistance to offering this at the partner institutions. This is not the place to discuss such a large issue, which is deeper and more complex than might be initially presumed, except to reaffirm the observation that the issues raised remain a considerable barrier to improving deposit rates and satisfaction.

As a consequence of the project's Open Access challenges, a deeper discussion is now developing within the various interested parties as to the role of Open Access and appropriate limits to access (if any).

Example Spin Off: Linking Dissemination to Repositories
It is easy to claim that a project has led to numerous, diverse spin-offs beyond the remit of the project as a consequence of the close collaboration of the VRE and Institutional Repository teams as initiated by CTREP. But the project team feel that it is important to back this assertion up with at least one substantive example.

One consequence of the institutional component of the project is that technical resource has been freed to further develop the Institutional Repository. CTREP included the appointment of a repository manager and liaison officer, which freed the Institutional Repository systems administrator and software developer from attempting to also fill those roles. As a consequence, the number of technical improvements to the Institutional Repository has increased substantially.

This is of relevance to the project because it enabled a technical collaboration between the project partners which were not foreseen in the original proposal, but which forwarded its aims.

One of the roles of CARET, the institution responsible for the VRE, is one of dissemination and user-engagement through the web (and similar means), of research content, in a broad range of specialist subject areas. To this end, CARET maintains an online presence for a number of research teams, presenting their research and teaching information (much of the raw material of repositories), in a user-centric manner.

Ultimately, it is hypothesised, at CARET, that the structure of a future VLE/VRE will centre around collaboration arranged around documents and similar information (videos, audio, polls, etc.), pulled from an associative “soup”, which contrasts with the current site- and tool-based model common to existing VLE/VREs.

Many efforts of the Sakai Foundation focus on making this a reality.

It is therefore seen as the role of the VLE/VRE providing institution to provide these creation, dissemination, discovery, aggregation and summary services, despite them being poorly evidenced by existing VRE technology, but with a view to medium-term integration of these services in a common research environment. Such integration is already being piloted as part of the Sakai 3 project.
Before CTREP, this per-project dissemination role was seen as quite distinct from the archiving concerns of each research project.

However, during CTREP, an opportunity presented itself to both the Institutional Repository and CARET teams, to investigate another model, whereby the Institutional Repository served as a datastore which backed the dissemination of a large image-based dataset (the Freeze Frame project).

Whilst some funding had been allocated to dissemination as a part of this digitisation project, the decision was made to go the extra mile, under the broad umbrella of CTREP, and to investigate this Institutional Repository / Disseminator linkup through the use of custom-developed repository Web Services. These services were developed by the technical team which had been freed by CTREP’s appointment of liaison and management staff.

The final architecture extended beyond the remit of the digitisation project, but was seen as a vital step towards the joint offering tactic which is central to CTREP’s institutionalisation agenda, and to serve as a testbed for further integration and institutionalisation.

This integration was a worked, with the site launched on time and being widely perceived as successful, and provides a platform for further technical integration. Since this tentative integration between VRE/IR in the other direction to that initially embarked upon (i.e. IR to VRE) the policy expression issues which were the focus of CTREP again raise their head. Therefore CARET has started to consider how they might be addressed in the light of CTREP, and envisages a solution similar to that developed for ingest.
Therefore, this spin-off integration can be used as a basis for investigating dissemination integration between IR and a future Sakai-3 based content oriented VRE.

**Example Opportunity: WebDAV**

We also feel that it is appropriate not only to *assert* that opportunities have been developed by CTREP which will be exploited in the future, but also to give an example of such.

The VRE offers WebDAV access to its resources. This allows a user to mount a folder on their local computer and place files into it using their regular operating system operations, which are then transferred to the VRE. Through using the VRE/IR adaptor, the process of transferring files is somewhat simplified by this functionality. A whole folder of documents can be transferred to the VRE, along with a metadata control file, and then, through use of the adaptor, ingested into DSpace.

When we realised that this was possible, it became clear that the opportunity presented itself to permit very simple ingest into the Institutional Repository via WebDAV.

However, we must address two wrinkles. The first is the perennial issue of associating metadata with data: an automatic extraction process could perhaps assist there, or else embedded file metadata. The second is to remove the step of needing to log into the VRE to use the ingest widget, but to eliminate it in such a way that procedures are executed according to IR policy without any active user interaction.

We are actively investigating how we might achieve these goals.
Outputs and Results

Institutionalised VRE/Repository Adaptor

The CTREP team developed a Java-based open source adaptor to connect a VRE to a Repository. It is committed into CARET's svn (https://saffron.caret.cam.ac.uk/svn/), and is available for download. A little superficial configuration will be required to deploy it to another organisation, but work is ongoing to reduce this necessity.

The adaptor work was more complex than initially envisaged.

As well as providing a widget which is embedded within the VRE, a powerful administration interface was developed to allow the configuration of this widget, and to perform a wide variety of institution policy-specific tasks (such as metadata crosswalking, routing of submissions to different collections, per user customisability of display and process, and so on) through the use of an event-driven imperative scripting language.

Illustration 5: An early version of the widget being tested in live Camtools
### Illustration 6: Configuring the widget: admin interface

<table>
<thead>
<tr>
<th>Option Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Audio</td>
</tr>
<tr>
<td>Book or Book chapter</td>
<td>Book or Book chapter</td>
</tr>
<tr>
<td>Conference Objects</td>
<td>Conference Objects</td>
</tr>
<tr>
<td>Dataset</td>
<td>Dataset</td>
</tr>
<tr>
<td>Learning Object</td>
<td>Learning Object</td>
</tr>
<tr>
<td>Image</td>
<td>Image</td>
</tr>
<tr>
<td>Article</td>
<td>Article</td>
</tr>
<tr>
<td>Manuscripts</td>
<td>Manuscripts</td>
</tr>
<tr>
<td>Map</td>
<td>Map</td>
</tr>
<tr>
<td>Musical Score</td>
<td>Musical Score</td>
</tr>
<tr>
<td>Presentation</td>
<td>Presentation</td>
</tr>
<tr>
<td>Software</td>
<td>Software</td>
</tr>
<tr>
<td>Report</td>
<td>Report</td>
</tr>
<tr>
<td>Thesis</td>
<td>Thesis</td>
</tr>
<tr>
<td>Video</td>
<td>Video</td>
</tr>
<tr>
<td>Webpages</td>
<td>Webpages</td>
</tr>
<tr>
<td>Other</td>
<td>Other</td>
</tr>
<tr>
<td>Managed Research Project</td>
<td>value: 3</td>
</tr>
</tbody>
</table>

Add more values: how many? [6] (to delete, blank out value)
This connector is deployed within the institutional VRE, and has been used to ingest a variety of papers, posters, transcripts, and so on, into the Institutional Repository. We are currently following up on leads to allow us to increase its use in other research groups.

The source code has been developed with a view to being usable in a large variety of places, and for a long-term future. To this end, we identified a number of places where uncertainty exists in the future of standards, protocols, and endpoints, and separated them behind APIs to allow their replacement or customization. This added to the cost of development, but was seen as invaluable in terms of long-term sustainability.

Further, there are places where a number of approaches could be used in a variety of ways by an institution to accomplish a given task (e.g. metadata to data binding). In these cases we developed software library functionality which wraps and encompasses this functionality, whilst presenting a simple interface to the script author. This functionality allows a configuring institution a great deal of lea-way as to if and how they can use these facilities through the scripting language, which will be a paradigm familiar to those who are comfortable writing “macros”.

However, the software additionally functions in a default mode, based on simplifying assumptions, should no such scripts be supplied, and therefore such experience is not mandatory. In addition, places where we consider reconfiguration will be sufficiently common practise, more conventional configuration (through forms, and such like) is provided. This raw configuration is then, optionally, passed through a script component for ultimate configurability.
Illustration 8: Structure of Repository/VRE Adaptor

Illustration 9: Data Flow for Schema Presentation and Ingest
Lessons Learnt Documentation

Throughout the project, the team bore in mind that the lessons learnt from institutionalising our product would be at least as valuable to other institutions as its technical outputs. To this end a document is in pre-print, "Culture Clashes and Bikeshedding: designing interoperability for connecting Repositories and Virtual Research environments at the Cambridge TETRA Repository Enhancement Project", having been prepared by a professional writer, retained for this purpose.

This document includes two detailed case studies of end-community engagement undertaken as part of CTREP, a discussion of the issues surrounding professional cultures, and a brief overview of the project. This document is expressly aimed at a general audience of individuals and institutions which may be considering embarking on similar projects in the future.

Spinoffs and Ongoing Processes

A major, but hard to quantify, outcome of the project has been a closer working relationship between the VRE and Institutional Repository teams. Initially necessary to ensure that the adaptor was a success, this closer relationship has this has led to a number of projects which adopted a more joined up approach to institutional offerings, particularly on projects concerning digitisation, image management, and so on; also the adoption of joined up approaches to policy, marketing, and so on.

One major development is the continuation of the spreadsheet-based ingest process developed during the project, to develop institution-wide pro forma spreadsheets to represent bulk ingest operations.

Institutional Commitment

CTREP has been positive experience for the Institutional Repository, and for CARET (as managers of the VRE). We are therefore committed to continuing the relationship beyond the end of the project. Indeed, it has been a struggle to draw a line under the project as developments continue which impact on CTREP. We are finding many opportunities to build on CTREP outputs as a part of other projects and programmes being undertaken within the University, so continuation of work is unlikely to become a resource issue.

Other Outcomes

In addition to the outputs described in the previous section of document, the team have developed skill concerning bridging institutional cultural differences with the end of increasing repository uptake. We now have a widget in use for the ingest of real research data into the institutional repository (by research groups not a part of the CTREP team). Some hand-holding is often required, but as we identify these issues they are being resolved. We are continuing to find opportunities to engage with communities to discuss the possibility of more of them using the VRE adaptor in a wider variety of circumstances.

We have developed expertise in the areas of policy expression which we are disseminating through a number of means (talks, source code, this document, other reports, etc.). This is already proving invaluable in a large number of independent projects in far flung fields: museum collections management, for example, where our experience of the embodiment of policy expression is being
developed further, to allow multiple, widely-varying museums to accept a single Collections Management product which they can, themselves, customize.

The project has challenged many of the preconceptions of members of the CTREP team, and even affected how one research team considers its data. A number of assumptions were challenged during the project (detailed in earlier sections: metadata extraction, declarative policy expression, open access, etc.) and have led to a deeper understanding of the issues concerned.

Whilst considerable space in this report has been given over to considering the technical work of the project, it's essential not not neglect to mention that over the period of the project, (largely as the product of the work of the CTREP repository manager and liaison officer in undertaking general development and marketing activity), the number of Institutional Repository communities has doubled and the number of collections tripled. There are now 200,000 items in Cambridge's Institutional Repository, and 500,000 bitstreams, occupying 7Tb.

Conclusions
The CTREP project successfully developed and deployed an integration between the Institutional VRE and Institutional Repository. Care was taken over the longevity of the source code in terms of changing requirements at each institution and in its ability to be redeployed at multiple institutions with different policies and cultures. We have examined this ingest process being used in practice and have refined our process as a result. As with any institutionally-deployed software, this work is ongoing, and all partners have shown a clear commitment to continuing its development.

At the same time relationships were built between technical cultures within the university and also between the Institutional Repository and various academic stakeholders. The issues raised broadened and deepened our understanding of issues around increasing repository uptake and those of academic staff, and we are endeavouring to disseminate that knowledge.

By grappling with the major issues around policy expression, innovations have had a profound impact on other projects at the university, in diverse fields from teaching administration to museum collections management, and we have discovered which aspects of the vast literature available on policy expression is and is not applicable in these circumstances.

Camtools, the University's VRE now has a widget which allows deposit to the University's Institutional Repository. Whilst already fully functional in the live environment, the widget is also incorporated into the plan for the next academic year's VLE/VRE upgrade, after which it will be fully in control of the VLE/VRE team (a change which will be transparent to the user).

As the widget is open source, with a little customization it can be integrated into similar environments. We are working to reduce the degree of customization required.

Implications

Direct Impact
As a consequence of CTREP, we have an integration between our institutional VRE and Institutional repository which allows VRE material to be easily deposited in the institutional repository. The institution is committed to further developing and enhancing this integration.

As a consequence of the work of the repository manager and liaison officer, institutional uptake has broadened and deepened.
**Cultural Impact**

**Technical Tools to Address Cultural Issues**

The well-known adage that there are “no technical solutions for social problems” should serve as warning when it is proposed that architectural patterns can ease cultural concerns. However, what we have endeavoured to achieve throughout CTREP has, rather, been development of valuable technical tools which, in certain circumstances, may be applied in inter-cultural and inter-institutional contexts. Whether or not such tools are appropriate and how they might be used remains a cultural matter.

The tools which we've found have the greatest impact, described in detail elsewhere, are

- use of firebreaks, i.e. technical interventions focussed on moving discussion forward by embodying the discussion's conceptual ground, e.g. allowing arbitrary crosswalking to forestall long crosswalking discussions. (see Methodology section of this report)
- use of shadowing, to allow a technology to prove itself before it requires detailed attention from administrators (see Implementation section of this report)
- use of server/client/server architecture allows software to be deployed in a production environment whilst remaining experimental, yet not destabilising that environment.
- use of imperative rather than declarative policy expression (“what do we do”, not “what is it”) to avoid stirring up a hornet’s nest of definition and purpose

**Understanding of Cultures**

We would like to collate, in concrete terms, another key impact of CTREP alluded to throughout this document. The CTREP project has assisted the stakeholders in understanding the various interests of the other stakeholders, and has helped to provide means to bridge divides.

Specifically, this has led to

- a deeper understanding of key issues in institutionalising repositories (e.g. issues around access, research ethics, repository scope, etc.), which we now seek to disseminate throughout the institution and more broadly.
- The design of creative approaches to VRE/IR integration for a further round of integration development (e.g. WebDAV based ingest, pro-forma spreadsheets, etc.).
- an increase in awareness of issues specific to each stakeholder's interests, to permit cooperation and joined up offerings to research teams (e.g. in areas around marketing, metrics, etc.).

**Policy Expression Implications**

Perhaps the most profound short-term impact of the project, however, was unexpected. It concerns the means of policy expression and configuration.

The server-side javascript scripting, or “macro”, approach, modified by traditional configurability for the simpler cases, was arrived at after a careful examination of the alternatives, and with a key focus on available skills and ubiquity rather than on technical merits.
We are therefore discovering its applicability in a large number of other places where similar policy and cultural issues arise.

These diverse projects have all benefited from CTREP deciding to take on the ambitious issues of deployable, and institutionalised policy-expression within the Institutional Repository domain.

**Glossary and References**

- **AJAX**: Asynchronous Javascript and XML, http://www.w3schools.com/Ajax/ajax_intro.asp
- **API**: Abstract Programming Interface, http://www.computerworld.com/action/article.docommand=viewArticleBasic&articleId=43487
- **CSV**: a simple means of representing a spreadsheet, http://www.nightflight.com/foldoc-bin/foldoc.cgi?comma+separated+values
- **Dark Archive**: repository material which is not publicly visible, or is restricted to a very few users.
- **DSpace**: repository software, http://www.dspace.org/
- **DSpace 2**: repository software in development, http://wiki.dspace.org/index.php/DSpace_2.0/Kernel
- **Dublin Core**: the ubiquitous bibliographic metadata standard, http://dublincore.org/
- **Fedora**: repository software, http://www.fedora-commons.org/
- **FOXML**: a simple ingest format of Fedora, http://www.fedora.info/download/2.0/userdocs/digitalobjects/introFOXML.html
- **Institutional Repository (IR)**: repository of digital material charged, by an institution, with disseminating and preserving that material, in a manner akin to the role of a library as to books
- **iRODS**: a rules-driven programming language, used on data grids, https://www.irods.org/index.php
- **JSON**: a simple, structured data-representation language, http://json.org/
- **Learning Landscapes Project**: CTREP research collaborators, http://www.caret.cam.ac.uk/blogs/llp/
- **Open Access**: the ability to freely access material within a repository with few restrictions
- **OWL**: Web Ontology Language [sic], a means of representing reasoning with declarative
statements, such as those represented in RDF, http://www.w3.org/TR/owl-features/

- **PLEDGE**: a collaboration between MIT, UCSD and SDSC to represent embodiments of repository policy, http://pledge.mit.edu/index.php/Main_Page

- **QUADS**: Qualitative Archiving and Data Sharing Scheme, http://quads.esds.ac.uk/projects/ee.asp

- **RDF**: Resource Description Framework, a means of representing assertions in a machine readable manner, http://www.w3.org/RDF/

- **RDFS**: Resource Description Framework Schema, a means of representing permitted statements within an RDF application, http://www.w3.org/TR/1999/PR-rdf-schema-19990303/

- **REI**: a policy specification language, http://www.cs.umbc.edu/~lkagal1/rei/


- **Sakai**: VLE/VRE, http://sakaiproject.org/portal

- **SDSC**: San Diego Supercomputer Center, http://www.sdsc.edu/


- **SWOR D**: Ingest protocol for repositories, based around ATOM, http://www.ukoln.ac.uk/repositories/digirep/index/SWORD_Project

- **Teaching for Learning Network**: CTREP research collaborator, http://www.tfln.org/


- **VRE**: virtual research environment, http://www.ariadne.ac.uk/issue44/fraser/

- **WebDAV**: a web-based protocol for the mounting of remote drives on a local computer, http://www.ics.uci.edu/~ejw/authoring/