UDDI summary report

E. Tonkin & T. Koch

March 2007

Note:As Traugott moved on to the Max Planck Digital Library in January 2007, this document was completed by Emma Tonkin. Any inaccuracies in this document belong to her.

Contents

1 Introduction

UDDI (Universal Discovery, Description and Integration) is an OASIS (Organization for the Advancement of Structured Information Standards) standard, or to be more accurate, several generations of standard. The UDDI initiative was designed to provide specifications for several technologies intended for the purpose of building registries of web services. Initially, the intent was to permit a centralised registry service (an "Internet white/yellow pages" as it were) though the emphasis later moved onto the provision of distributed registry services. The standard covers both registry design and query methods, though it leaves open the possibility of using a number of different methods of service description.

Phase 2 of the IESR included a research work package, led by UKOLN, in which the possibility of making IESR metadata available using a UDDI registry was investigated, along with alternative models for the delivery of the registry. Current developments in related arenas, such as the Grid, were also monitored as part of this work package.

Matthew Dovey carried out a UKOLN study on the topic of mappings between UDDI and IESR metadata (for the IE Architecture).

In work carried out by Pete Dowdell, the initial expectations for the provision of a UDDI interface to the IESR were collected, and available software and libraries were enumerated. Following his departure, this work was continued by Emma Tonkin.

2 Meetings and reports

Apart from internal meetings at UKOLN and at IESR level, two broader meetings organised by UKOLN specifically addressed the UDDI issue:

1. the Distributed Service Registries Workshop, Warwick, 14-15 July 2005 [Distributed] with international participation, especially in a presentation by Dovey [Dovey 2005b] and in discussions.

2. a UDDI-IESR meeting at UKOLN, University of Bath, 14 Oct 2005 with participation from Matthew Dovey from the Oxford e-Science Centre and Weijian Fang, from the Grimoires project and the University of Southampton, as well as the participation of IESR and UKOLN

colleagues.

The aim was to intensify the information sharing and discussion about service registry solutions between the JISC IE and eScience communities with a primary focus on the use of UDDI. The meeting was very informal and intensely discussion oriented. Weijian Fang from Southampton presented the Grimoires project and registry software. UKOLN colleagues presented overviews and some of the findings and open questions of the investigation regarding the suitability of UDDI for the IESR. Matthew Dovey offered information from his working context, comments, discussion of state-of-the-art solutions, the newest UDDI version and alternatives. The meeting contributed to a more shared understanding and an insight in the limited role UDDI plays in the Grimoires registry but did not reach a final decision regarding the suitability of UDDI to IESR. Participants agreed upon the necessity for undertaking some further investigation of different mapping options between IESR and UDDI, identified during the meeting.

2.1 Important Reports/steps in the IESR UDDI discussion

Matthew Dovey's 2002 report on behalf of UKOLN on the information environment infrastructure mentioned the possibility of using WSDL and UDDI as the basis for service description within this environment :

> A recent study by Matthew Dovey (Dovey 2002), considers the possibilities for using WSDL and UDDI as the basis for service description within the JISC IE, albeit acknowledging the need to provide access to richer collection descriptions within or alongside these technologies.
>
> Quoted from (Powell et al, 2002)

This was followed up by further work by Dovey, including the following two reports- a proposed mapping between UDDI and IESR :

UDDI-IESR Mapping (2003)

and :

Dovey, Matthew J. 2005b

Emma Tonkin then examined various approaches to defining a UDDI interface to the IESR service in the following report :

UDDI and IESR, 2005

Following the drafting of a report describing this work, a meeting was held at UKOLN in October 2005 in which the mapping was discussed, along with a presentation discussing the Grimoires project (see previous section for details on this topic). This resulted in further work in early 2006 on defining methods of evaluating the suitability of each of several possible approaches to mapping.

2.2 Further interest in Service Registries & UDDI

The eFramework working group took a great deal of interest in the topics of Service Registries and UDDI throughout 2005, aiming to catalyse wider participation in the discussion process. A review of various JISC service registry initiatives was also carried out during this time. Discussions took place about future avenues for data sharing, discussion and cooperation ; the question of the role of UDDI in the eFramework environment arose at that time.

Several later meetings related to the topic of the distributed service registry approach have seen little discussion of UDDI, which might indicate that it is not seen as a critical technology at this time.

Several other events with strong involvement of IESR partners did focus on Service Registries, but did not discuss UDDI in any detail, e.g.:

1. the Digital Library and its Services, Breakout session "Services and Collection Registries", led by two colleagues from UKOLN, British Library, London, 7 March 2006

2. the Workshop on Digital Libraries Service Registries, organised by NSF/NSDL with participation of US colleagues and IESR, Washington D.C., 22-23 March 2006.

The limited interest shown in a UDDI interface may indicate that UDDI is not in common use, that it is in common use only in limited contexts (such as distributed or GRID computing contexts), or that IESR-related user groups do not see a close relation between the role of the IESR and the contexts in which UDDI is typically used - ie. the industrial/W3C standards underlying a service-oriented architecture/web services environment may not be perceived as relevant to work in this arena.

3 Issues

3.1 UDDI view of IESR data

Mapping between UDDI and IESR is complicated principally by the fact that UDDI is designed to appeal to businesses. It has a simple model of services, service providers and businesses, which is designed to allow more complicated relationships between businesses to be indicated by defining relationships between entities.

It is possible in UDDI to define a hierarchy of entities (businesses) within UDDI, each of which may own services. This allows fine distinctions to be mapped, such as 'this business is the owner of a business which administrates a business that itself controls a service'. However, this is not a model that relates directly to relationships as mapped out in IESR. Sophisticated provider relationships do not map directly between IESR and UDDI.

This leads to the following conclusions:

- IESR and UDDI are intended to capture information appropriate for slightly different processes and actors. For this reason, UDDI cannot accommodate fine distinctions of IESR and vice versa.

- Any mapping between IESR and UDDI is likely to be lossy. The representation in UDDI will not contain all of the information initially contained in the IESR record.

- A mapping between IESR and UDDI is not reversible. Any importing of records via a UDDI interface will lead to an incomplete IESR record, which will contain empty elements that must either be filled by hand or ignored entirely.

With the above in mind, it is perhaps simplest to consider UDDI gateways as a convenient view (faceted view) of underlying IESR data, rather than as a mapped equivalent to IESR data.

Potential mappings are discussed in the Appendix.

3.2 Shift in context for UDDI

UDDI is now in use as an enterprise registry or repository; that is, one UDDI server (or server cloud) exists per business context.

When initially (commercially) launched, UDDI was designed as an integral element in service discovery for service-oriented architectures. The design was intended to permit machine-to-machine (m2m) discovery and use of services ; it therefore provided for a great deal of detail in the description of service interface and binding to enable the latter to occur.

This aim is not appropriate or necessary in every context in which a UDDI registry may be used. In this case – a case in which service descriptions exist primarily for human eyes, such as a cross-domain registry designed principally to provide starting points from which system developers can work – the m2m work provides little additional benefit, yet severely increases the complexity of the system.

In such contexts, UDDI is used for discovery of services according to a different/wider set of metrics. These may be very contextual, not to say political. Rather than placing the strictest emphasis on machine-readable service description, there is a need for detailed, accessible human-readable metadata. Where the machine-readable interface is used, UDDI is often serving only as a router (ie. redirecting to one of several interfaces for a known service class – for example, 'An access point to this data set is required. My usual access point is down. What alternatives may I use that are accessible to me?').

3.3 UDDI and SOAP

UDDI is operationally (though not conceptually) closely linked to SOAP, the Simple Object Access Protocol that permits remote procedure calls to be issued to a structured network service based on XML and HTTP. It is also closely linked to the underlying suite of related web standards (WSDL, etc).

3.3.1 Heterogeneous service description

It may in design and practice be used to describe services using protocols other than SOAP. Examples of tModels that may be used as bindingTemplates for a variety of transport types can be found in OASIS' UDDI documentation; for examples, see (Boubez & Clément, 2002).

3.4 Use in a distributed registry

The fact that a specific mapping is required for this work to be undertaken - a 'profile' as it were - has implications for distributed registry development. These include the following :

- The compatibility of the records produced by the mapping to records produced by other organisations will have a clear impact on the usefulness of these records in a general UDDI 'cloud' or 'swarm'. The use of an incompatible model will potentially imply that, although the records follow the UDDI specification as such, they will not appear when standard search-and-retrieval techniques are applied - thus leading to a 'soft incompatibility'.

- Cooperators may have to make use of the IESR proposed profile for the UDDI model, or at least ensure that their model follows the IESR proposal to the extent that search-and-retrieval functions appropriately.

There is also a need to evaluate current uses of the UDDI specification with respect to expected use cases of the IESR, with particular emphasis on the level of similarity of the data held within each system. The assumption that it is appropriate to share data between the two systems depends heavily on the assertion that users of a UDDI directory will benefit from discovery of IESR metadata – the accuracy of this assertion depends entirely on current use contexts for UDDI.

3.5 Evaluation

In the wider standards arena, the digital library environment and the W3C specification process have seldom seen eye-to-eye. That is to say, the implications of working with SOAP/UDDI based standards include a number of risks for organisations in the digital library environment. One is handling user expectations or observations, such as the belief that UDDI and SOAP-based solutions are overly complex, or the assertion that use cases for UDDI and those for IESR may not mesh closely – that is, that the benefits provided by UDDI may be of little relevance to the digital library environment.

The IESR project bridges this gap by providing both interfaces, which constitutes a realistic and pragmatic approach to the support of multiple communities of practice, of which many may potentially operate on a small scale.

3.6 Alternatives

3.6.1 Service Description

The UDDI standard is generally used together with WSDL, the Web Services Description Language. There are several alternative technologies that perform a similar task to a greater or lesser degree of precision.

- WSDL 1.1 focuses on versatility, providing a means for description of non-web service standards.
- A simplified version, SOAP Service Description Language (SSDL), is principally designed to describe SOAP-based services (Parastatidis et al, 2006).
- SMEX-D, a Simple Message EXchange-Descriptor, was proposed by Tim Bray
- Various RDF/OWL approaches.
- Web Resource Description Language, WRDL.[1]
- WSML, Web Services Modelling Language (from the European Semantic Systems Initiative)
- Web Application Description Language, WADL
- Resedel (restful services description language)
- RSWS (Really Simple Web Service Descriptions)

[1] See `http://hinchcliffe.org/archive/2005/05/31/280.aspx`

3.6.2 Registry alternatives

Alternatives to the UDDI registry itself include the Grimoires system (itself built on the UDDI specification). ebXML is also a possibility. ebXML, electronic business XML, is an parallel initiative to the W3C Web Standards work with the aim of researching, developing and promoting efficient and interoperable standards for exchanging XML-based business data electronically. The initiative is jointly sponsored by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS).

Because the W3C and ebXML work eventually converged, ebXML makes use of SOAP as its default message transport. The OASIS ebXML standard defines a data model for e-business objects, a message transport, and a registry, also designed for e-business objects. The ebXML registry may be compared to UDDI. The level of granularity is somewhat different; UDDI was designed to support description of web services in general terms, whilst ebXML is expected to be used in a business context. The ebXML data model is UML-centric, whilst UDDI is associated to XML schema. Further work from the OASIS has focused on the two approaches, examining the relationship between the data structures (see for example Kibakura, 2004), and defining a superstructure the JAXR - Java Access to XML Registries, which provides an abstract interface across both.

3.7 Retrievability of UDDI records

One criticism that could, perhaps, be levelled at UDDI relates to the type of metadata stored within the system : the extent to which, in the words of Infoworld's Jon Udell[2], UDDI "fits into the process of description and discovery". Udell suggests that service information should be captured within the wider context of relevant institutional knowledge.

Udell here captures a very significant point. The fate of UDDI as a cross-domain service registry strongly implies either that the stated aim – the requirement for which the UDDI registry was originally engineered – was in some sense misdirected, or alternatively that the implementation failed to support the process as expected. There are many possible reasons for this and, though a few guesses are made here, exhaustive analysis of this issue is beyond the scope of this document. However, Udell's suggestion is a good one.

[2] http://www.techworld.com/networking/features/index.cfm?featureid=1890

A few questions that might be worth asking : to what extent do services "make sense" outside their original context? To what extent are service descriptions designed to be readable outside a business context? Are links enabled to corresponding information sources or community venues – is there any attempt or mechanism to situate completed development work within its original context or related areas?

This may represent a process issue, and therefore a matter of providing and setting appropriate policies and documentation. It may also be an opportunity for greater integration of the Registry into a wider, heterogeneous information environment, in order to support a heterogeneous developer population with many differing approaches to application/service design and development. It is difficult to directly support such a variety of approaches, but entirely possible to provide a number of "hooks" required for third-party services and applications (for example, a stable URI is one such "hook"; the availability of verbose content via HTTP GET is another, since it permits indexing via Google).

3.8 Architecture

Initial expectations included that service descriptions would be shared dynamically between the IESR and a UDDI registry via an appropriate protocol, ie. UDDI, OAI or SRW. This may be carried out in a real-time manner, or in batch mode; alternatively, service descriptions could be cached locally.

Whilst harvesting from the IESR could be carried out as expected, directly (dynamically) harvesting from the IESR in a sufficiently responsive manner proved to be problematic. The issues were not related to IESR - instead, it was simply a question of limited availability of libraries appropriate for both the jUDDI and IESR environments. Few of the available UDDI libraries were (at the time of investigation) able to offer a full interface for publishing data to a UDDI database in an automated manner - most were designed only to allow records to be retrieved from UDDI servers. As a result, a pragmatic alternative was sought.

jUDDI was set up to make use of an SQL backend database; therefore, harvesting from the IESR directly into jUDDI's database represented one feasible alternative to this limitation.

At the time of writing, however, this may well have changed. The Perl library UDDI::Lite, available on CPAN, has been updated throughout 2006 with several new releases and functionality upgrades. Publication directly via the UDDI interface is therefore now likely to represent a feasable/preferable alternative. This should be revisited/reevaluated as one of the first steps in a proof-of-concept development. However, bear in mind that the publication process must support the chosen mapping approach in its entirety, as well as the basic process of publication of business entities, services and binding templates. The choice of mapping will therefore have an impact on the requirements (and complexity) of the development process.

3.9 Mapping

The meeting 14 October 2005 addressed the suitability of UDDI for presenting a view of the JISC IE Services Registry. Both Matthew Dovey (2003) and Emma Tonkin (2005) have written papers to consider the central question: How should the collections, services and agent information in the JISC IE SR be mapped into the UDDI model? Pete Johnston has summarised the issue as:

> ”The core entity types in the UDDI data model are the ”business”, the ”business service” and the ”binding template”. The UDDI model supports the description of typed relationships between businesses, and the categoryBag feature provides a mechanism for representing searchable attributes of these entities. There are several options for mapping the IESR model (with its entity types of ”collections”, ”services” and ”agents”) to the UDDI model. At present, it is not clear which option is the most appropriate.”

The meeting agreed the following way forward:

In order to establish the most appropriate mapping of IESR data to the UDDI model, we require a more detailed evaluation of the various mappings suggested by Matthew and Emma. This evaluation will require the following steps:

- Consider how an IESR record (such as a description of Zetoc as a collection and a set of services) would be treated using each mapping

- Consider how each mapping fulfils the use cases for the IESR (this might involve more work on fleshing out use cases

- Consider how each mapping relates to how others do UDDI (eScience in UK and beyond, commercial world)

Part of the approach to develop the mappings was to write up some of the ”key” queries IESR could support now, and for which one should aim to provide the functional equivalent via UDDI. This was done by Pete Johnston in Spring 2006 [Johnston 2006]. The queries would form part of the ”criteria” - not the only criterion but part of them - for evaluating the different mappings: if the result of a mapping supported most of the queries then this constituted a point in its favour; and vice versa, if the result of a mapping supported very few of the queries then that was a point against it.

4 Conclusions and decisions

Proposals of using UDDI to access the IESR registry content date primarily from about 2001-2003, because of international developments with commercial involvement.

It seems that UDDI can be used to access IESR metadata. The technical and software requirements can be fulfilled (Tonkin 2005). However, there are at least four different ways of mapping IESR elements into UDDI ones. The question therefore becomes which of these methods suffices for the role, and of these, which represents the more promising solution. Methods of evaluating each mapping have been suggested in this document :

- Testing with specific IESR records, ie Zetoc
- Providing a compatibility score relating to use cases
- Evaluating common mappings

We note again that the relevance of UDDI (global public) registries seems to have changed, but that there remains evidence that UDDI is in use in various contexts (see for example Grimoires).

Recommendation:

UDDI will not be the main access route to IESR, but might very well be provided as one option during 2007, if the mapping approach is found suitable and easy to implement.

Recommendation:

Investigation of the ways in which the UDDI specification is applied by those stakeholders with an interest in the Service Registry, and the completion of an evaluation process similar to that described here, will help to inform the development of a proof-of-concept.

Recommendation:

To consider the UDDI interface within context (as part of a wider system), and to consider what it offers in the Web 2.0 context.

Traugott Koch, Jan 2007
Emma Tonkin, March 2007

5 References

Allan, Rob et al (2002). UDDI and WSIL for e-Science. http://tyne.dl.ac.uk/Papers/UDDI/uddi/

Boubez, Toufic and Clément, Luc (2002). UDDI Other Core tModels, Version 2.04. http://uddi.org/taxonomies/UDDI_CoreOther_tModels.htm

Distributed Service Registries Workshop, Warwick, 14-15 July 2005 http://www.ukoln.ac.uk/events/dsr-workshop-2005/

Dovey, Matthew, J. (2001). UDDI (Universal Description, Discovery and Integration) Standard and Initiative. JISC Technology and Standards Watch Report TSW 01-01: UDDI, June 2001. http://www.jisc.ac.uk/index.cfm?name=techwatch_report_0101

Dovey, Matthew, J. (2002?) JISC Information Environment Service Level Descriptions. Study for UKOLN. http://www.ukoln.ac.uk/distributed-systems/dner/arch/service-description/study/ http://www.ukoln.ac.uk/distributed-systems/jisc-ie/arch/service-description/study/

Dovey, Matthew, J. (2003-09-18). UDDI and the Information Environment Service Registry. (Initial draft Altern. title: UDDI-IESR Mapping).

Dovey, Matthew, J. et al (2005). Evaluation of UDDI for UK e-Science. UK Engineering Task Force. UK Technical Report UKeS-2005-04. http://www.nesc.ac.uk/technical_papers/UKeS-2005-04.pdf

Dovey, Matthew J. (2005b): Evaluation of UDDI for the UK e-Science GRID. Presentation at the Distributed Services Registry Workshop, Warwick, July 14-15, 2005 http://www.ukoln.ac.uk/events/dsr-workshop-2005/presentations/m-dovey.ppt

Fang, Weijian (2005). The GRIMOIRES Service Registry. Presentation at UKOLN, 14 Oct 2005

Grimoires' homepage http://www.grimoires.org

Grimoires' current release and documentation http://twiki.grimoires.org/bin/view/Grimoires/SoftwareReleases.

Grimoires related papers http://twiki.grimoires.org/bin/view/Grimoires/GrimoiresRelatedPublications.

Johnston, Pete (2006). IESR-UDDI. Some Usage Scenarios for IESR Metadata.
http://homes.ukoln.ac.uk/~lispj/iesr/iesr-uddi/scenarios/

Kibakura, Keisuke (2004). UDDI as the registry for ebXML Components.
http://www.oasis-open.org/committees/uddi-spec/doc/tn/uddi-spec-tc-tn-uddi-ebxml.htm

Koch, T. (2006). Issues raised at Warwick Distributed Services Registries workshop
http://homes.ukoln.ac.uk/~tk213/DSR-issues.html

Parastatidis, S. Woodman, S. Webber, J. Kuo, D. Greenfield, P. (2006). Asynchronous messaging between Web services using SSDL. IEEE Internet Computing, Volume: 10, Issue: 1, page(s): 26- 39, ISSN: 1089-7801

Powell, A and Lyon, L. (2002). The JISC Information Environment and Web services.
http://www.ariadne.ac.uk/issue31/information-environments/

Tonkin, Emma (2005). UDDI and IESR. August 2005.
http://iesr.ac.uk/docs/outputs/uddiandiesr/

UDDI. http://www.uddi.org/

6 Appendix

6.1 A: Challenges in UDDI mapping

6.1.1 Introduction

Mapping between UDDI and IESR is complicated principally by the fact that UDDI is designed to appeal to businesses. It has a simple model of services, service providers and businesses, which is designed to allow more complicated relationships between businesses to be indicated by defining relationships between entities.

It is possible in UDDI to define a hierarchy of entities (businesses) within UDDI, each of which may own services. This allows fine distinctions to be mapped, such as 'this business is the owner of a business which administrates a business that itself controls a service'. However, this is not a model that relates directly to relationships as mapped out in IESR. Sophisticated provider relationships do not map directly between IESR and UDDI.

This leads to the following conclusions:

- Any mapping between IESR and UDDI is likely to be lossy. The representation in UDDI will not contain all of the information initially contained in the IESR record.
- A mapping between IESR and UDDI is not reversible. Any importing of records via a UDDI interface will lead to an incomplete IESR record, which will contain empty elements that must either be filled by hand or ignored entirely.

With the above in mind, it is perhaps simplest to consider UDDI gateways rather as a convenient view (faceted view) of underlying IESR data, rather than as a mapped equivalent to IESR data.

The mapping that initially caused this discussion to take place (not, however, the only point of contention between the two models) was the following. Consider Fig. 1, on the following page.

As mapped out here, the IESR agent roughly translates to a business entity. The IESR service roughly translates to a business service, plus binding template. However, the collections information does not have a direct analogue on this diagram. (See later in this document, fig. 4-6, for graphical mappings).

Discussion with Matthew Dovey (who originally authored the mapping) elicited the following classes of possibility;

- mapping iesr:Collection (or maybe ”access to collection”) to business service, and iesr:Service to binding template (this was Matthew’s original option).
- mapping ”type/category” of iesr:Service per iesr:Collection to business service (so you’d have different business services for ”search collection X”, ”harvest collection X” etc), and iesr:Service to binding template

The first option would result in one business service per iesr:Collection, and with the second, multiple business services would exist per iesr:Collection (and the same collection attributes would be attached to each of those binding services). Neither class is ideal; mapping the service to the binding template has the disadvantage that it runs counter to the original use cases of UDDI, meaning that there is a possibility that the result would be sufficiently different to the expectations of UDDI tool authors that, for example, search functionality would be broken. If service definition were held within the binding template, one of the results would be that categorisation has to happen via tModels, and convenient plain-text descriptions are not available (default UDDI searches are generally plain-text in the first instance).

A third option was mapping iesr:Collection and so forth to businessEntity relationships. This has the advantage that the service description layer is not disturbed, and in some ways seems to fit most closely with the original intent of the UDDI designers. However, the relationships that may be described using this model are not terribly complex, and the likelihood is that important data would not be represented if this method were used.

There is also the question of pragmatism; the IESR structure is important, yes, but if the UDDI model represents only a thin compatibility layer/gateway layered on top of the IESR service model, the fact that relationships are not well mapped is not, in and of itself, of primary importance unless they are pivotal to the process of searching the IESR.

A further outcome that came from these discussions was the suggestion that choosing the best of these options is a decision that needs to be informed by looking at the way in which the UDDI model is in practice used. The next step to be taken, then, is to investigate the existing tools and search strategies used within UDDI, and determine the most effective mappings between IESR and UDDI in terms of effectiveness and functionality.

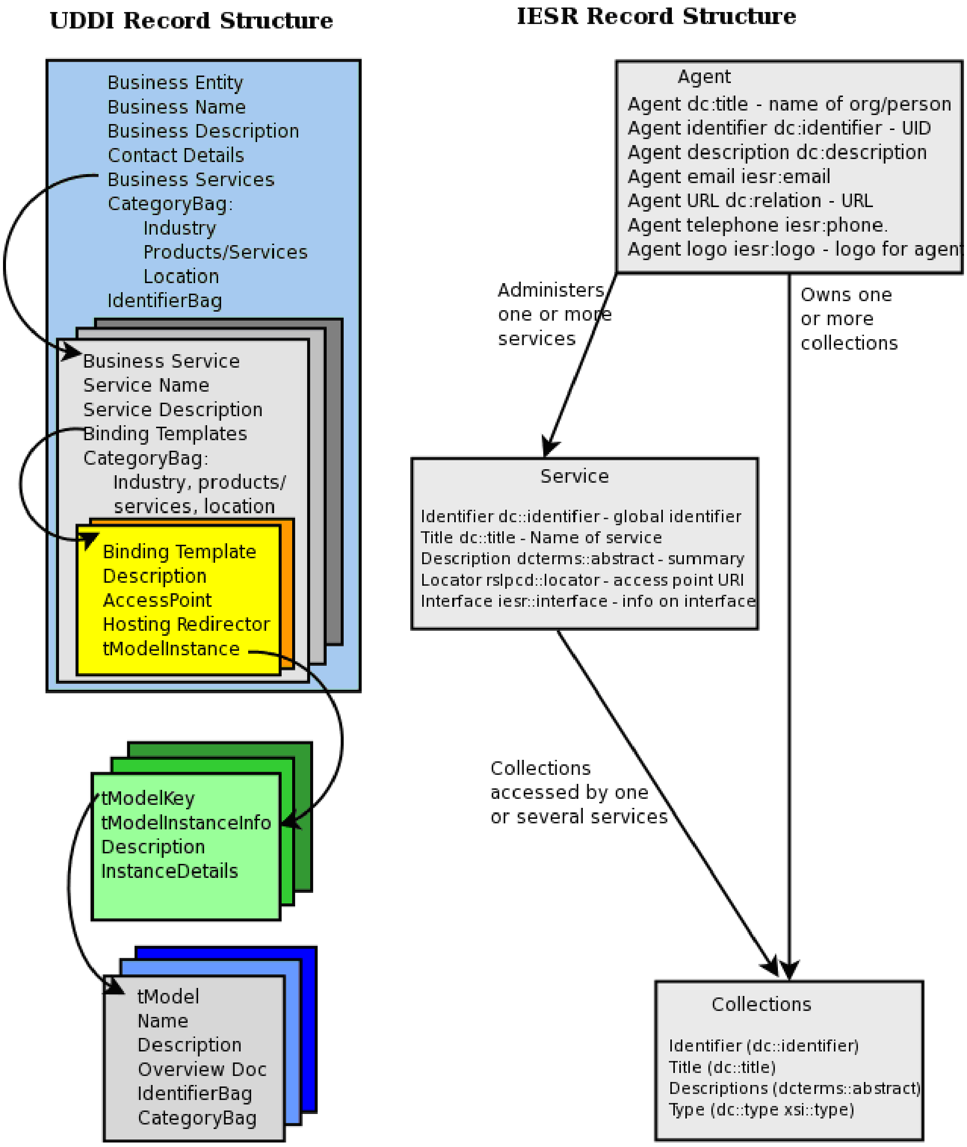

Figure 1: Comparison of IESR and UDDI record structures

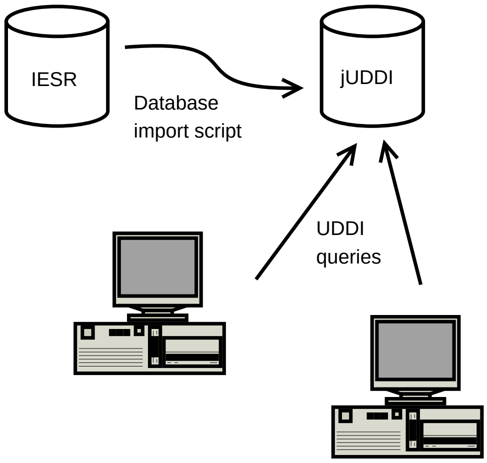

Figure 2: Current architecture

6.1.2 UDDI/IESR architectures

The current architecture (see Fig.2) depends on a regular import of data from IESR into jUDDI. That is, jUDDI is used as the server, and data is imported directly from one database (or from one XML feed) into the jUDDI backend database tables.

This architecture is not particularly convenient or particularly simple; it was chosen largely because jUDDI was an available, stable, and at the time fairly well-supported and mature application. However, no new version has been made available since June 2005, over a year ago, implying that either it has reached a level of maturity in which neither new patches nor new features have been required, or the project has stagnated somewhat since that date. An alternative architecture would require a light proxy layer in between IESR and the UDDI clients doing the job of a UDDI server (or, more realistically, a small subset of that functionality). Unfortunately, as was the case when jUDDI was originally chosen for investigation, there are still no light UDDI server implementations available. The UDDI specification (at `http://uddi.org/pubs/uddi_v3.htm`) is not unreasonably large, and allows for the possibility that only a subset of the specification is implemented by a UDDI node. The API set of principal interest for our purposes is UDDI Inquiry. This would lead to an architecture that looked a lot more like that shown in Fig. 3.

We have found no technical reason why this should not be possible - it may be that the models are dissimilar enough that queries cannot be mapped directly to one another, and that therefore such a proxy would have to be stateful (ie. it would have to collect sufficient information in order to put

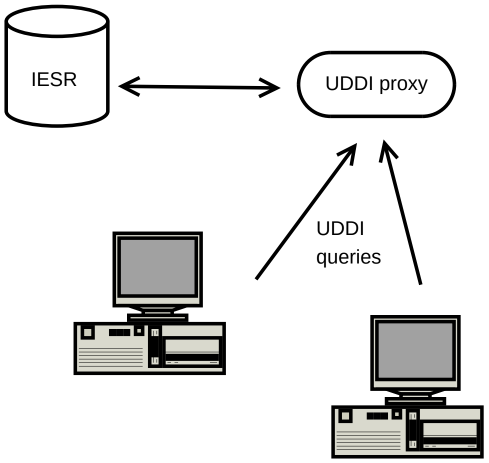

Figure 3: A UDDI-IESR proxy layer

together a coherent query, before querying the IESR layer, meaning that the session would have to persist across multiple queries, rather than simply 'rephrasing the question' as it were).

6.1.3 Choosing a mapping

From the perspective of usability the important question is not whether a mapping is possible - the answer to that is a clear 'yes', although the previously investigated method, making use of jUDDI, is unlikely to be the perfect solution. That is, a light interface layer over IESR could offer an adequate mapping, without introducing the complexity and overhead of periodic harvesting of one database into another. But as regards mapping, the important point is the following: what mapping actually responds to user needs?

In other words, users obviously make use of UDDI for a reason – if they do at all - the decision by IBM, Microsoft and SAP to discontinue their joint UDDI registry effort brings this assertion into some doubt. This decision implies that the centralised vision of UDDI was unsuccessful, that small, localised UDDI servers have proven to be the way to go - here a central IESR/UDDI interface fits into this is a question that can only be answered by deeper investigation of the reasons why this effort failed.

In order to choose a mapping, then, there are various relevant questions:

- Which clients are most commonly used?

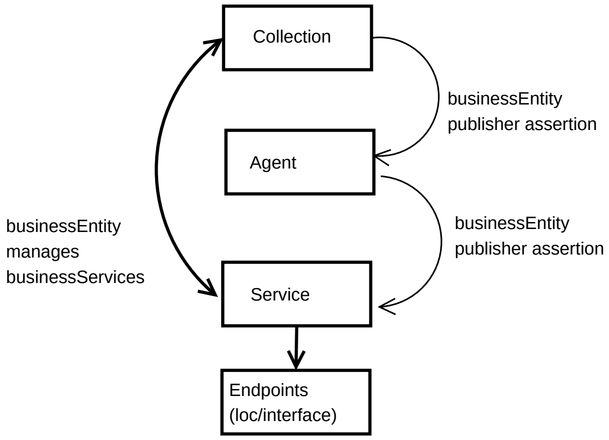
Collection
businessEntity publisher assertion
Agent
businessEntity publisher assertion
businessEntity manages businessServices
Service
Endpoints (loc/interface)

Figure 4: Collection and agent relationships mapped through businessEntity publisherAssertions, with service and endpoint information mapped as businessServices and bindingTemplates.

- Which use cases are most common, most successful and most widely adopted in the context of UDDI?
- How do these clients query the system/how do users query the system?
- Is there likely to be some difficulties with using the tModel solutions?

Three mappings were considered: see Figure 4 (*A*), Figure 5 (*B*) and Figure 6 (*C*).

A principal difference between these mappings is the use of tModels; it is not clear how accessible tModels are for search/discovery/retrieval purposes. A major advantage of UDDI is the fact that various IDEs such as Visual Studio provide simple search-and-retrieve methods and interfaces for locating handy web services, often based around simple keyword matching; if these are broken or their functionality impaired by the mapping used, this severely limits the usefulness of the exercise.

There is little data publicly available regarding the popularity of UDDI access methods. Again, this may reflect the 'closed' nature of successful UDDI installations, or alternatively imply that the system is not being widely used. We may hypothesise that the most commonly used clients are either a) bespoke software agents, or b) UDDI client use through GUI software development tools such as C++.net. The availability of tModel-based classifications to a visual environment UI is an important point; to what extent can tModels be manipulated or searched?

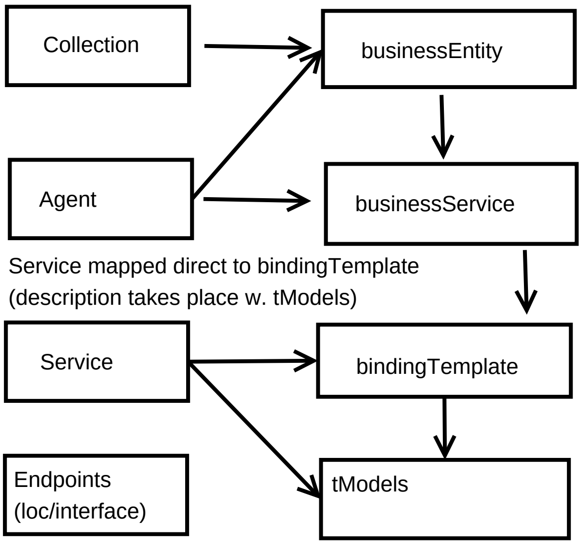

Figure 5: Collection and agent relationships mapped through businessEntity and businessServices, with all service information placed directly into bindingTemplates (described by means of tModels)

	A	*B*	*C*
"keyword search' via GUI tools	*1*	*0*	*2*
tModel-based (bespoke) search and retrieval	*1*	*2*	*1*
Browse via tModel content	*1*	*2*	*1*

However, it is clear that there is not enough data available to complete this analysis. Appropriate means for gaining this data might include : evaluation via questionnaires, task analysis and observation of UDDI usage in appropriate environments. This may ideally be completed hand-in-hand with a more general stakeholder analysis. That it is possible to browse and search through each classification method is not in doubt; that it is consistent with current practice for developers making use of UDDI is another question. Since there are clearly several plausible modes of use for the service, it remains only to examine current practice by potential stakeholders.

6.1.4 Conclusions

Examining real-world instances of present-day use of registries based around UDDI is an important step in choosing which of these mappings is most appropriate.

Our investigation suggests that the most frequently-used search method

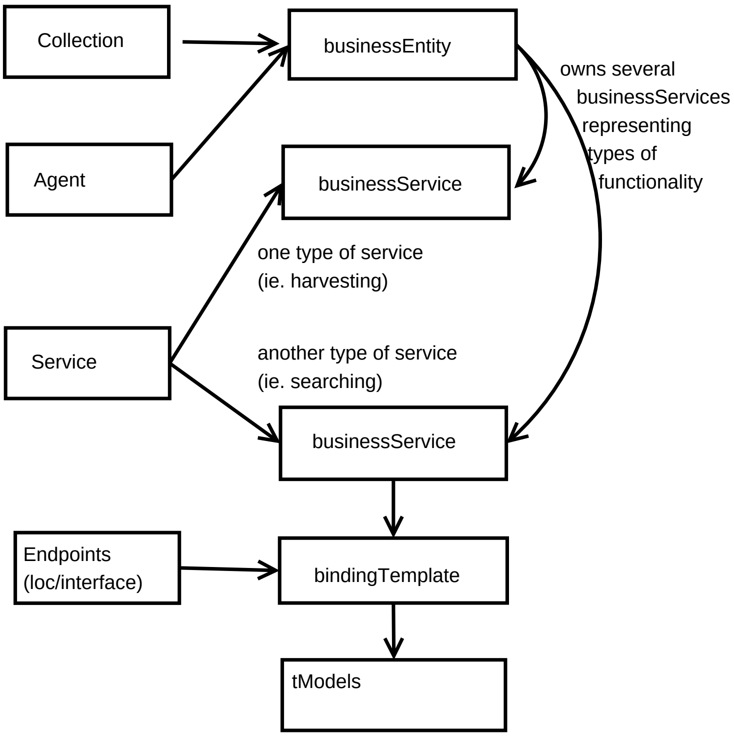
Collection
businessEntity
owns several businessServices representing types of functionality
Agent
businessService
one type of service (ie. harvesting)
Service
another type of service (ie. searching)
businessService
Endpoints (loc/interface)
bindingTemplate
tModels

Figure 6: Collection and agent relationships mapped through businessEntity, with businessServices representing each type of service available from a given businessEntity, and interface information (ie endpoints and protocols available) mapped into bindingTemplate (most closely resembles UDDI's own model).

is simple keyword search, which depends on search of plain-text descriptive text found in the service description provided within the businessService, whilst bindingTemplates are not generally searched using tModels or otherwise in the first instance. bindingTemplates in the UDDI model tend to represent various locations at which a given service can be found, and various protocols by which it can be accessed - but they are not generally used to describe the service itself, or searched to discover services of a given type. That said, it is of course possible to use bindingTemplates for this purpose if appropriate tModels are used, but the support for this strategy provided by existing software would have to be ascertained in greater detail in the first instance.

This said, the question remains as to whether human users searching the IESR via UDDI, via integrated development tools or otherwise, is actually of great relevance from the perspective of the project. If the aim is to enable machine-readable search methods via the UDDI interface, other metrics recommend themselves; for example, it becomes more important to fit in with the uses made of UDDI within Grid computing.

Em Tonkin, October 2006