ANGEL CONCEPTUAL MODEL DESIGN

Document:	ANGEL CONCEPTUAL MODEL DESIGN
Category:	Project Deliverable (4.1)
History:	v1.00 24-Jul-01 agreed by Project Board v0.03 18-Jul-01 final internal review v0.02 15-Jun-01 for review by ANGEL-Team v0.01 05-Jun-01 main diag. only; Team review
Author:	John Paschoud <J.Paschoud@lse.ac.uk>
Date:	24 July 2001

Contents:

Introduction

This document describes the high level conceptual design of the information systems that will be implemented to deliver the major objectives of the ANGEL project. The purpose of this document is to provide:

- Broad initial guidance of subsequent design and development activities within the project;
- A basis for consistent dissemination about the structure of the ANGEL systems, to others outside the project.

1. How ANGEL relates to external services

ANGEL will be largely invisible to end-users. It will be designed as (mainly) a collection of middleware components that will provide services to independent applications, that in turn offer other services and (probably, but not necessarily) Web-based user interfaces to end-users. We have classified such front-end applications into two groups:

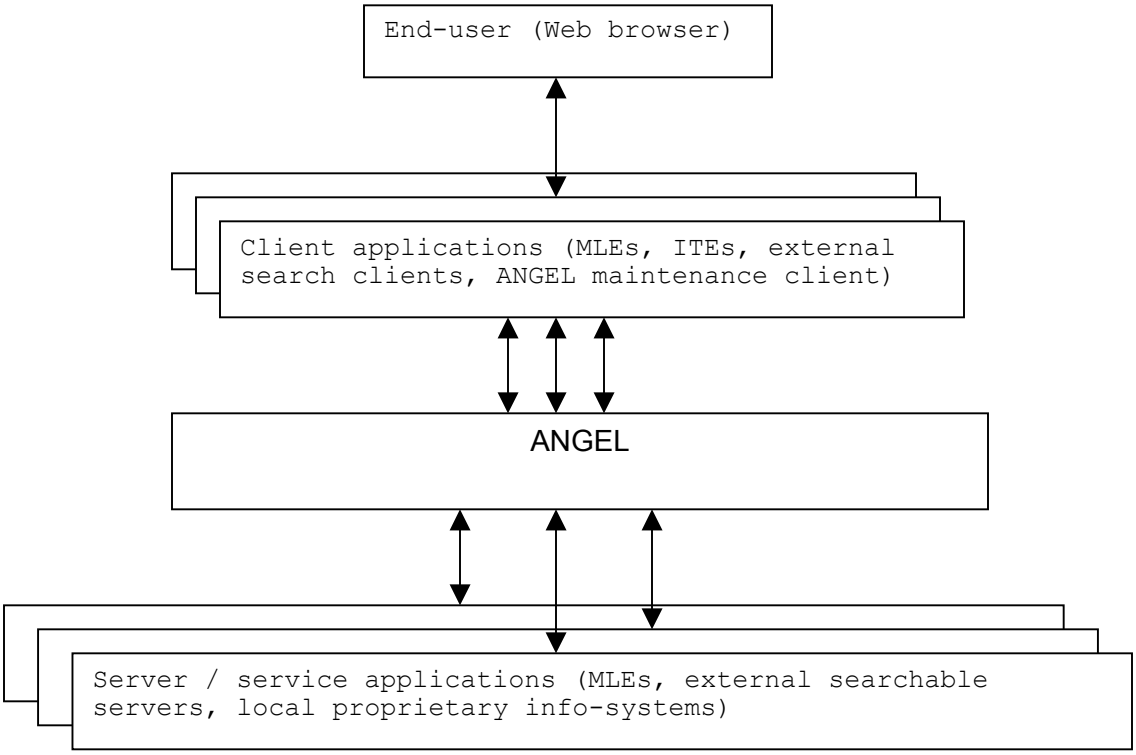
- 'Managed Learning Environments' (MLEs), which should be interpreted to include so-called "Virtual Learning Environments", and would include products such as WebCT[i], Blackboard[ii] and others that conform to the DNER definition[iii] of VLEs, and meet ANGEL's MLE Minimum Scoping Specification[iv].
- 'Interfaces To Everything' (ITEs) - a term we have coined to describe the (usually) institutional portal developments that aim to be a single point of access to "everything" that an institutional end-user needs to access. "LSE for You"[v] and the "Edinburgh Student Portal"[vi] are two such examples.

At least one native end-user interface to ANGEL will be provided, to enable management of configuration and internal metadata. Ideally this will be implemented as purely XSLT (XML Style-sheet).

ANGEL will also inter-operate with external (and possibly unknown) clients by offering a variety of standard protocol server interfaces, including the capability to act as a Z39.50 target (server). An ANGEL system will normally be configured to operate on behalf of a single academic institution, or possibly a consortium of smaller institutions that can agree to offer a single set of resources and services to all of their end-users.

At its' 'bottom-end', ANGEL will act as a client to various external services at which data can be searched and retrieved, using standard protocols (such as Z39.50 and IMS), and to various other information management systems (such as the patron management elements of a library management system) using standards-based or proprietary APIs. An architecture is also envisaged in which ANGEL service systems installed independently by different institutions can inter-operate, mainly to enable managed sharing (and shared management) of resources metadata. It is likely that some applications that act as clients of ANGEL (such as a MLE, that uses ANGEL to find library catalogue resources) will also act as servers to it (such as a MLE, that ANGEL can query in IMS terms to find subject-relevant course material).

On a very abstract and simplified level, the position of ANGEL could be depicted as:

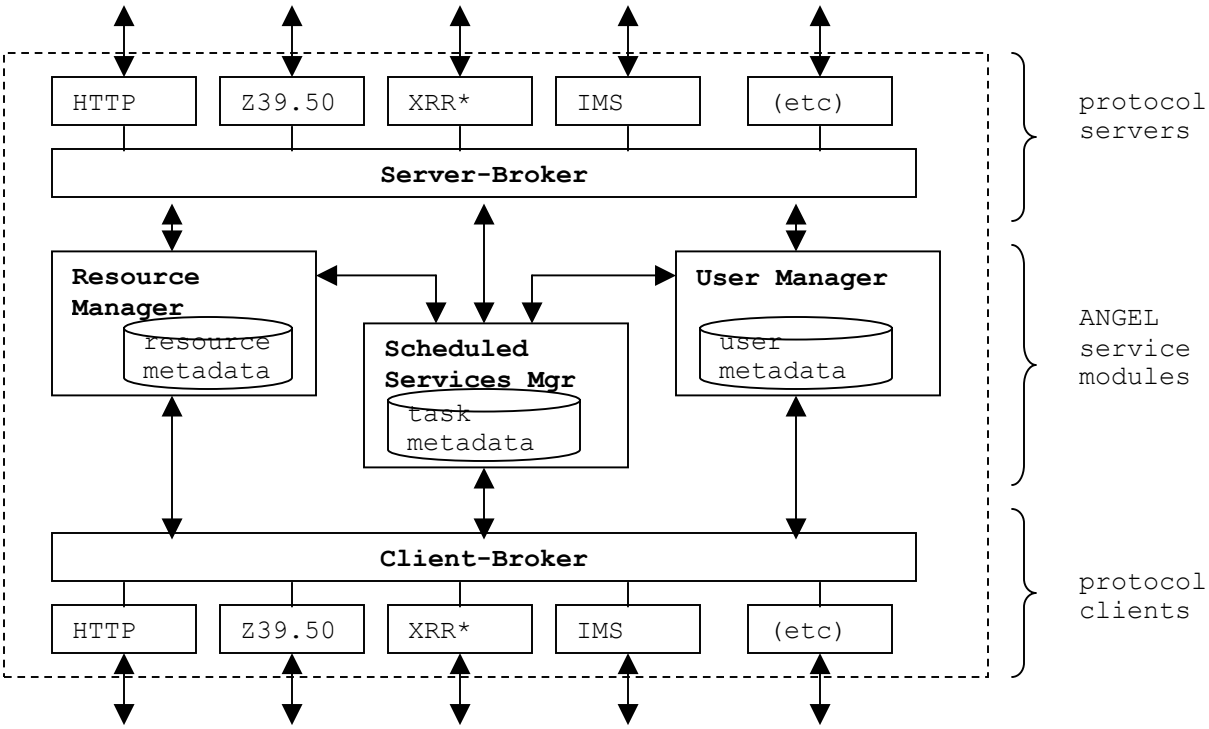


2. *Internal Architecture*

The internal structure of ANGEL will reflect these relationships, but need not be known to the external clients and services shown above. It can be viewed as three distinct layers. Central to the functionality that ANGEL provides will be three core service modules:

- The Resource Manager (RM), controlling metadata about information resources,
- The User Manager (UM), controlling (or interfacing to externally-managed) information about end-users,
- The Scheduled Services Manager (SSM), providing independently scheduled 'robot' functions such as current awareness alerting and maintenance checking of referenced URLs and other application-specific links between distributed service components; containing information about tasks and scheduling.
-

The RM and UM will together provide the functions of end-user authentication, and of authorisation of access by end-users to specific resource content. This functionality is described in the paper: "ANGEL Authentication & Authorisation Model", produced as Deliverable 6.1 of the ANGEL project.



A set of protocol-specific servers, managed by a Server-Broker (S-B), provides service connections to respond to requests from client applcations such as MLEs, ITEs or external search clients. For many types of request, a client will need to be recognised and registered with the ANGEL system, although some services (such as a Z39.50 search) may be configured to respond to any valid client that knows the appropriate URL and port address to request-to.

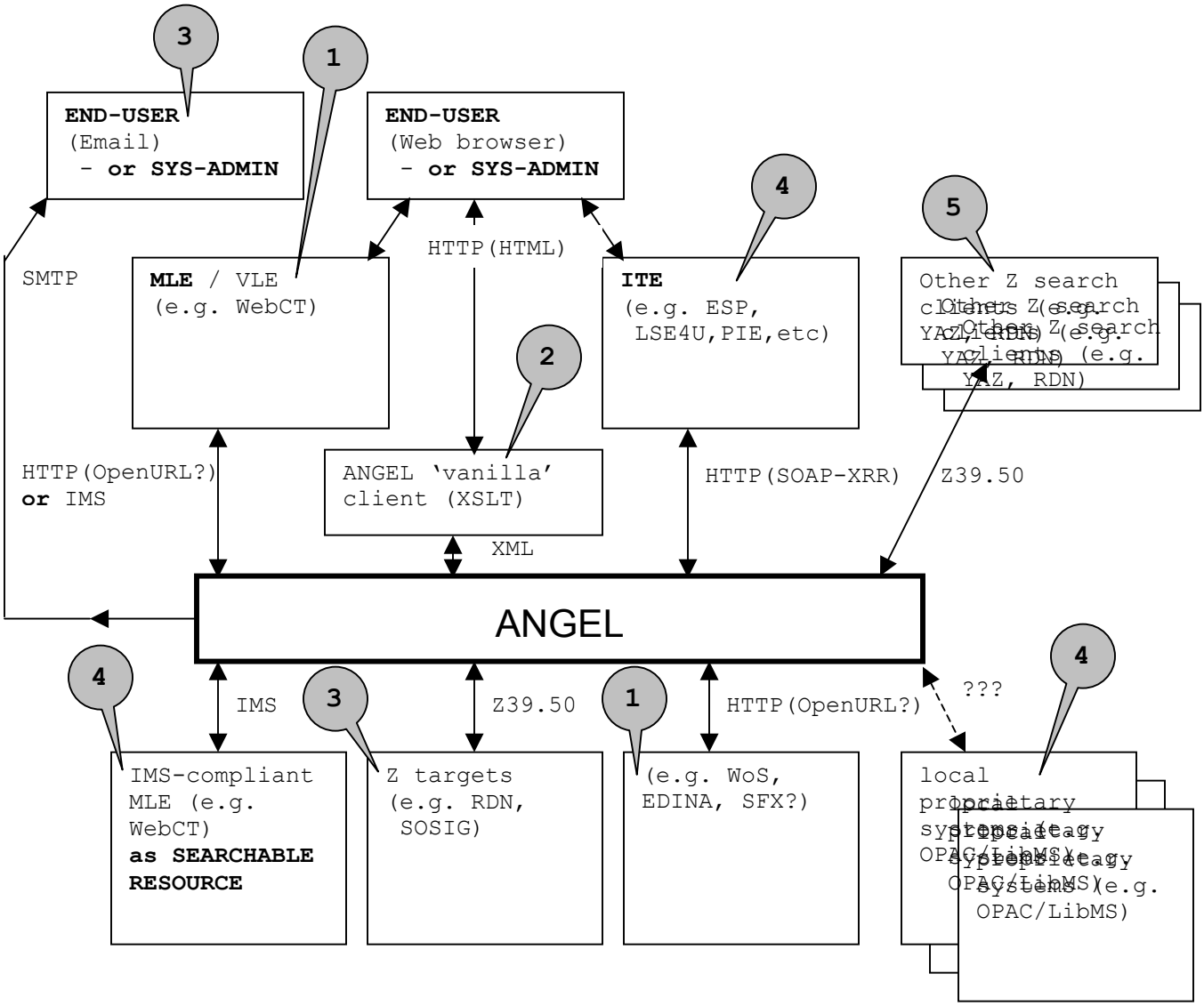
A similar set of protocol-specific clients is managed by a Client-Broker (C-B) to make requests of information services (or functional services) external to ANGEL. Some of the protocol servers (and possibly the clients too) should be built in a multi-layered way - e.g. to deal with an OpenURL[vii] within a HTTP 'get' or 'post' request. The "(etc)" protocol of the diagram represents an extensible variety of standards-compliant or proprietary interfaces that can be added to implement service links to other systems within the institution. The ANGEL Project will fully publish details of the interface between Brokers and protocol interface modules, to enable such further development by third parties or in-house by institutions adopting ANGEL. Standard protocols to be considered and implemented by the ANGEL Project are explicitly not limited to those shown on the diagram. For example, Scheduled Services will require the use of SMTP or other email protocols.

Each of the two brokers will be relatively closely coupled to its' set of protocol-specific interfaces, with each of the two probably running as a single executable. But the 5 major ANGEL components (RM, UM, SSM, S-B and C-B) will communicate using a more open, XML-based protocol, provisionally labelled here as "XRR". This will enable scaleability and performance management of large scale ANGEL services in an institution, by allowing the internal architecture to be distributed across a number of physical servers. We should also attempt to enable scaleability and fault-tolerance, by ensuring that the architecture can accommodate multiple instances of each component, and cope with possible redundant duplication of the data held within the RM and UM.

The protocol "XRR" is a (yet to be defined, and properly named) XML-based request & response protocol, used for internal communication between the service modules and the interface brokers, and also available to external ITE applications, or to enable inter-operation between two or more ANGEL systems. XRR will be based on the (well-defined) Xrep protocol that was designed for implementation of the Decomate2[viii] system. XRR may be used internally between the components of a single ANGEL service in an 'unwrapped' form, and also made available for external applications in a 'wrapped' form, such as the SOAP[ix] Webservices architecture. "XRR" will be defined in the parallel paper "ANGEL Internal Messaging Specification", produced as Deliverable 4.2 of the ANGEL project.

3. Examples of Clients and Services connected to ANGEL

It may also be helpful to define the functions of ANGEL and the possible contexts in which it is expected to inter-operate by examples. It should be stressed that any typical institutional ANGEL service is unlikely to implement all of those described here, and that these are intended to be illustrative and do not constitute definite commitments by the ANGEL project to implement specific protocol interfaces, or interfaces to any particular services. Similarly, this is not intended to be an exhaustive list of all the interactions that are likely to be possible via ANGEL.



1 The content of a directed learning package, previously prepared by an instructor and navigated by a student within a VLE package such as WebCT, may include a link comprising a URL that conforms to the OpenURL specification. This could be intended to retrieve the "appropriate copy"[x] of the full-text content of a journal article (or other 'library' resource) - from whatever holding or access arrangement is most cost-effective, or most accessible to this particular student user at their current location. This might not be the same source as the "appropriate copy" found by the instructor when compiling the course material (because the instructor may have had different access privileges to those of the student). The ANGEL Resource Manager will locate and deliver the material, with reference to knowledge of the current access authorisation rights of the student user, negotiating user authentication barriers where necessary. If the material is not currently accessible (in some immediately deliverable form), ANGEL will 'degrade gracefully', offering the student information to help access the material by locating a print copy holding (in some physical library to which the student can gain access), or negotiating an inter-library loan request.

2 Library and learning resource administrators will use a 'thin', ANGEL-specific client web interface to directly maintain the internal resource and user metadata that configures the institutional ANGEL services for which they are responsible. If this is to update changing target URLs of target resources, or renewed licences to commercial information services, they may be prompted to do so by email messages from…

3 …the ANGEL Scheduled Services Manager, which can be tasked to periodically check such non-institutionally-controlled (and therefore "unreliable") access. Similarly, the SSM could be tasked by an end-user (instructor or student) to periodically repeat a search (via the Resource Manager) for items matching a topic of interest, emailing any new results found from either bibliographic databases or accessible full-text sources.

4 A student user of a non-IMS-compliant institutional portal (an "ITE") can follow a generic "Angel"-labelled link, from information identifying an institutional course (or timetabled lecture session for a course). ANGEL can search for and return (or, return direct links to) IMS-compliant repositories (such as a combination of WebCT and Blackboard courseware servers) material labelled as content for the course or course-event. A parallel search by ANGEL of data proprietary to the institutional Library Management System (LMS) may also find information on course-specific reading lists stored therein, or user-specific information on relevant print items currently on loan or on reserve for the student.

5

The institutional effort spent on maintaining collection-level resource metadata within the ANGEL Resource Manager can contribute to the national content infrastructure of the DNER, presenting the collections described within the RM as a Z39.50 target to external searches from Z39.50 (or RDN-compliant) clients, such as those built by the eLib 'clumps' projects.

It is envisaged that richer functional interaction, particularly with respect to distributed resources metatdata, will be possible directly between two or more institutional ANGEL services (or, between multiple co-operating ANGEL servers within a single host institution). Each would treat others as an external XRR client or server, depending upon the type of transaction and its' initiator.

References

---

[i] WebCT: http://www.webct.com/

[ii] Blackboard: http://www.blackboard.com/

[iii] JISC/DNER VLE definition: http://www.jisc.ac.uk/pub00/reg-vle.html

[iv] *ANGEL MLE Minimum Scoping Specification: [John Eyre – **TBD**]

[v] LSE for You: http://www.lse.ac.uk/admin/mis/MyInformationService.html

[vi] Edinburgh Student Portal: [John MacColl – **TBD**]

[vii] OpenURL (standard under development): http://www.niso.org/commitax.html

[viii] Decomate2 archive: http://www.bib.uab.es/decomate2; Xrep DTD defined in: *D3.1 Technical Design Core System* at: http://www.bib.uab.es/project/eng/delirepo.htm

[ix] SOAP (Simple Object Access Protocol) Webservices architecture: http://www.w3.org/TR/SOAP

[x] The "appropriate copy problem": http://www.niso.org/CNRI-mtg.html has a good definition, mainly the work of Priscilla Caplan and Dale Flecker.