TELOS, a Service-Oriented Framework to Support Learning and Knowledge Management

By

Gilbert Paquette, Ioan Rosca, Stefan Mihaila et Anis Masmoudi

Centre de recherche CIRTA (LICEF), Télé-université

Abstract

This chapter presents the basic orientations, the main use cases and the conceptual framework of a TeleLearning Operating System. TELOS is a system under development within the LORNET research network (www.lornet.org) aiming to integrate components and services, and research results, produced by the different LORNET teams. TELOS research is at the convergence of three main trends: learning object repositories that facilitate the access to knowledge resources; learning and knowledge management support systems that use these referentials as building blocks; and the integration of these referentials and these systems in the context of the Semantic Web.

1. Introduction

There exists actually many specialized tools and hundreds distance or e-learning platforms (WebCT, TopClass, LearningSpace, Ingénium, Docent, etc.), also called Learning Management Systems (LMS) or Learning Content Management Systems (LCMS), available through a Web browser. There are also a number of comparative studies of their features and even a decision system to select an e-learning platform.[1]

Recent reviews of e-learning platforms show that there are not great differences between them. The current platforms are mostly designed for predefined actors (author, trainer, learner). They are focused on predetermined delivery models for self-training and on-line asynchronous conferencing. Actually, most e-learning platforms look more like an extension of the former CBT authoring tools. Their efficiency as quick authoring tools for the web is often achieved by reducing drastically the variety of instructional strategies, each course having similar structures and components.

The advent of learning portals and Web services presents an interesting evolution towards more flexibility, presenting another vision of learning than just giving access to a predefined, preformatted and pre-digested content. Even though learning portals also need delivery platforms, more important are the services around which the portal is organized: access to courses or learning resources, tools to define a path of learning events, links to training support by individuals or organizations, peer-to-peer collaboration services and communities of practices, access to a reservoir of content experts and so on. The true potential of learning portals is only slightly uncovered and needs to be fully exploited based on new research and development activities.

Compared to the evolution of generic software (text editors, spreadsheets, etc.), e-learning systems are now in a similar position than the integrated software of the last decade where text, spreadsheets and database éditors could transfer data only within the integrated suite These have been replaced by integration mechanisms at the operating system enabling data communication between any compliant tools through desktop operations.

[1] For this, consult www.brandon-hall.com)

The TeleLearning Operating System (TELOS) will provide similar flexibility for e-learning environments. It will allow the implementation of interactions between actors using resources dynamically related to the operations they perform in the system. Hence, within TELOS, by aggregating resources and functions differently, it is possible to build quite very different distributed learning environments such as electronic performance support systems (EPSS) integrated in a workplace activity, communities of practice or, at the other end of the continuum, formal on-line training and technology-based classroom activities, as well as different forms of blended learning or knowledge management.

A new generation of Web-based learning delivery systems is needed to integrate advanced solutions to interoperability problems, aggregation and flexible designs, adaptive agents, knowledge extraction from documents and advanced multimedia objects processing. The actual LCMSs and portals do not provide many functionalities yet and are far from integrating them in a coherent integrative (instead of integrated) system. The integration of the structures or the processes can be made more flexible by sharing knowledge and ontologies referentials, by the coordination of the communication between the actors, by interoperable technical infrastructures, by standards of representation of the aggregation formulas and by the use of aggregation facilitating interfaces. The TELOS system will integrate these strategies in an organic vision about the educational phenomena and on models of the aggregation process used as facilitating multi-actor interfaces.

In the last five years, an increasing number of organizations have recognized the importance of learning technologies and knowledge management. This has resulted in attempts to identify, formalize, organize and sustain the use of their knowledge, through the reengineering of work and training processes and the ubiquitous use of Web-based technology [Davenport & Prusak 1998, Sveiby 2001]. Knowledge Management is now strongly influence by the building of Communities of Practices [Wenger 2001] based on research in Computer Supported Collaborative Work and Learning (CSCW/CSCL). In the same time frame, an important international movement headed by IEEE and IMS, is elaborating standards enabling users in different parts of the world to interoperate and reuse computerized resources, "learning objects" or "information resources" made available in "learning object repositories" distributed on the World Wide Web [Wiley 2002]. These major trends converge and integrate in another ambitious effort to construct the next Web generation, the "Semantic Web" [Berners-Lee et al. 2001]. The Semantic Web aims to associate knowledge with documents, and more generally to learning objects, distributed on the Web.

These major international trends form the basis of the LORNET research program and underline the need for a Web-based system like TELOS to support learning and knowledge management. Although many initiatives are blooming in this area, the TELOS is unique by its goal to integrate the major approaches and technologies that can contribute to support learning and knowledge management on the framework of the Semantic Web.

This research can be situated in the context of the growing field of *Service Oriented Conceptual Frameworks*[2]. A *framework* creates a broad vocabulary (an ontology) that is used to model recurring concepts and integration environments and is equivalent to the concept of a *pattern* in the software community. A framework supports the development by organisations of their own implementation infrastructures, using a flexible *service-oriented* approach.

Service-oriented frameworks [Blinco et al 2004; Wilson et al 2004] are rapidly gaining popularity with the wide adoption of Web Services, and because of the lower costs of integration coupled with flexibility and simplification of software configurations. In a service-oriented approach, the application logic contained in the various systems across the organisation, such as student record systems, library management systems, Learning Management Systems (LMS) and so on, are exposed as services. Each service can then be utilised by other applications. For example, a student record system may expose services defining student enrolment and registration processes and related information, which can then be

[2] For example of eLearning frameworks, consult [CETIS-JISC 2004] [IMS-AF 2003] [OKI 2003]

used by a learning management or a library system.

The ultimate aim of a Framework is, for each identified service, to be able to reference one or more open specifications or standards that can be used in the implementation of the Service. A framework can support a number of organisational infrastructures that are still coherent and consistent with respect to one another. A framework does not aim to build a generic learning or knowledge management system. One of the primary goals of a Framework such as the one presented here is to encourage "coherent diversity", by providing alternate service definitions which can then be used to meet the diverse goals of the organisation.

Following in this chapter, we first define the basic orientations of TELOS. Then, we present the main use cases that have served to identify the services composing the system. The last section will synthesize these components in a service-oriented conceptual framework describing the organization of the system, before we conclude on future research that remains to be done.

The ideas presented here are the result of work that has started at the LICEF-CIRTA research Center since the beginning of the nineties : on the concept of virtual campuses [Paquette 1995], on the Explor@ system seen as learning portal generator [Paquette 2001], on the construction of a Web-based support system for instructional designers called ADISA [Paquette, Rosca et al 2001], on resource aggregation [Paquette et Rosca, 2003] and learning resource management [Paquette et al 2004]. This chapter in mainly based on recent results achieved within the LORNET[3] network by the authors [Rosca 2005; Paquette et al. 2005].

2. TELOS Orientation and vision

We first present now ten orientation principles that guide the development of the TELOS system, a general 4-level description of the system and the interaction between its main actors.

2.1 Orientation Principles

Here we present the main orientation principles that lead the development of the architecture and of the TELOS system.

Solving Real Learning and Knowledge Management Problems. The TELOS system aims at facilitating learning and knowledge management activities. This entails the need to examine real educational and knowledge management problems, to analyze them thoroughly, to observe future users of the system very early in the project and to provide solutions to real user problems, not only in terms of system's tools, but also in terms of processes to use them effectively in real contexts. We must avoid being technologically driven instead of solution-driven, so the driving force is the careful definition of use cases that guides the design of the architecture and the development of the system.

Reusing and Integrating Existing and New Tools. LORNET is an oriented research project aiming to integrate technologies from different fields and to develop new ones when they are educationally significant. We reuse, as much as we can, existing editors, communication tools, interoperability protocols and specification from norms and standards international bodies, guided by uses cases that underline the need for new tools or new ways to assemble or extend them. In these activities, we focus on specific TELOS core components that facilitate the reuse of existing tools by their users.

Concentrate on Essential Developments - Reduce risks. The goal of the architecture is to reduce the risks by shifting the accent from tool development to careful analysis, evaluation and well-planned

[3] LORNET (www.lornet.org) is a Canadian research network led by the first author and financed by the Canadian Governement and private companiges for five years until 2008. The network groups 6 research centers and laboratoiries and over a hundred researchers, research professionals and graduate students. L

specification. This will enable the TELOS team to focus on essential developments, and leave more costly development or adaptation to industrial, university or public partners in the network.

Flexible and Pragmatic Aggregation. Pragmatic aggregation means a convergence of technological means and human interventions or interactions to achieve certain goals. So the system should have enough flexibility to be used in a variety of situation, from formal well-planned instruction, to more or less structured self-training, emerging communities of practice or performance support systems integrated with work environments. The success of TELOS will come from its demonstrated utility in a diversity of situations.

A Society of Human and Computer Agents. Software engineering sometimes sees the "system" to be solely composed of software components separated form their users. At the opposite, we adopt a multi-agent view where human and computer agents are interacting components of the system, providing services to each other. Extending the "Human-in-the-loop" we recognize that sometimes, organizational adaptations, advising, documentation support or human communication activities can be more appropriate (and less costly) than building new tools. This approach also favors maximal results with realistic efforts.

Build Technology-Independent Models. The important work involved in the TELOS system should survive the rapid updating pace of technologies in general. At the start, it enable TELOS to operate on different, network, hardware and operating system configurations, and to integrate it with other learning or knowledge management systems. The architecture is built to protect the conceptual models from technological instability. The conceptual specifications are kept separate from any implementation. The TELOS system should then be able to reuse such "conceptual programs" despite different previous technology environment, and adapt it to new technological implementations. So the conceptual models are not just prerequisite to building the TELOS system; they are part of the system, as one of its most fundamental layer.

Learning Ecosystem Models for Planning, Support and Evaluation. Most distributed learning systems today do not have a model of the processes, the users, the operations and the resources that they intend to support. Besides direct support for learning and knowledge management tasks, we aim to introduce tools to model the complex processes involved in a distributed learning system, before its use (to design it), during its use (to support users and observe their behavior) and after its use (to evaluate and maintain the system). These modeling components and tools are built-in features of the TELOS system. They aim to enable users to interact efficiently in pre-planned, as well as emerging and user-controlled events where the initial environment is transformed, thus implementing a "learning ecosystem" approach.

Modularization and Layer Independence. The very flexible system envisioned will amount to a very small kernel, at a very high level of abstraction, capable of assembling services that generally form the core of a system, for example functions like learning object aggregations, component coordination and control, ontology-based indexation and search, function modeling and so on. The architecture will promote horizontal modularity: horizontally between components, and vertical layer independence from an abstract representation, to a concrete implementation, to a run-time version of TELOS applications.

Construct Reusable and Interchangeable Models and Components. TELOS being model-oriented, it then becomes possible to implement the model components in various forms and alternative tools, classified by their functionalities and grouped in interoperable classes. TELOS then appears as a flexible assembly system enabling the integration of tools, already existing or to be produced, by various groups, to support a variety of learning and knowledge management models. Even at the kernel level, the general functions could be covered by one or more alternative "kernel" modules, accessible on a service bus for selection by system configurators and designers.

An Assembly and Coordination System. The TELOS kernel will then be mainly a set of computer agents capable to play parts of "TELOS scripts", maybe using alternative compilers or interpreters within different operating systems. TELOS will not be another huge distributed learning platform or a system to generate rigid platforms, even though it can assemble components specific to some intended set of

applications. The term "TEleLearning Operating System" should be seen as a metaphor. TELOS is planned essentially as a set of coordination and synchronization functionalities supporting the interactions of persons and computerized resources that compose a learning or knowledge management system.

2.2 System's Levels and Main Actors

Figure 1 shows a cascade of more and more specific system levels and their corresponding actors. The TELOS core is managed, adapted, extended by system engineers. With it, technologists in different organisations produce one or more TELOS Learning and Knowledge Management System (LKMS), each generalizing the idea of an "on-line platform" adapted to an organization's particular needs. Unlike the present situations, each platform is extensible, and its components are reusable in other platforms.

With any TELOS LKMS, designers can create, produce, deliver and maintain a number of Learning and Knowledge Management Applications (LKMAs), that is courses, learning events, knowledge management portals, etc. The LKMS is a platform for the aggregation of resources, activities and actors. Each LKMA, composed using a LKMS, groups one or more actor-specific aggregates called Learning and Knowledge Management Environments (LKMEs) intended for certain types of participants: learners, content experts, coaches, tutors, evaluators, managers, etc. An LKME is an aggregate of documents, tools, services, operations (activities) and actor agents.

Before delivery, an LKMA and its different LKMEs are instatiated by an actor called an application administrator to start a new session involving a group of participants. Using these instances, the participants produce results, that is resources and outcomes called Learning and Knowledge Management Products (LKMP) which are stored in a database for reuse or adaptive support.

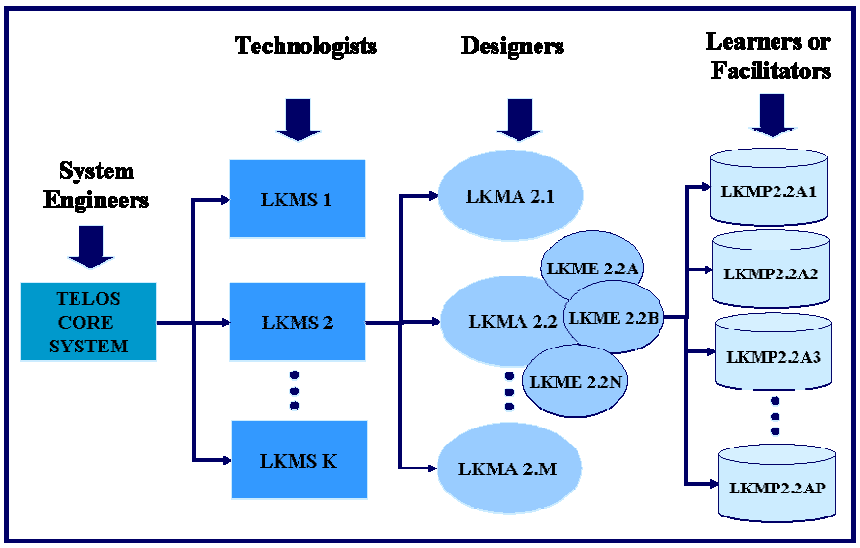

Figure 1 – Four-level Cascade of Systems or Products

Figure 2 presents the five main actors on figure 1 and the communication pattern between them. Learners and facilitators normally use a learning and knowledge management application (LKMA) that provides them with communication channels. This LKMA can be a structured environment built by a Designer using a learning and knowledge management system (LKMS), or simply a set of general TELOS services for emergent exploration. A LKMS is a generalization of the actual LMS, LKMS, CMS or platforms assembled by a technologist using the TELOS system core and the libraries developed by a TELOS engineer.

Note that the roles played by a person are interchangeable. Some persons acting as learners or facilitators can also be their own designers, and their own technologists or even engineers. Conversely, persons building environment can also be facilitators to designers. The interchange of roles can be provided by

software role agents providing an interface between a human or system actor and the functionalities and data needed to achieve that role. Also some of the roles can be played by software agents, for example an intelligent advisor agent or a peer help or resource finder acting as a facilitator.

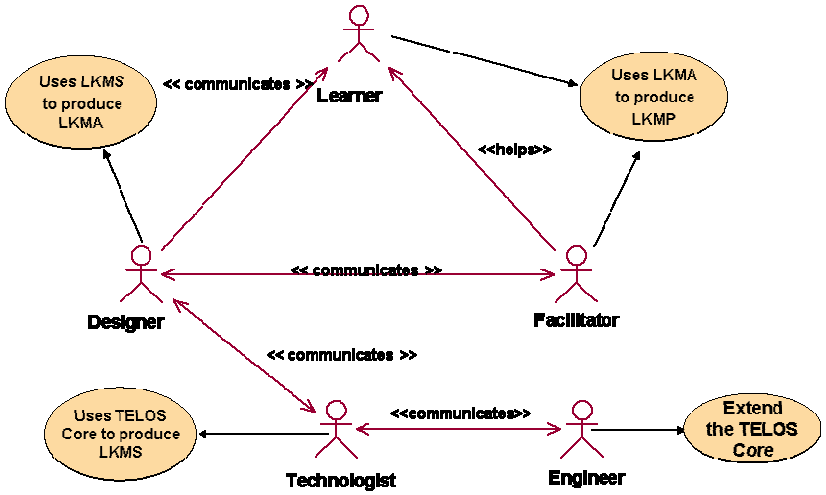

Figure 2 – Interactions Between the Main Actores

3. User Operations and Main Service Components

TELOS is a specialized operating system on top of a computer network designed to support learning and acquiring knowledge at a distance. In this section, we present the main TELOS use cases and a first overview of the TELOS components needed to support these use cases.

3.1 Three Operational Levels

Figure 3 presents a first conceptual view of the system by presenting its main use cases.

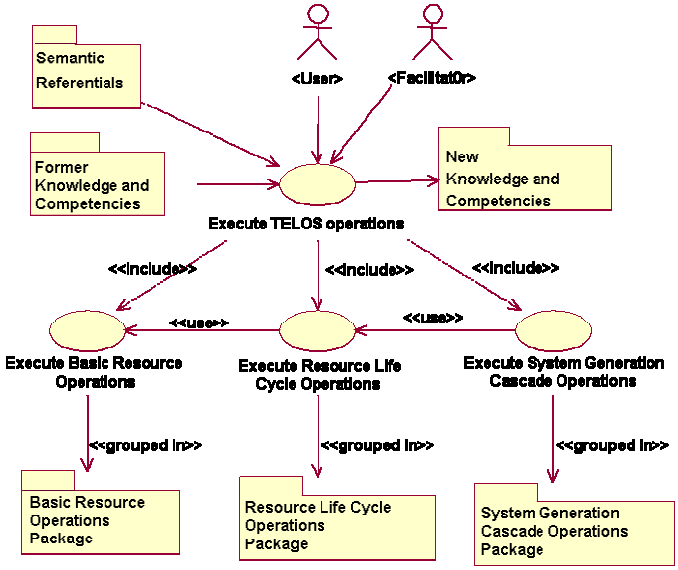

Figure 3 – User Levels of Operations

A TELOS user (or a team of users) possibly helped by a facilitator takes the responsibility to perform a TELOS operation. In this operation, users and facilitators use or modify resources in the TELOS Core and produce new resources that sometimes are put back (embedded or referenced) in the TELOS Core.

Every time a user performs an operation, his/her previous knowledge and competencies are changed to new ones. This fact is the essence of learning by doing and doing by learning. In the TELOS system, it is possible to represent explicitly implied knowledge and evolving competencies related to the resources (persons, operations, documents and tools) using one or more semantic referentials. Semantic referentials can take the form of standard or specific metadata, classifications and taxonomies, thesauri or ontologies.

In TELOS all the operations are driven (or at least initiated) by humans, always through some user-interfaces and mediated by computer programs. There are three basic sorts of operations depending on their level of granularity.

- *Basic operations on a resource* consist of asking or delivering a service using a resource either directly, or indirectly, mediated through a TELOS agent provided by the system or mediated by another resource- through that resource's designated delegate (resource using the systemagent) or through another resource.
- *Resource life cycle operations* consist of a series of four sub-operations (phases) where a resource is composed, managed (prepared) for use, used in some activity, and analyzed, providing feedback to start, if necessary, a new resource life cycle. These operations are generally performed in sequence by a team of different actors.
- *System generation cascade operations* are even more global. They consist of extending the TELOS Core with new resources, using it to produce a Learning and Knowledge Management System (a platform), designing with it one or more Learning and Knowledge Management Applications (courses, learning events, knowledge management environments) and finally, using these applications to learn and to produce results grouped in a LKMP (Learning and Knowledge Management Product). These operations are generally performed by different actors as presented in figure 1 and 2.

The relation between the three levels of operation is represented on figure 3 by "use" links. A system generation cascade is generally composed of many resource life cycle operations. Also, a resource life cycle is composed of many basic operations.

We can use different metaphors to describe these general processes. In a manufacturing metaphor, the resource life cycle corresponds to a process where a product passes through different productions operations. In the system generation cascade, the TELOS Core is like factory that produces machine components or complete machines; the products of this first factory are used to build machines that will be used in another factories (LKMSs); those machines are used to build cars, aircrafts, etc. (LKMAs), which are used by end users to produce an outcome. Using a biological metaphor, a simple operation corresponds to a life moment of an organism; a resource life cycle is an ontogenesis, the process of an individual organism growing from a simple to a more complex level. Finally, a system generation cascade is a phylogenesis, a process that generate new organisms from parents, similar to the evolution of life.

These images are important to understand the role of the TELOS Core within an evolving TELOS system. It is similar to the genome, the code of life that composes an organic system at a moment we could call its birth, when it starts to evolve in the hands of its users towards a more and more complete and useful system. Also, as a manufacture, the TELOS Core itself starts with a complete set of components to produce LKMS factories, but it will also be open to improvement, adding new processes and operations, to produce more versatile machines.

The role of this TELOS framework is thus to identify clearly what basic components are needed in the TELOS Core for this evolution process to start. To achieve this, we will represent conceptually the system at a fully operating stage, at a certain time Core (t) after the system has evolve from its Core (0) position.

3.2 Basic Operations on a Resource

We will start identifying components of the TELOS Core by first looking more closely at basic operations. Whatever his/her role in the system, a user needs to ask or deliver services by performing one or many basic operations. The simplest case is when the user interacts directly with a resource through its user-interface (UI), if it has one. This resource, for example an email client or a piece of simulation software, is referenced in a Resource Library within the TELOS Core or obtainable from a search in external resource repositories. The user obtains the resource interface that enables him or her to interact with the service. When a separation between a resource's UI and its internal logic exists, possibly with a network distance in-between, the user must obtain this resource's UI thus enabling him or her to interact with the services it offers. For this, we need a resource distributor provided by the central part of the TELOS Core called the *Kernel*.

Two other situations are displayed on figure 4 that require also an agent distributor component within the Kernel. In these cases the user does not act directly on the resource but through an agent (mediator) provided by the central part of the TELOS Core called the *Kernel*. In one case, a *user agent* can represent the user in the system to mediate his or her interaction with the resource. For example, the user agent can provide filtering of the operations with the resource or trace the user/resource interaction to prepare adapted advices. In the other case, the user will interact with an intermediary resource that will use a *system agent (SA)* to obtain the services of the resource. This is done through the TELOS communication hub and a communication coordinator within the Kernel will take care of the communication operations.

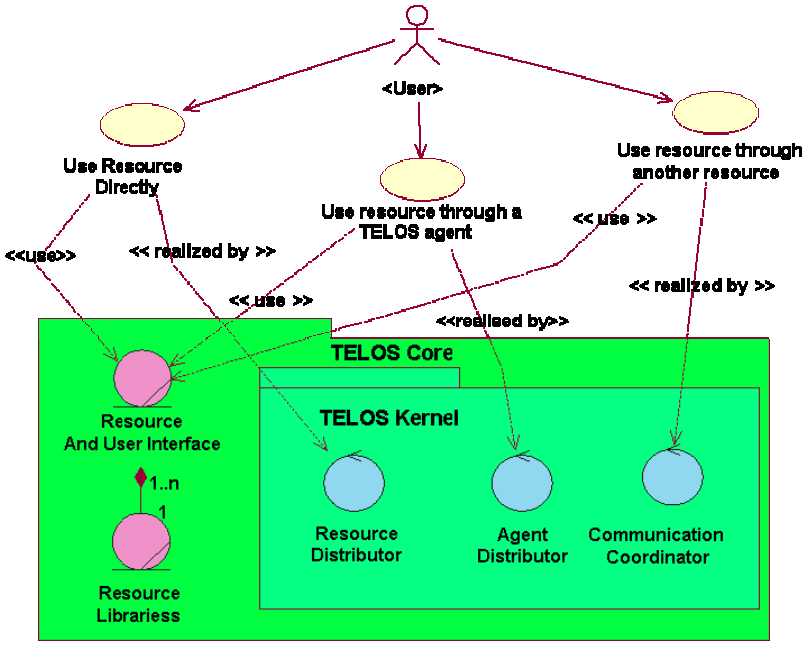

Figure 4 – Basic Operations Involving the TELOS Core

3.3 Resource Life Cycle Operations

As shown on figure 5, a resource life cycle is composed of four basic operations (or phases): compose, manage, use and analyze. These operations form a sequence with feedback loops. They are performed by corresponding actors: composers, managers, explorers and analysts that use corresponding tools or services.

The resource life cycle begins with the composition process where a composer creates, edits, or composes a model or template for a class of reusable resources, using an authoring tool. Depending on the type of resources, different authoring editors can be used, such as a simple text editor, a more complex learning design editor, or an LKMS aggregator. For example, with a learning design editor, a user will produce a learning scenario or a process structure that may be concretized at the next phase by associating precise documents, services or participants to objects in the scenario model.

In the next resource life cycle phase, a manager (a person or a team) may reference the resource model in a TELOS repository, compose a metadata description, produce a certain number of "concretizations" of the resource (versions of the resource model) obtained by parameterization, concretization, adaptation, and finally activate the concrete resources for the use phase. For example, in this phase, a course Manager will assign students and tutors to a version (instance) of a course for a certain session and prepare their communication within a forum and a videoconferencing system.

In the third phase, an explorer finds and uses the resource instance, producing exploration traces: annotations, logs, and other products, for example messages exchanged in a forum, an exam or an essay.

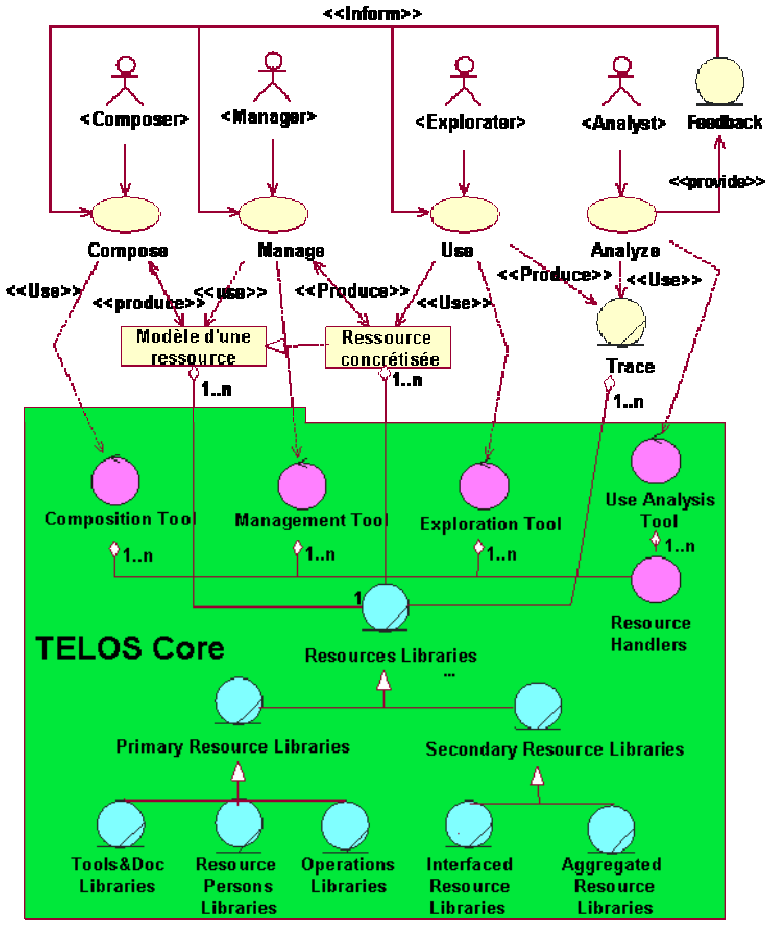

Figure 5 – Resource Life Cycle Operation and Related Components

Finally, in the last phase, an analyst observes the exploration products (during or after the use of the resource instance) and analyzes this data to offer some useful feed-back to the other resource life cycle actors. This feedback can be made available in a simple Web page to all participants, or take the form of an advice specifically addressed to the resource explorer, manager or composer (or to improve itself). For example, data from a forum-based learning activity can be analyzed and recommendations can be issued

to the composer to modify the assignment, or to the Manager to add participants or to separate the group in smaller teams, or to the explorer to interact more regularly.

All the users and the tools implied in the cycle may be equipped with agents (interfaces) connecting them to the services hub. Located within the TELOS Core there are system tools that provide communication between users, resource Handlers and resource libraries or repositories. Handlers are services provided by TELOS at different levels of the system.

The Resource Life Cycle can be applied to all sorts of primary resources: documents that provide information to users, tools that help user to process information, persons acting as information resources, or operations descriptions that provide ways to process information. The lifecycle of a primary resource may apply even to an application's (tool) software development process, where for example we have the developers in the role of "composers". The primary resources are referenced in the Primary Resource Libraries, integrated in the TELOS Core. Particularly for small primary resources, the distinction between "Resource Model" and "Resource Concretization" may be thin or inexistent. For example, a primary resource like a notepad is created by a developer "as is", and is already ready-to-run, as an instance not a model.

This Life Cycle can also be applied to secondary resources. A secondary resource is either an atomic entity resulting from a primary resource - through a special phase of preparation - or an aggregation from other atomic secondary resources. Prepared resources are obtained by wrapping, filtering, scripting or extracting parts of a primary resource for facilitating direct (possibly remote) use, or potential aggregation. Aggregates are obtained by grouping, integrating or orchestrating some prepared resources. An aggregate may recursively contain other aggregates. Resources obtained in this way are referenced in the Secondary Resource Libraries, also integrated in the TELOS Core.

In all four basic operations of the resource life cycle, semantic referentials, integrated in the TELOS Core, can be used to describe (index, reference) the implied knowledge, the technical properties or the organizational context, or the administrative context of use of any resource. Semantic referentials can take the form of standard or specific metadata, classifications and taxonomies, thesauri or ontologies. This functionality will enable TELOS to operate on the Semantic Web.

3.4 System Generation Cascade Operations

The TELOS system is extended by a four-level cascade from the TELOS Core: grand-parent tools producing parent tools, which are used to compose children tools that, in turn, are used to produce results. The cascade involves four sets of operations: TELOS Core extension, Core use for LKMS composition, LKMS use for LKMA composition, and LKMA use for LKMP production.

In the first step of the cascade, an engineer may extend the TELOS Core, using the *Core Modifier*, a Handler within the Core, to add new resources or resource handlers for example. He can also add components in any library, for example a system module to interact with a new non-TELOS system, a new prepared interface for an existing primary resource (application, document, operation or resource–person), or an extension of a semantic referential. Eventually, this process may be organized as a resource aggregation life cycle as presented in the previous section. Here, the composition phase of the life cycle is the Core extension by the engineer and the use phase is the use of the Core by a technologist

The second generation step is the construction, by a technologist, of a LKMS using the *LKMS Manager*, another Handler Tool within the Core. This operation may be viewed as a composition part in the LKMS life cycle. The LKMS use by a designer is the use phase of the LKMS aggregation life cycle.

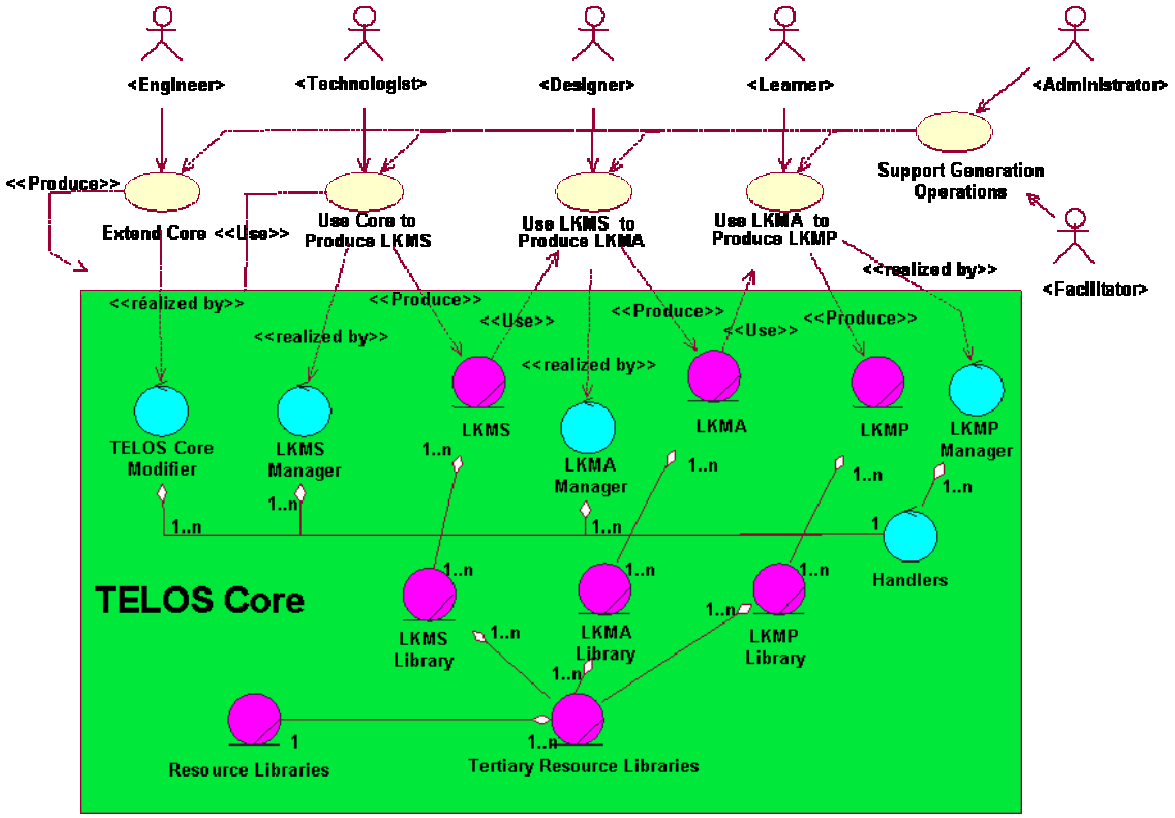

Figure 6 – System Generation Cascade Operations and Corresponding Components

In the third generation step, a designer may use a LKMS instance to produce a LKMA model, for example a course scenario or community of practice environment, using a *LKMA Manager*, another Handler Tool within the Core. The use of the LKMA by learners is the use phase for the LKMA aggregation life cycle.

The final generation step is the use of an LKMA instance to allow knowledge acquisition by a learner. Using a *LKMP Manager*, another Handler within the Core, the learner may produce some traceable results (activity logs, annotations, learner-produced resources, test evaluations etc) usable in the analysis phase. If these learner products are used for other purposes, for example to support a personal portfolio, their creation can become the composition phase for a LKMP life cycle. Any other data that a LKMA may produce internally during its execution (i.e. to preserve its internal states as a system), which are rather private and thus not meaningful for external use, shouldn't belong to a LKMP.

At any of the four system generation steps, facilitators can provide different forms of assistance to engineers, technologists, designers and learners, and also to other facilitators.

Each of the four steps involved in the *System generation cascade* may assume internally its own *Resource Life Cycle*, that is, may have standard sub-phases. For example, an LKMA might be in the phase of composing, management (preparation for use), use (exploration) or analysis.

A shown on table 1, at any of the system generation steps, except the first one where Core administrators manage the core composed by engineers to produce Core instance for the technologists, administrators play two sets of roles. For example, LKMS administrators manage LKMS to produce instances for the designers, but they also analyze the Core to end a LKMS life cycle, providing feedback to the engineer. Table 1 reconciles the resource life cycle roles with the system generation cascade roles.

Cascade Roles: / Life Cycle Roles:	Engineer	Core Administrator	Technologist	LKMS Administrator	Designer	LKMA Administrator	Learner	LKMP Administrator
Composer	Core		LKMS		LKMA		LKMP	
Manager		Core		LKMS		LKMA		LKMP
Explorer			Core		LKMS		LKMA	
Analyser				Core		LKMS		LKMA

Table 1 - Relation between Resource Life Cycle Roles and System Generation Roles

The LKMS, LKMA, LKMP models (classes) and activated instances resulting from the generation cascade process are *Tertiary Resources* that can be referenced and embedded in corresponding *Tertiary Resource Libraries* within the TELOS Core.

3.5 Semantic Referencing of a Resource

The semantic referencing of resources can be involved in any basic, resource life cycle or system cascade operation. Semantic referentials can be built in the TELOS system to represent knowledge and competencies that can be associated with a resource. Competencies are "degrees" of knowledge, for example skills and performance that a user can perform with respect to a given unit of knowledge. They can take many forms: standardized metadata (e.g. IEEE LOM), taxonomies, thesauri, or ontology documents written using the Ontology Web Language (OWL), Resource Description Frameworks (RDF) or topic maps. [Davies et al 2003]

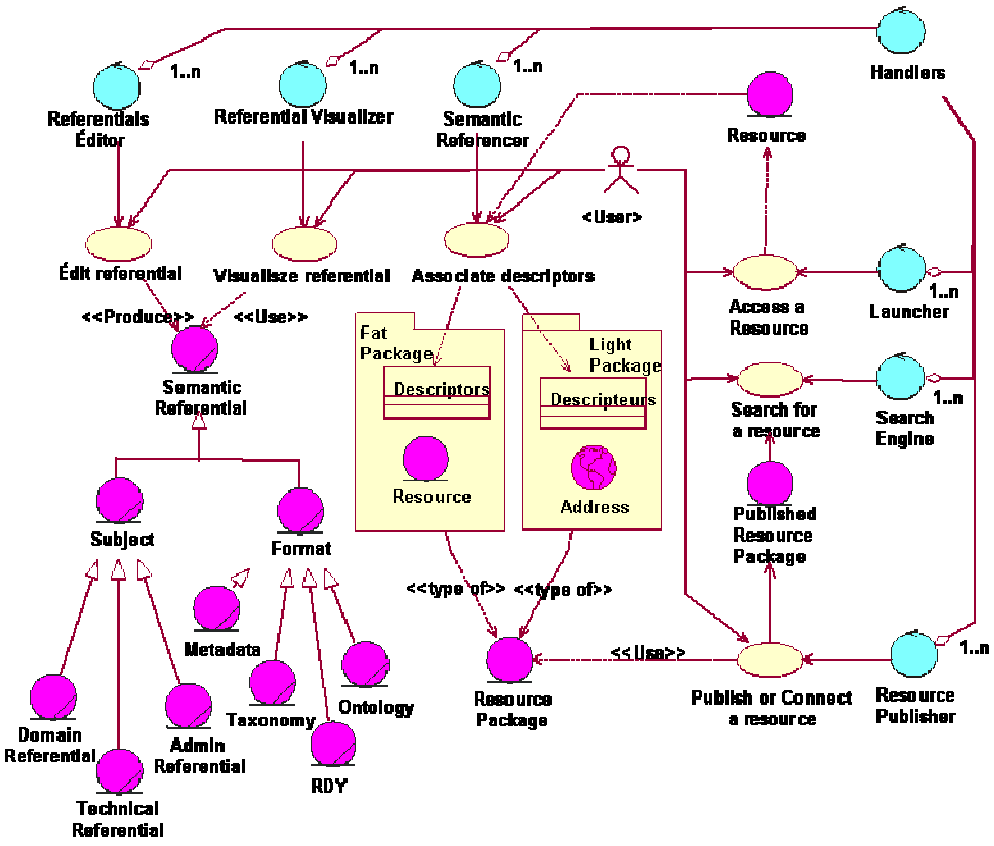

Figure 6- Semantic Operations and Corresponding Components

Three basic handler categories are provided to process semantic referentials: *editors*, *viewers* and *indexers*. Editors help a user construct a semantic referential. Viewers provide access to semantic referentials to help browse the knowledge and competency classes and properties. Finally, referencers provide functionalities to associate semantic descriptors to any kind of resource. Semantic descriptors can be selected in pre-defined semantic referentials by a person using the indexer or by a specialized TELOS resource-tool (software agent) extracting the descriptors from the resource using data mining or other forms of knowledge extraction.

TELOS referenced resources generalize the IMS referenced resource concept used by the SCORM, describes as "content packages". To be shareable, they must be published by a handler called a *publisher*. The publisher places the referenced resource in one or more Internet-accessible resource repositories.

Another handler, a *finder*, provides one or more search methods to find a resource, displaying a list of resource names corresponding to a find request, together with their descriptors.

Then, selecting one of the resources, a *retriever* uses the resource location to perform the necessary operations to provide access to a person or a software agent, display a document or launch the resource. If needed, a special handler in the kernel, the controller, can be called by a retriever to facilitate the interaction with the resource.

4. TELOS Framework Organization

In this section we present the organization of the TELOS framework and the general services it will provide to engineers, technologists, designers, learners and the Core, LKMS, LKMA and LKMP administrators and facilitators, involved in the systems generation cascade.

4.1 TELOS Core and Kernel Structure and Extension

Figure 7 displays the general structure of the TELOS Core presented in previous sections. In this section, we will identify the main services used by the Engineer and by the Core administrator, and their facilitators to extend the TELOS Core. The TELOS Core comprises a Kernel, a Core manager that enables the evolution of the core, seven Core libraries of resources and their corresponding handlers.

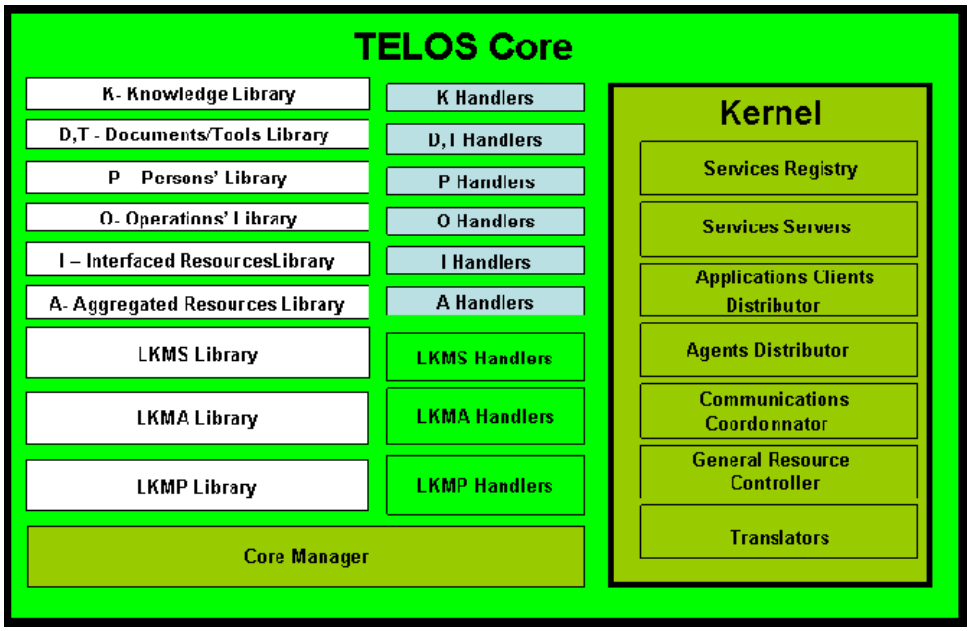

Figure 7 – TELOS Core and Kernel Main Components

The Kernel contains service registries and servers, application client and agents distributor, a communication coordinator, a general resource controller and translators between protocols and standards. The knowledge (K) library contains semantic referentials to describe application domains as well as technical and administrative metadata needed for resource management. The other librairies group resources according to their aggregation level: primary resources (documents and tools, persons or operations), secondary resources (interfaced ou aggregated), and tertiary resources (LKMS, LKMA or LKMP).

Engineering services

The TELOS core engineers may use the services of a special handler, the *core manager*, to modify the core structure, the core handlers or the structure and composition of the core libraries. With it, the engineers can:

- *Modify semantic handlers and semantic referential structures.* Add, suppress or modify core knowledge handlers and library structure: ontology editors, viewers and indexers, conceptual maps editors, viewers and indexers, librarian cataloguing and indexing tools etc.
- *Add, suppress or modify primary, secondary or tertiary resource handlers and resource library structures*. Of particular importance are the addition of modification of assistance or support tools: helpers, advisers, peer matchers etc may be added, respecting the conformity with their context of operation; and also the addition or modification of control tools: New floor control managers, resource sharers, security watchers, privacy verifiers.
- *Modify core structure*. These operations are delicate, involving backward compatibility problems, but add flexibility to TELOS. They include adding a new interfaced resource type to the existent ones; adding a new aggregation type to the existent ones (collection, function, fusion, project etc.); adding a cascade fabrication type to the existent ones for LKMS, LKMA, LKMP; updating a core handler. When a new version of the TELOS core is produced, with some additional core services, it is possible that the TELOS clients have to be also modified accordingly.
- *Modify global core support and control*. The global support and control offered by the core manager can be modified. This might also involve eventual kernel modifications.
- *Note, trace, receive support in their engineering operations.* As for any composition activity, these services can eventually be operated in a collaborative way.
- *Save, suspend, resume Core modification.* Performs these operations involved in any composition activity, operated in multiple sessions, determining the evolution of the TELOS system.

Core administrator services

A core administrator manages the content of some primary, secondary or tertiary library. He may act on demand of another actor (technologist, designer or learner), interact with this actor to perform or delegate to him the right to add, update or eliminate a semantic or a primary, secondary or tertiary resource.

Core facilitators services.

A core facilitator supports the Engineer or the Core administrator in their task. Here, most of the time, facilitation services will be provided by the on-line TELOS technical documentation or by another senior engineer or administrator.

4.2 Core Use for LKMS Construction

Figure 8 presents the general structure of a LKMS embedded in the TELOS Core or external to it. In this section, we identify the main services used by the Technologist (LKMS composer), by the LKMS administrator, and by the LKMS facilitators to both of them.

The technologist constructs an LKMS model by extracting resources from the core libraries and handlers

to be included in the LKMS. To achieve this, he will use a LKMS handler (editor) provided in the Core. Such a handler may use a combination of aggregation principles (collections, integration and orchestration). This variable geometry is the main purpose of constructing LKMSs instead of having only one platform or LCMS. Initially, the TELOS core may be equipped with a unique LKMS handler and therefore produce LKMS with similar structures. More diversity can be added by an Engineer, adding new LKMS handlers in the TELOS Core.

When the construction process is considered finished, the LKMS model is handed to the LKMS administrator to start the LKMS instance preparation phase. The administrator may begin the adaptation to the context of LKMS use. This phase may consist of a chain of sessions producing finally an LKMS active instance like the ones shown on figure 8. The administrator can place the LKMS to work in the core LKMS library (embedded LKMS) or enable it to function external to the Core, in an interfaced, linked and even autonomous way. In the last case, all the necessary handlers and part of the kernel will have to be included in the external LKMS.

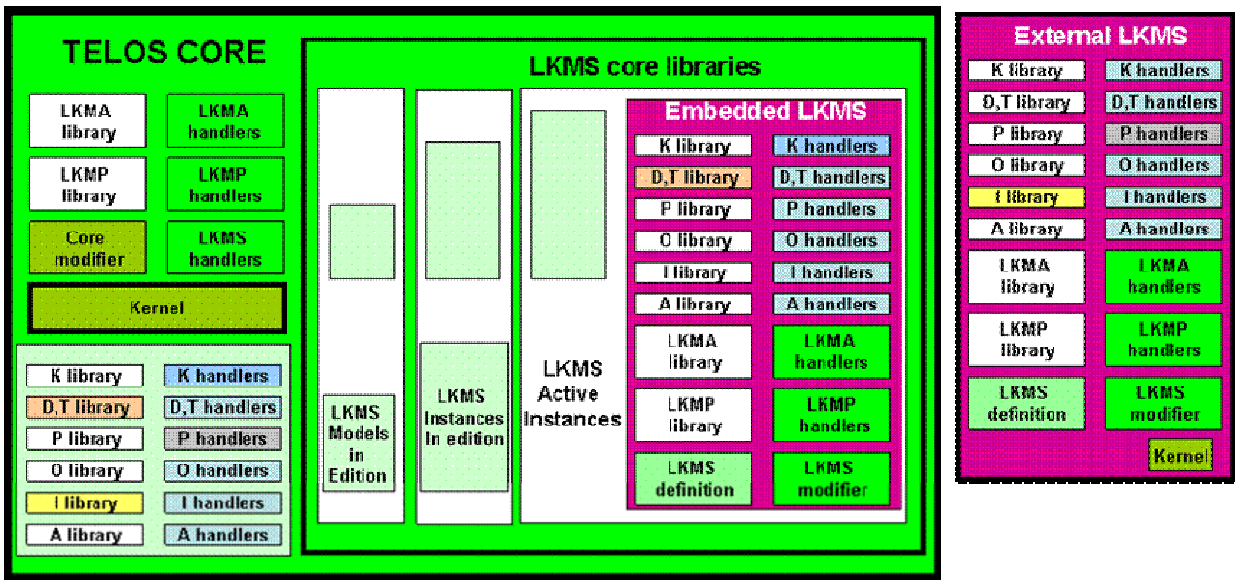

Figure 8 – LKMS General Aggregatin Structure and Positions

Technologist Services

The LKMS composition by the technologist can be seen as an aggregate composition. Therefore he can access composition services such as finding components, making notes, producing traces, receiving help etc. Here we will concentrate on the more specific aspects. With an LKMS editor, the technologist can:

- *Compose the LKMS knowledge referential.* If the global Core referentials are not available or pertinent, a LKMS specific K referential may be necessary using a knowledge editor.
- *Organize the LKMS structure* using an aggregation principle that will be selected and choose the LKMS future destination (embedded, or external: interfaced, linked or autonomous)
- *Organize LKMS parts* (adding resources libraries and handlers to the LKMS model): knowledge handlers and documents (ontologies, metadata, etc.); primary resources (documents, tools, persons and operations) and their handlers; interfaced or aggregated resources and their handlers (editors, publishers, interpreters, structure editors, resource binders, support and control editors explorers, generators, data viewers etc.; LKMA and LKMP parts, libraries and handlers, or a component placed in the LKMA library
- *Prepare the LKMS modifier* for later adaptation.

- *Organize support and control* parts of the global, depending on the destination (embedded, linked, autonomous)
- *Note, produce traces and receive support* in his construction activity.
- *Save, leave, suspend, resume LKMS construction*. A LKMS in construction may be saved in the LKMS core library. As for any other aggregate composition, the LKMS construction may be done in several sessions, the technologist suspending or resuming the composition cascade. If the composition is cooperative, we may speak about leaving a session continued by another technologist.

LKMS Administrator services

The composition and the exploration of an LKMS will vary depending on the complexity of its structure. A "thin" LKMS will have a simpler structure but its execution will involve remote invocation of handlers and resources. A fat LKMS will possess its own libraries but still need to communicate with some core handlers. The heavy LKMS may be a cumbersome entity (with resource replication problems) but it will function autonomously. These are the main choices a LKMS administrator must make when preparing an LKMS instance for publication. We now present the main services available to him.

- *Chose a saved LKMS model and create an instance*. The LKMS models are re-usable structural templates with a certain degree of flexibility in the template instantiation. The LKMS administrator begins by choosing a LKMS model, usable as an instance generation support (from the first core library).
- *Prepare the LKMS instance* (extend edition). The administrator may adapt parameterize, concretize, extend the LKMS for insertion in the core or for external activation, deciding on the thin-fat-heavy alternative and the deployment context and finally, installing the instance in its use. This can be done in one or many LKMS instance preparation sessions.
- *Note, produce traces and receive support.* As any other meta-managed activity.
- *Save, leave, suspend, resume*. The technologist's work may be saved in the second LKMS core library that contains LKMS instances in preparation.
- *Activate the LKMS*. The LKMS instance is activated to work in the chosen contexts: embedded in the Core or published for external linked or autonomous use. It then passes to the third LKMS core library that contains LKMS active instances.
- *Core use analysis and feed-back.* The LKMS administrator is, at the same time a Core analyst of the Core life cycle process. He may analyze the core use to provide some feed-back propositions to engineers.

LKMS Facilitator Services

LKMS construction may involve technologists facilitators, working synchronously or asynchronously to observe, guide, evaluate and replace the LKMS administrator in some of his operations, eventually by supporting also the LKMS instantiation workflow. If the LKMS composition uses a LKMS composition workflow aggregate (for example, a meta-function), the support can be treated as an execution phase of that workflow.

4.3 LKMS Use and LKMA Construction

Figure 9 presents the general structure and four positions of a LKMA, embedded in the TELOS Core LKMS or LKMA library, embedded in an external LKMS or totally external to any LKMS.

Theses LKMAs are composed by designers (for example an educational author) using LKMA handlers (editors) and some "raw material" (secondary resources) placed in an embedded or external LKMS. They

may also use directly a LKMA handler and resources placed in the TELOS Core. Designers compose a LKMA model that can be placed (and modified) in the first section of the Core LKMA library, or within the LKMA library of the constructing LKMS (embedded or external).

The LKMA model is a template that may be reproduced, with some adaptations and concretizations by a LKMA administrator to produce LKMA instances. After a certain number of operations, in a single or a chain of preparation session, this administrator produces a final LKMA instance that can be activated in the core LKMA library, in the LKMA library of an embedded or external constructor LKMS, or as an autonomous LKMA.

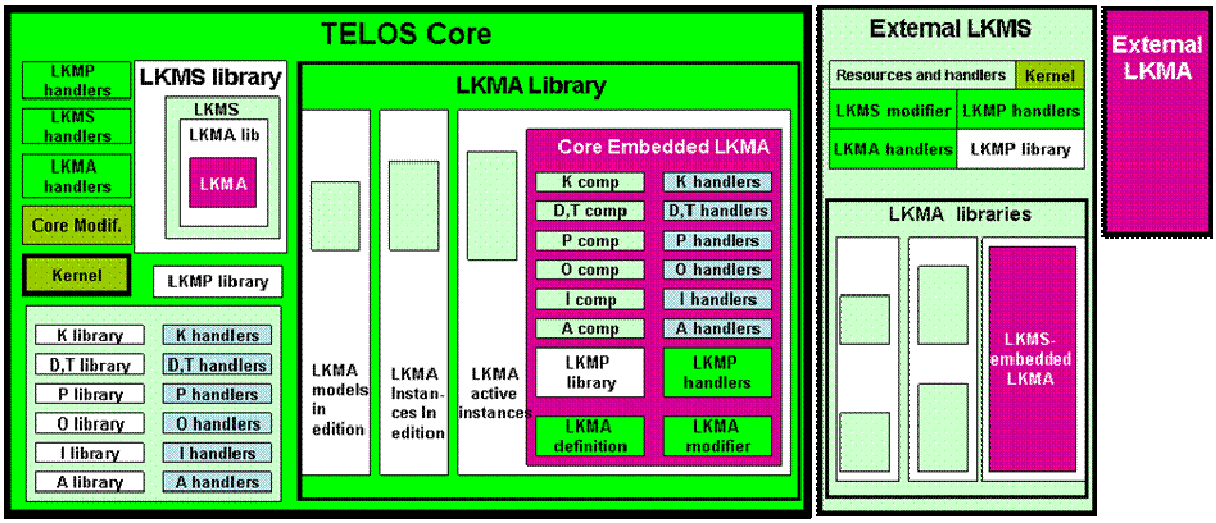

Figure 9 – LKMA General Aggregation Structure and Positions

Designer Services

- *Organize K referential layer*. Generally, the semantic referential used in a LKMA already exists in a LKMS or in the core and can be retrieved from there. However, sometimes, the designer will need to built a local semantic referential in the form of a domain ontology, conceptual maps, thesaurus, catalogues, indexing keys, etc) or a local "add-on" document, completing (adapting) a knowledge document. The most interesting case is when the LKMA is used by learners and their facilitators for an emergent modification of a semantic model.
- *Define structure*. The central part of any LKMA is its aggregate internal organization structure (in the thin case, it may be the only part). It may consist of only one aggregation layer (conforming to the collection, fusion, project, function or other aggregation types), or in a recursive cascade of aggregate definitions. This process is also called “scenario building” or “learning desing”.
- *Organize an LKMA content package.* Add, suppress or update the aggregated resources, sometimes with their handlers, and their semantic descriptors.
- *Organize actor support and the control layer*, depending on his activation type, its position
- *Note, produce traces.* The LKMA designer may annotate his activity. This data is not included in the final aggregate, but is observable in the same way as other composition sessions. The LKMA editor used in the composition process may also intercept and save some composition activity traces.
- *Cooperate and receive support*. The LKMA composition activity (observed directly or by traces) may be collaborative and may be supported by the system (for example using design metafunctions) or by specialized facilitators (for example by technologists having produced the producing LKMS).

- *Save, leave, suspend, resume.* As for any other aggregate composition, the LKMA construction may be done in several sessions, the designer suspending or resuming the composition cascade. If the composition is collaborative, the designer may leave a session and the composition can be continued by another designer.
- *Publish template.* When the construction process is finished the LKMA is published in the second section of a LKMA library, either within the core, or in the embedded or external producing LKMS used by the designer.

LKMA Administrator Services

When the LKMA model has been published for instantiation, it can move to the second section of a LKMA library, for an administrator to start the instantiation process. This can be done in one or many LKMA instance preparation sessions. As with other processes, collaborative LKMA instance preparation is possible, eventually driven by a LKMA preparation meta-aggregate or workflow composed previously.

- *Chose a published LKMA model to create a new instance.* The LKMA administrator begins by choosing a LKMA model usable as an instance generation support. He declares a new LKMA instance, based on the chosen model to the second section of the LKMA library where it may be edited.
- *Prepare the LKMA instance* (extend edition). The administrator adapts the LKMA for core or external activation (parameterization, concretization, eventually a composition extension), decides on the deployment context: embedded thin/fat/heavy or external interfaced/linked/autonomous available possibilities. He then installs the instance in its future use context in the third section of the LKMA library. Thid concretization process may be distributed between the composing, managing and using phase of the LKMA, applying various organization strategies or "life cycle modes".
- *Note, produce traces and receive support.* As any other meta-managed activity.
- *Save, leave, suspend, resume*. As in any other management activity.
- *Activate the LKMA.* When an instantiation process is finished, the manager "closes" the instance editing chain and activates the instance for execution. The LKMA instance is integrated in the third section of the LKMA library (core, LKMS embedded, or LKMS external) or placed in an external context for LKMS-free use. In that last case, and for some types of LKMA aggregates, this step may imply a LKMA "compilation", producing an executable LKMA.
- *LKMS use analyze and feed-back*. The LKMA administrator is, in at the same time a LKMS use analyst. He may analyze the LKMS use by designers for producing and managing LKMA and make some feed-back propositions to technologists

Designer and LKMA Facilitators Services

The designer or the LKMA administrator may need synchronous or asynchronous support from technologists having composed the LKMS they use. Another possibility is to embed the LKMA life cycle process in a design workflow or meta-functions. In that case, the meta-function acts as a system facilitator observing, guiding, evaluating and doing some operations for the LKMA designer or administrator, eventually supporting collaboration between teams. An example of this is given by the ADISA learning design support system [Paquete, Rosca et al, 2001]

4.4 LKMA Use and LKMP Construction

Figure 10 presents the general structure and eight positions of a LKMP composed using a LKMA. The learners and the facilitators that participate in LKMA instance "sessions" are the end users that will produce results that compose these LKMPs. Some LKMA results such as annotations, learner documents,

evaluations etc., may be edited directly by the participants, while others, for example traces, may be obtained by TELOS agents placed in an autonomous LKMA, in the sustaining LKMS, in the sustaining core or in the kernel. These results are normally placed in the data layer of the executing LKMA instance. They can later on be selected for inclusion in LKMP libraries as learning or knowledge management products. They can also help compose user or group ePortfolios.

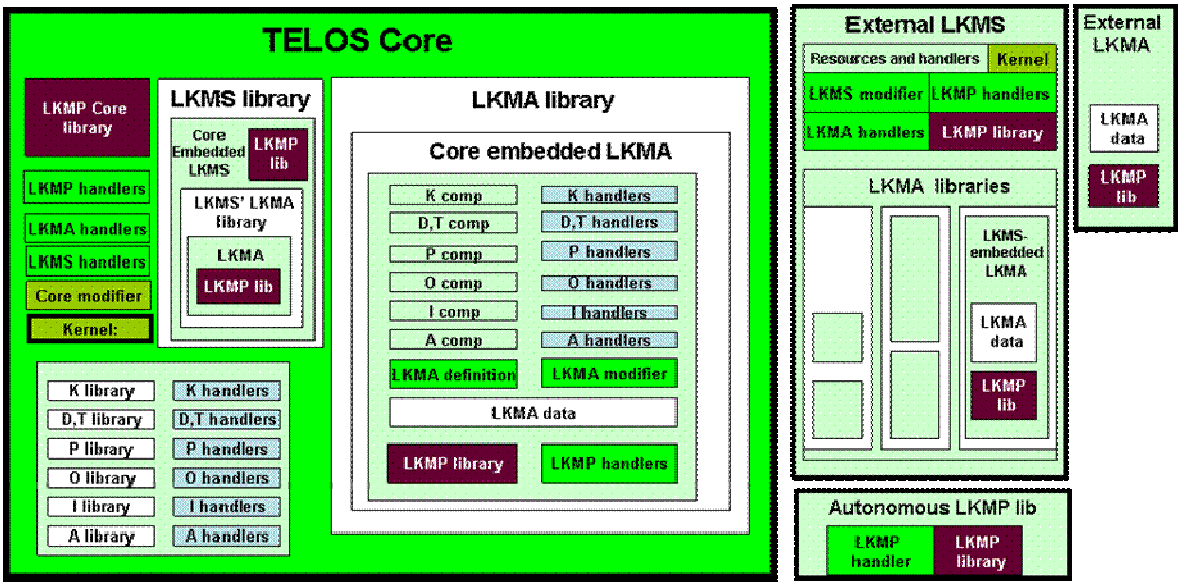

Figure 10 –LKMP General Aggregation Structure and Positions

Learner Services

- *Find and access a LKMA instance*. The learner begins a session by accessing an active LKMA instance, placed in the core LKMA library, in the LKMA library of a core LKMS, in the LKMA library of an external LKMS or in an external autonomous position. If he resumes the use of an interrupted instance exploration, he will obtain the corresponding LKMA saved data. A LKMA instance participant can be added to a collaborative session already opened by another participant.
- *Explore an LKMA instance.* The LKMA exploration depends on its aggregation structure and the collaborative facilities provided by LKMA handlers. In collection aggregates, it may consist in choosing and using resources grouped in collection. In fusion aggregates, the structural relations will constrain the learner freedom in using the components but will also reduce the "lost in space" effect. In project aggregates, every learner may dispose of special interfaces or environments, conforming to his roles in the project. In function aggregates, the learner will be guided by the model of the operational flow, for example a learning design structure.
- *Collaborate with session partners*. If the LKMA provides synchronous and/or asynchronous collaboration, every participant may be helped in coordinating his activities with his partners, whether they act in the same, previous or later instance sessions. Sometimes the partners are working on different operations (cooperation), sometimes they make a parallel approach to the same operation (collaboration), and sometimes they are cooperating or collaborating as proposed by the collective activity structure of the aggregate.
- *Receive support.* The learner may use the support integrated in the LKMA or delivered by a supporting session or instance partners.
- *Concretize some instance resources.* For example, the learning design may provide place-holders for run-time learner productions. These "instance delivery time concretizations" may also involve a mechanism for the transmission of produced objects between instance participants.

- *Use component resources and produce new resources* (documents, tools, aggregates, etc). The main purpose of the LKMA is to facilitate the access to its component resources. The facilitation may consist only in the organization of a selective interface allowing the opening of the resource (tool, document, service, user communication). The TELOS resource controllers may also offer resource use services: access negotiations, downloads, installation, decoding, on the fly dependence solving, action interception and logging, scripted events injection, inter-resources parameter propagation, concurrent use solving, QoS. Adaptations, etc. Some LKMA resources (acting as editors) may help the learner to produce results (documents, tools etc) usable later in another operation, or by an analyst observing the resulting data or by a LKMP administrator.
- *Produce intercepted traces.* Some LKMS sustaining an LKMA session may intercept the user actions (depending on the declared ethic policy for this LKMA instance). These "interceptions" may be used by the user observing his own advancement in the same or previous sessions, by a session or instance partner (connected simultaneously or later), by a LKMA data analyst or a LKMP administrator.
- *Mark advancement and annotate.* The learner may declare explicitly his advancement and make related annotations. These "notes" will complete the "interceptions" and the "evaluations" forming the LKMA instance "data".
- *Assess learning.* Learning is the main goal of using a LKMA. Because of its internal intellectual nature, it can only be observed and managed indirectly. The LKMA may provide some tools for the evaluation of the learning results or may make some deductions about the knowledge evolution using "the learning by doing presumption". If the learner makes some production with success, he can be assumed as possessing the associated knowledge or having attained the associated skills or competencies.
- *Add results to a LKMP*. The learner may add some of his personal results to a LKMP, a learning and knowledge management product repository that can be placed in one of eight places (a shown on figure 6-4): the core LKMP library, the LKMP library of an embedded LKMS, the LKMP library of an embedded LKMA, the LKMP library of a LKMA embedded in a core LKMS, the LKMP library of na external LKMS, the LKMP library of a LKMA embedded in an external LKMS, the LKMP library of a external autonomous LKMA, in a completely autonomous position.
- *Manage the LKMP and ePortfolios*. Whatever the LKMP's position, the learner manages its content that can form his personal ePortfolio.

Learner Facilitators Services

- *Observe LKMA use*. Any support activity, pre-defined or launched by a service request from the learner, may begin with the observation of the learner's operations. The observation can cover previous or planned activities by the learner, in the current open sessions or closed sessions of the same instance, or even in other instances of the same LKMA model. This observation depends of the awareness possibilities offered by the LKMA collaborative explorer: perceive the partner presence and actions, see the notes and the traces, access the resources produced by learners, access learner group data.
- *Guide LKMA use*. The facilitators use the communication possibilities of the collaborative LKMA explorer to guide the learner in his activities, using synchronous and asynchronous, written, oral or video messages.
- *Execute some support operations.* Some support from the facilitator may consist in actions: executing some operations, adapting some parameters, preparing some conditions etc.
- *Make annotations and evaluations*. The facilitator may add annotations to the LKMS exploration

data about the learner (and eventually his own) activities. He also may evaluate the learner competence or modify some data about the learner's knowledge and competencies, according to his mandate. He may compose some evaluation result documents. These elements may be placed in the LKMA data layer and added later to some LKMP.

Learners and Facilitators Services

- *Open instance execution.* The learner or the facilitator that starts the first session of an active LKMA instance, opens this instance execution chain.
- *Close instance execution.* The learner or the facilitator that considers that the LKMA instance execution chain is completed, declares the closure of the instance. Only the view facilities remain available, as a natural way for observing instance results.
- *Suspend the active session of an instance.* A learner or a facilitator may suspend an LKMA instance session, if they are the only active users and they decide to quit momentarily the current exploration.
- *Resume a suspended instance in a new session*. If a learner or a facilitator wants to continue the exploration chain of a suspended LKMA instance he must resume the instance by opening a new session.
- *Join an active session*. A user learner or a facilitator may join a collaborative exploration of an already active LKMA session currently used by other users.
- *Leave an active session.* A user may leave a collaborative exploration of an active session that is continued by other active users.

LKMP Administrator Services

- *Manage LKMP*. The LKMP administrator may change the content of a LKMP, depending on the privacy policy.
- *Correlate LKMP and resources libraries.* Some products may be promoted from a LKMP to a TELOS resource library by the LKMP administrator. In this case, the LKMP products may be placed in the shared library of the producing LKMA, wherever its deployment position is; they can also be replaced in the LKMP library with pointers to these actual resources.
- *Correlate LKMP and user accounts.* Some LKMP products, for example LKMA instance exploration data, new knowledge references to the user, or new documents produced by the user sustain an update of the user accounts.
- *Analyze LKMA data.* The data (traces, annotations, documents, products) resulting from a LKMA instance exploration (closed or not) may be accessed for analysis by a LKMP administrator. It is possible that the data analyst uses the data view facilities of the LKMA explorer, working on a closed instance. Some other data analysis tools may be used.
- *Produce observations and propositions.* The LKMP administrator, acting as an analyst, may produce documents reflecting his recommendations to LKMA designers or users: reports, statistics, LKMA or LKMS change requests etc

4.5 Summary of TELOS Services

Figure 11 present a summary of the TELOS services that we have described in the previous sections as TELOS Handlers. They are grouped into five categories of services: *Kernel Communication services*, *Semantic services* and three groups of resource management services: Secondary *Resource interfacing services, Secondary resource aggregation services* and *tertiary resource management services*.

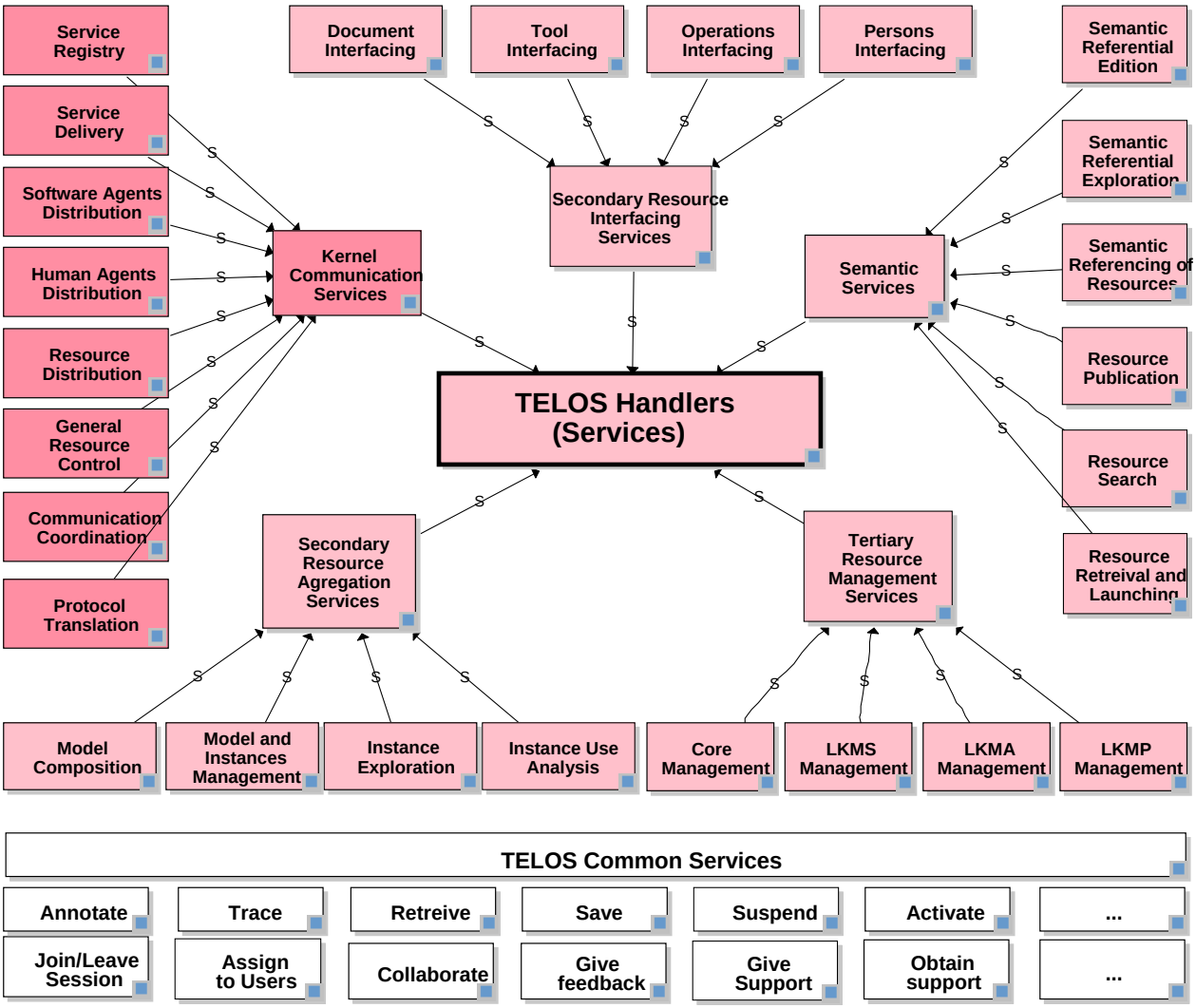

Figure 11 – TELOS Handlers (Services) Taxonomy

To this TELOS system services layer, we can add the infrastructure or common services mentioned in the previous sections. These will in general be provided by the operating system but might require some adaptation for their different levels of use within the higher lever handler services.

5. Conclusion

In conclusion, we underline what constitutes the strength and the originality of the proposed framework, together with some questions that need to be answered.

TELOS is based on aggregation models that fully integrate users with the resources they use, especially in the central pedagogical process from model composition, to instantiation, to exploration and to analysis/feedback. TELOS manages this resource life cycle at all levels and in all its phases in an integrated way. But to repeat resource life cycles recursively is not enough to define the global process, so TELOS adds another level for resource generation where “grand-parents” systems give birth to parents,

which themselves give birth to children systems or environments. Again, TELOS managers this system generation cascade in a full integrated way.

In this global process, workflow (or function) generation is of particular importance and difficulty. Function are workflows inspectable, interactive and collaborative multi-actor interfaces. The users of an instance of a function model can use the function as an interface to inform themselves, to declare or be evaluated on their progress, to obtain assistance, to access operation and coordinate with others, to find and use material and human resources.

The knowledge/action cycle is at the center of our preoccupations. This is why TELOS combines emergent modes of operation where users organise and define their own operations (of function components) and orchestrated modes where they use pre-defined functions. A cybernetic loop closes when the action in a process execution are captured to produce a workflow model that can be reused. For example, a designer can use this feature to register a possible workflow in a course, and then offer it as a learning design.

With regards to knowledge representations, we have taken a pragmatic approach, looking for flexible and usable solutions. We reject one extreme not to include any representation because such a resource management system would support learning poorly. We also reject the other extreme where complex representations, theoretically more satisfactory, would exceed the capacity of most persons to use them. In many applications, simple metadata or lightweight ontologies can be used with immediate benefit, for example pairing peer users or users with appropriate resources, while at the same time supporting a facilitator to guide its interventions towards other users. We are looking for such benefits that will repay the effort made to integrate knowledge representation into the environments' design.

Our work on TELOS is continuing on two levels : the further definition of the technical architecture of the system and the reconstruction of our actual Explor@-2 system [Paquette 2001; Paquette et al 2004] to bring it closer to the technical architecture proposed here. Both streams of research will be synchronized periodically as the research process unfolds. Meanwhile, other LORNET research teams work at the development of a variety of components that are taken in account in the TELOS architecture, such as new resource interoperability services, learning design and function management editors and players, adaptive learning objects edition and integration, knowledge extraction from resources and advanced multimedia components.

Références

[Berners-Lee et al. 2001] Berners-Lee, T., Hendler J., Lassila.O. *The Semantic Web.* Scientific American, May 2001, Feature article.

[Blinco et al. 2004] Blinco, K. Mason, J. McLean, N. Wilson, S. *Trends and Issues in E-Learning Infrastructure Development.* White Paper presented at alt-i-lab 2004.

[CETIS-JISC 2004] *eLearning Framework Documents.* http://www.elframework.org/

[Davenport & Prusak 1998] Davenport T. H. and Prusak L. *Working Knowledge: How Organizations Manage What They Know*. Harvard Business School Press, 1998.

[Davies et al 2003] Davies, J. Fensel, D, Van Harmelen, F. *Towards the Semantic Web, Ontology-Driven Knowledge Management*, Wiley, 288 pages

[IMS-AF 2003] *IMS Abstract Framework* http:// http://www.imsglobal.org/af/index.cfm

[OKI 2003] *OKI Architecture Overview.* Open Knowledge Initiative, http://web.mit.edu/oki/product/whtpapers/arch_overview.html

[Paquette 1995] G. Paquette. *Modeling the Virtual Campus*. in "Innovating Adult Learning with Innovative Technologies (B. Collis and G. Davies Eds) Elsevier Science B.V., Amsterdam, 1995.

[Paquette 2001] G. Paquette. *Designing Virtual Learning Centers*. In H. Adelsberger, B. Collis, J. Pawlowski (Eds) Handbook on Information Technologies for Education & Training within the Springer-Verlag series "International Handbook on Information Systems", pp. 249-272

[Paquette and Rosca 2003] Paquette G. and Rosca I. *Modeling the Delivery Physiology of Distributed Learning Systems*. Technology, Instruction, cognition and Learning, , vol 1-2- Spring 2003, pp. 183-209.

[Paquette et al 2004] Paquette G., Lundgren-Cayrol K., Miara, A. and Guérette L. *The Explor@2 Learning Object Manager*, in R. McGreal (ed), Online education using learning objects, pp 254-268. London: Routledge/Falmer.

[Paquette, Rosca et al 2001] Paquette,G., Rosca, I. , De la Teja, I., Léonard, M., and Lundgren-Cayrol, K. *Web-based Support for the Instructional Engineering of E-Learning Systems*, WebNet'01 Conference, Orlando

[Paquette, Rosca et al. 2005]. Paquette G., Rosca I., Mihaila S., Masmoudi A. *TELOS Conceptual Framework, version 0.8*. LORNET Technical Documents.

[Rosca 2005] Rosca, I. *TELOS Conceptual Architecture, version 0.5*. LORNET Technical Documents

[Sveiby 2001] Sveiby, K.E. *What is Knowledge Management?*
http://www.sveiby.konverge.com/articles/KnowledgeManagement.html

[Wenger 1998] Wenger, E. *Communities of Practice. Learning, Meaning and Identity. Learning in Doing: Social, Cognitive, and Computational Perspectives*. Cambridge University Press : Cambridge, Mass. 1998

[Wiley 2002] Wiley D.A. *Connecting learning objects to Instructional design theory: a definition, a methaphor, and a taxonomy*. In Wiley (Ed) The Instructional Use of Learning Objects. Agency for Instructional Technology and Association for Educational Communications of Technology, Bloomington, Indiana, 2002, 281 pages.

[Wilson et al. 2004] Wilson, S. Blinco, K. Rehak, D. *Service-Oriented Frameworks: Modelling the Infrastructure for the next generation of e-Learning Systems*. White Paper presented at alt-i-lab 2004.