TELOS - an infrastructure for educative inter-operation through the Internet

Ioan Rosca[1]

[1] conceptual architect, Montreal Teleuniversity
ioan.rosca@licef.teluq.uqam.ca; ioan_rosca@yahoo.com
http://www.ioanrosca.com/education/

Abstract. The LORNET project seeks technical and semantic inter-operation between educational service sources and resources repositories, accessible through Internet. I have defined the TELOS' conceptual architecture so that it sustains the modeling and the management of distributed instruction activities: from the emergent ones (searching human and material support and chining operations freely) to the orchestrated ones (through rigid or adaptable scenarios). The proposed system is based on a "knowledge layer" used for referencing all the components: persons, documents and activities. It trails the lifecycle of the new resources, obtained by "aggregating" the existent ones and the lifecycles' chaining- in production cascades. Apart the study of other pedagogical workflow managers I have tried new physiological formulas for procedures' coordination, using "functions"- managed with the GEFO prototype. These interesting utilities (progressive adaptation, semantic matching etc.) can also be used in the modeling, demonstration, management and evolving engineering of the TELOS system.

1 The LORNET project

I have synthesized my preoccupations for the modeling and management of educative processes in my PhD thesis [11]. It was also then that I chose the principles that have since oriented my research [5]. I have extended these investigations participating in some big projects [2] and managing prototypes' development. For example, I had to conceive an integration plan for the various systems, components and ideas developed by LICEF: EXPLORA (virtual campus management platform), MOT (knowledge structure, pedagogical scenario and resource conception/diffusion editor), ION (distributed resource controller and aggregator), VAL (cooperative pedagogical workflow manager) etc. I also studied the inter-operation between LICEF's and other's systems (AdapWeb, Sigal, Nomino etc), the transition of the EXPLORA platform towards a service provider position (the SavoirNet project) the transition of MOT to IMS-LD compatible tools and the inter-operation of pedagogical resource repositories using metadata records (the EDUSOURCE project).

At the closure conference for the projects dedicated to the stimulation of Internet usage in Canadian education, financed in the Tele-Learning network [10] I proposed a new cycle, dedicated to the coordination of efforts, in order to insure inter-operation between the instruction and knowledge management systems, reducing unpleasant and costly redundancies. The tendency towards "Web Services" had opened interesting possibilities but new problems were surging: access management and negotiation, confidentiality, intellectual rights, concurrent interests, operational and development autonomy of the connected systems etc. A "distributed pedagogical operating system's" architecture had to be defined (protocols, base services, strategies that could progressively lead to the inter-operability of the entities bound to the system and respecting its norms). It was a major challenge. I was invited to materialize my ideas, as conceptual architect of TELOS (tele-learning operating system), in the LORNET (learning object repository network) project (approved in 2002, launched in 2003 and during until 2008). I have done this between 2003 and 2005. Here are some extracts from the strategic principles, witch I inserted in the vision document [8]:

"Solving Real Learning and Knowledge Management Problems. [] to provide solutions not only in terms of system's tools, but also in terms of processes [] to use them effectively in real contexts. [] the driving force will be the careful definition of use cases []
Reusing and Integrating Existing and New Tools [] to integrate technologies from different fields and to develop new ones when they are educationally significant. We will reuse, as much as we can, existing editors, communication tools, interoperability protocols []
Concentrate on Essential Developments - Reduce risks [] shifting the accent from tool development to careful analysis, evaluation and well-planned specification.[]
Flexible and Pragmatic Approach. [] accommodate a variety of situations, from planned instruction, to more or less structured self-training, emerging communities of practice or performance support systems integrated with work environments. [] the emphasis will be on the relationship between a model and the phenomena assisted, supported or orchestrated by it.
A Society of Human and Computer Agents. [] a view where humans and computer agents are interacting parts of a unique system. [] build or use computer tools only when they are really useful. Sometimes, organizational adaptations, documentation support or human communication may be more appropriate (and less costly) []
Build Technology-Independent Models. [] protect the conceptual models (as intellectual capital of the LORNET research community) from devaluation by technological instability [] These conceptual models are not just prerequisites to the TELOS system's development; they are part of the system, maybe its fundamental layer.
Observing, Planning and Supporting Learning Ecosystems. [] tools to model the complex processes involved in a distributed learning system: before the process (to design), during it (to support users and observe their behavior) and after it (to understand, evaluate and react). They will enable the users to get involved efficiently in pre-planned as well as emerging events.[]
Structural modularization and evolutional segmentation. [] The architecture will promote "horizontal" (structural) modularity (between components) and "vertical" (evolutional) segmentation (layers for various stages: specification, architectural model, etc) . []
Reusable and Interchangeable Models and Components [] alternative tools, classified in classes by their functionalities. [] Even at the "kernel" level, the general functions could be covered by one or more alternative modules, accessible on a distributed "services bus" []
An Assembly and Coordination System [] TELOS will not be another huge distributed learning platform or a system to generate rigid platforms [] will essentially be a coordination and synchronization set of functionalities for the interaction of persons and computerized resources that together constitute a learning or knowledge management system."

Starting from these principles, I've elaborated the TELOS conceptual behavior and architecture ([5], [12]/TELOS), that I will succinctly present in the next chapter.

2 System's behavior and architecture

Apart the orientation towards the facilitation of technical and semantic inter-operation - already mentioned - the main ideas are: the indexation of all elements on a common knowledge and competences reference system; the observation of the resources' lifecycles and of their recursive or cascaded aggregation; the modeling and facilitating of emergent, planned or adaptable processes; the use of procedural aggregates (operations and functions) for orchestrating man-machine ensembles; the analysis of the "competence equilibrium" around operations for launching support and optimization services (selection, matching, alerting etc); the management of the concretization "modes" and of system processes - with meta-functions.

2.1 TELOS Kernel and communication bus

The "structure" of the systems produced with the TELOS tools will be designed by their conceivers or will eventually be established dynamically, according to run-time needs. That is why the connection between TELOS modules must be assured by an internal "communication bus", coordinated by a "kernel" that deploys and connects the communication interfaces. All the core modules must be "plugable" to this bus, using a TELOS inter-communication protocol (working above the network layer protocols). Remote service requests will also be resolved by resorting to this inter-operation layer.

When a user U wishes to act on a resource R that he does not dispose of, he must take necessary actions in order to obtain it and install it in his working context or to obtain an interface through which he can ask remotely for its services. TELOS will put at his disposal a "user-agent" witch will help him in these actions and through which, eventually, even after putting them in contact, the user-resource dialogue will be intermediated. Another situation appears when the user operates on a system S witch, in order to serve him, takes "background" steps to obtain a "system-agent", through which it can obtain from R the services necessary to satisfy U's requirements.

The TELOS' kernel will support the distribution of the resources (or of their remote access interfaces) and of the user (or system) contact and communication agents. It will also facilitate the coordination between them (the chaining of the declare/ ask/ deliver/ receive phases- in order to satisfy remote micro-service requests). The beneficiaries or the providers of these services will be various categories of users (TELOS or external) and systems (TELOS or external). The kernel will also contain a general resource controller (delegating the control of any resource to the appropriate handler) and an import-export module- opening TELOS for communication with systems based on other norms.

To request and receive these services, the systems and participants from within the TELOS space can also use the interfaces and agents installed at their construction, while the soliciting external systems and users must first obtain an agent (client interface), distributed by the kernel.

2.2 The knowledge management (detailed in [5] and [6])

Knowledge representation. Based on the language's natural reference system, knowledge domains (spaces) can be built, according to various organization norms. These systems, establishing relationships between the representations, model (declare) relationships between the represented knowledge (and between the realities that they reflect). Therefore they can enrich (explicit) the "meaning" of knowledge and can be used as reference systems.

Reference system's organization norms. The various forms of organization and, consequently, of the indexing and retrieval processes (classification, relational structures, dictionaries, hypertexts, declarative languages, graphs etc) have all their qualities. The best potential of automatic inference (assistance) is obtained when the reference system is organized according to a "computer-comprehensible" logic - hence the interest for ontologies.

Decomposition and aggregation. Knowledge can be detailed by decomposition, as a notional sub-space, organized conforming to a norm and usable as a reference system for its sub-knowledge. This decomposition process can continue in cascade. In the opposite way, aggregation can take place, linking (merging, fusing) notional spaces.

Knowledge engineering. Confronts difficult problems such as: evolution of reference systems (refining, correcting, versioning) and recalculation of already operated references [3]; translations imposed by the collaboration between entities using reference systems with different organization norms, semantic inter-operation languages, accommodation of multilingualism, etc.

2.3 The management of participants and of their competences

Participants. Lucrative or instructive processes (emergent or planned) can involve, as actors, operating in various postures, participants of several types, selected from those registered in the corresponding repertories: *persons*, *groups of persons* (enumerative collections), *categories of persons* (defined by common characteristics), *teams* (aggregations implying roles and protocols) and *agents* (human or artificial, acting for others). The pertinent choice (intervention) of a person requires the prior declaration- in the record's fields of every potential participant- of information witch can influence the decision (negotiation) of its implication (competences, interests, availabilities and requirements, communicational, linguistic and technical particularities/preferences).

Competences. The "knowledge references" that we encounter in expertise declaration are, sometimes, equivocal (they signal that a person detains them or aims at obtaining them?; that a support person presumes them or can explain them to the envisioned public?) hiding a binary approach : knows / does not know. This reductionism eludes the gradual character of "learning". In a support (instruction) system, it is exactly the evolution of the subjects' understanding and the contributions to this evolution that must be observed. We need qualitative and quantitative descriptions of someone's position relative to knowledge: "*competences*" - managed conforming to a norm C.

Mastering level, abilities and postures. Similar to the executions' evaluation for exterior operations (for person-object relationships as "utilize", "produce", "modify"), the cognitive operations (relationships in the person/knowledge pairs space) can be evaluated by a "mastering level"- measured on a scale M. Another organization [1] proposes "abilities" *(knowledge/ comprehension/ application / analysis/ synthesis / evaluation),* witch can be treated as separate person-knowledge relationships (sustaining fine-grained qualitative inferences) or as a universal competence scale for the "to know" relationship. In order to observe the competence equilibrium around pedagogical operations, I have organized [5] the characterization of the P participant's competences on "postures": (*knowK, aimK, explainK(x,y), describeK(x,y), evaluateK(x,y), recommendK(x,y)*)- where the parenthesis show a predicate depending on the detained (x) or aimed (y) "mastering level" of the person (learner etc) to which P could explain (describe in a document, evaluate, recommend) the knowledge k.

Competence management. Once the Mi scales chosen, we can express, based on them, the "mastership" for each ability or posture, obtaining vector type characterizations of the competence P, referring to k. But the task to calculate the global competence starting from this decomposition or using the competences for a knowledge decomposing of K… is not obvious. The organization of the competence reference systems poses additional problems (as the negotiation of concurrent evaluations - according to an authority protocol). All this difficulties led to simplifications as the (discussible) use of uniform evaluation scales like: 0-1, 0-10, 0-100, A-F - instead of a fine-grained competence management.

Secondary "participant resource". The persons that will intervene as protagonists (users) or as support actors - are represented in the system by their records - used by the retrieval procedures. A participant can be accompanied by a complex object (knowledge model, preferences and specialized access interfaces, communication channels, agents- mandated by him or for him). This "secondary resource" "wraps (interface) the participant" and represents him in the system.

Actor concretization. In some aggregate models (teams, operations, functions), generic actors are initially stipulated, going to be concretized before the model's execution or even at run time. A participant's selection relies on the comparison between the profile declared in his record (mandate, preferences, competences) and that of the actor (role) specified in the generic model. This evaluation can be supported by the secondary resources that represent the participants and by the agents watching the competence equilibrium.

2.4 Resources' management

Material and documentary resources. Involved as support tools or activities' products - they can be put in collective and personal repositories (portfolios). They are shared, respecting administrative protocols (negotiation of the acquirement, access and use) and technical norms (inter-operability, adaptation to use conditions). Some are treated as objects without informational (explicative) connotation, others have documentary dimension. To facilitate the retrieval, the characterization record of each documentary resource has, besides the general fields (identifier, author, address, size, version, publishing date etc) and those dedicated to technical (T) and administrative (A) references, some fields referencing (indexing) the content (notional domain- D).

References to knowledge and competences The "knowledge references" encountered in the indexation of documents pose problems to that of person referencing case (in fact an explicative document can be considered the author's representative towards the expected user). Therefore appear the "*competences*" - eventually organized by abilities and postures.

Instruments and their concretization. Also similar are the resources' selection mechanisms concretizing the abstract documentary instruments (searched freely or specified in a scenario or aggregate)- based on references within descriptive records.

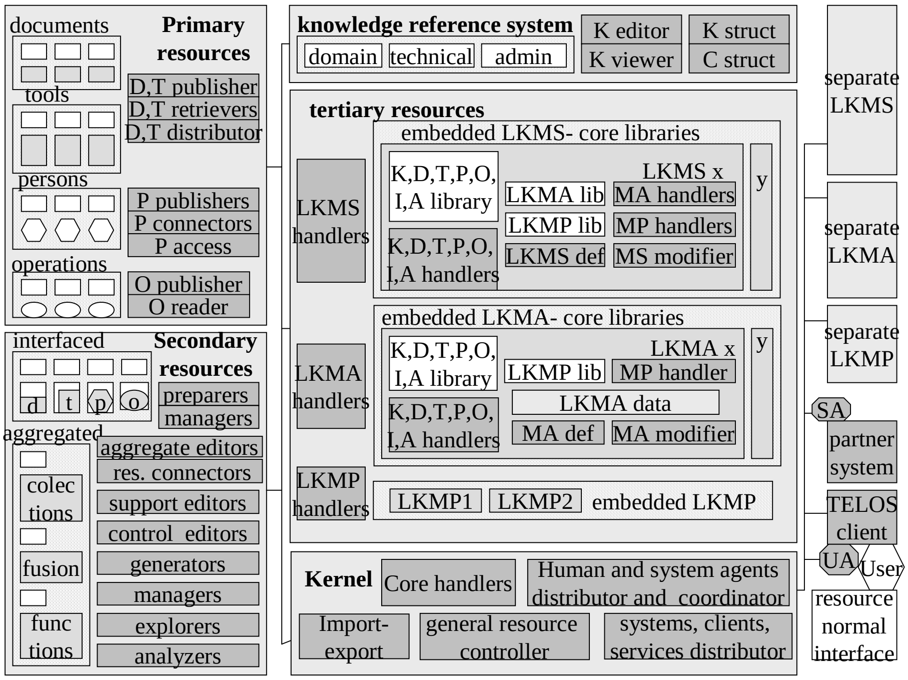

figure 1: TELOS fabrication base

Secondary resources. The management of physical objects (without a computer interface) is reduced to their declaration, retrieval and negotiation. The computer applications can be enriched with access and manipulation facilities, the primary resource being "wrapped" with an "interface" layer, which can intermediate its use. Based on this secondary resource's "methods", the primary resource can be manipulated remotely by a participant or even by another resource (or agent)- piloted cooperatively or automatically ("command batches"). But the most important facility created by wrapping a resource is the preparation of its use in aggregations.

Resource aggregation [9] Is the process of composing a new resource from the ones already registered in the system. *Collection* aggregates are simply sets of resources, accompanied by elements' management facilities. *Fusions* - are systems having the global behavior determined by components' interconnection. *Orchestrations* (see next paragraph) are procedural aggregations - the resources being connected to the operations in which they must intervene. Continuing the aggregation process, we can obtain increasingly complex structures. These ones will enrich the (secondary) resources library of the TELOS system (or of a system built with it).

Core knowledge, primary and secondary resource management. In addition to the zone dedicated to the management of knowledge reference systems and of repositories designating or containing primary entities (resources, persons, tools and documents, operations - see below), TELOS contains a secondary resource library (interfaced or aggregated: collections, systems and orchestrations) along with the instruments (handlers) required for their manipulation (edition, use etc). TELOS engineers add to these zones the handlers and libraries required for the management of system aggregates (LKMS, LKMA, LKMP) and the kernel- for managing the agent communications (atomic services).

System aggregates (tertiary resources). LKMS. With the help of a specialized editor, technicians can use secondary resources as raw material (or as design instruments) to build "learning and knowledge management systems" (*LKMS*) - placing them in their dedicated library ("embedded") or installing them in an external beneficiary's context ("separated"). As any aggregate, a LKMS can pass trough concretization (adaptation) from the model state to the state of executable instance. The functioning of these LKMS can (linked case) or can not (autonomous case) maintain ties with the core - depending on the completeness of their structures: starting with the "thin" cases (containing generally definitions and data), going through the "fat" case (containing the LKMA produced and managed by the LKMS) and ending with the "heavy" situation (containing the base handlers, for autonomy).

LKMA. Using an LKMS, designers can construct learning and knowledge management applications (*LKMA*), placing them in the system's central library, in the library of the used LKMS (embedded or remote) or completely apart. After the edition (by a course designer), the LKMAs can also pass trough a particularization phase (by an administrator) before the start of the instance execution. The LKMA' use will generally require the support of the producer LKMS (and eventually even core services), but it can also gain autonomy trough the enrichment with appropriate handlers (becoming LKMA"). The simplest form ("thin") of an LKMA may contain only his aggregate definition (pointing to the resources and the handlers placed in the core (or LKMS regions) and the exploration results.

LKMP. The material results (traces, annotations, user constructions) of an LKMA use (that go along with learning), called here "learning and knowledge management products" (*LKMP*), can be managed in the context of the generating LKMA (eventually necessary in their observation or interpretation), placed in the libraries of the core or of the support-LKMS, or in personal portfolios. The system's feed-back loops (repositories' enrichment, resources' evaluation, competences' evolution, knowledge reference modification- can originate the LKMP' analysis.

Phylogenetic cascades. A component produced by the system core engineers can be adapted and incorporated in an LKMS, then placed into an LKMA - from where it can finally get into an LKMP. For this reason, TELOS insist on the longitudinal treatment of process chains, exploring the modeling, adaptation, orchestration and reproduction formulas of procedures, with the help of the function manager ([12]/gefo).

2.5 Procedure management (see also [12]/gefo)

2.5.1 The procedures and their decomposition/ aggregation/ connection

A procedure has a dual character, as a "structure in process". Structurally, it is formed by interconnected "components": persons (the actions' executants and their assistants) and objects (to be used or produced, or supporting resources). Processually, it consists of a chain of actions (steps, phases). The combining of the structural and processual approaches leads us to a decomposition in interlaced "threads" (roles).

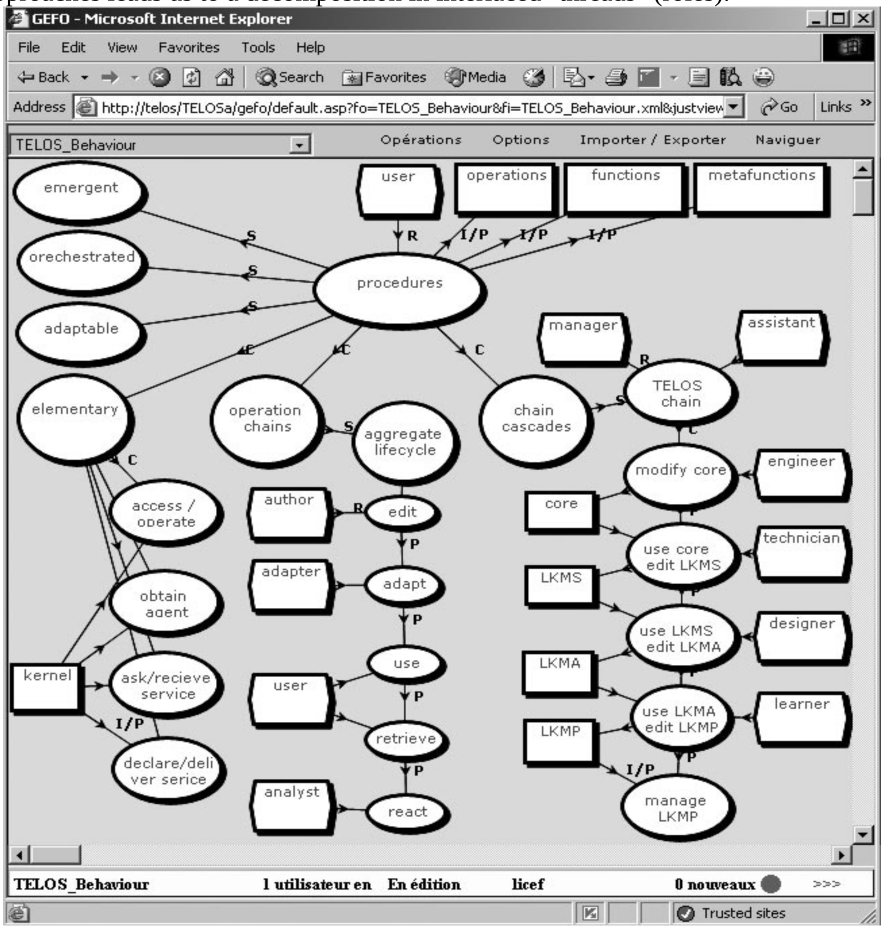

Figure 2: Procedures' management

The procedures are represented /modeled/ managed/ reproduced by *operations* (witch describe the basic action and the implied elements), *functions* (witch describe the decomposition in phases and roles) and *meta-functions* (witch describe the relationship between a real procedure and its model). Pragmatic considerations and the adoption of some metaphors from biology lead us to consider the following granularity levels: elementary operations, operation chains and chain cascades

Elementary procedures. Action units having a coherent goal, implying at least one protagonist-supported by persons and objects and producing micro-services chains.

Operation chains. The component operations are chained temporally (by precedence relations) or trough common elements (actors, resources)- conforming to "workflow" and "flow control" rules (decisions, joins, loops etc) depending on execution data. It may be the case of a material operation chain- to be performed/learn. Or of a cognitive process: understanding or learning. Un important case is the "*lifecycle of a resource*" (edition/ adaptation/ retrieval/ use/ annotation/ modification etc). The composition of an aggregate [9] pass through such an "ontogenetic" cycle. When the evolving aggregate is a model and the cycle reflects its dynamic relation with the reality witch it first describes and finally guide- we deal with a meta-chain/function.

Chains cascades. Procedure aggregation can continue in both directions: the actions can break up in sub-procedures and the procedure as a whole, can be part of an including meta-procedure. The same operation may belong to several chains, at different granularity levels. A resource/person resulted from a procedure can be used/implied in another- as an input/subject or support object/person. Thus appear the "phylogenetic" cascades of knowledge and objects reproduction (for example, the main production cascade of TELOS)

2.5.2 Representation and facilitation levels

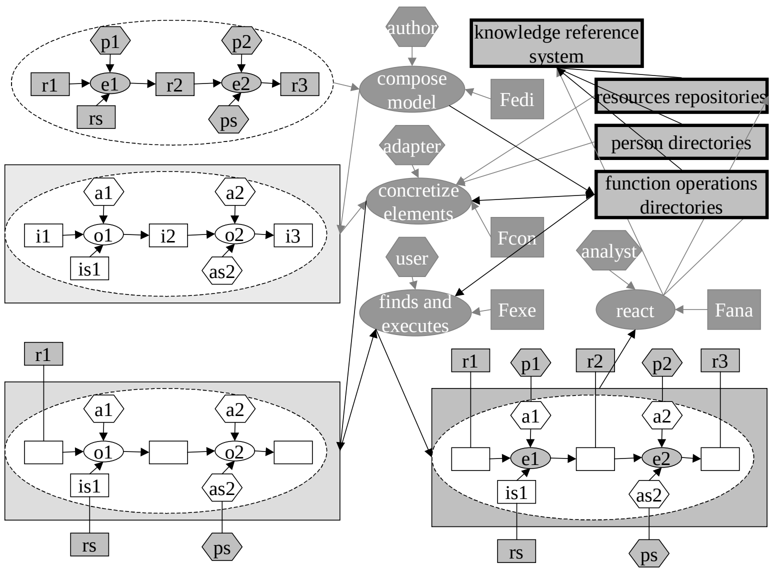

figure 3: Procedure representation: operations, functions, metafunctions

The operation- is a model of a procedure, using representations for the elements reflected in its "mirror": actors (hexagons- designating generic participants or specified persons) and instruments (rectangles- generic or concrete resources) connected to the central activity (realized or to be realized- represented with an oval).

The function- represents the procedure's decomposition in actions (and eventually the rules for deciding the continuation).

Function/operation life cycle. After the definition of a generic model and the particularization of "derivate models" by the concretization of some appropriate elements- found in the persons or resources directories-, these models can be indexed and published in a repertory, becoming retrievable - as any resource. The users of such "procedural aggregates" take advantage of the assistance facilities prepared during edition: operation guiding, execution supervising, resources manipulation, participant coordination, run-time support connection (matching etc).

Meta- functions. The transformation cascades operating between a model and the procedural reality that it mirrors and (re)produce (as the one presented in the above figure) can be managed with/as *meta-functions*. These tools may pilot the component "progressive concretization" process (conforming to the function "life mode")- so that the "competence conditions" be respected or optimized.

Emergent mode. 1 The participants having this right (mandate) add new documentary resources to the appropriate repositories 2 The resource users exploit the retrieval instruments (based on the link between the language of the requests and the knowledge reference spaces employed for the semantic, technical and administrative indexation). Some facilities for adapting the selection according to the users' competences may also occur. 3 Operations are chained freely, according to the users' necessities (initiatives). The concurrence situations are solved by negotiation mechanisms. 4 As a result of using some resources, data (traces or annotations) can appear. This information constitutes suggestions for refining and correcting references or even reference system- in subsequent phases. 5 Through participative or exterior actions observation, or through post-factum data analysis, the emergent process' occurrence can be understood and eventually modeled. It can be a source for conceiving "operations" and "functions" - preparing analogue chains' orchestration.

Orchestrating mode. The actors A (and generic instruments I) appearing in the operations' models have c(a/i,k) competence characterizations, analogue to those of the participants P (or documentary resources R) witch will concretize them- allowing the action of selection criteria as c(p,k)>=c(a,k). If all operation's elements have been specified (connected) - with the exception of its "user" - we are dealing with an assistance "contract", placed in the "prepared activities" repertory, waiting for its clients. The competence levels required (C1) and obtained (C2) for/through the execution of this particular activity are signaled. But the concrete users have a level c1 (instead of C1), and intend c2 (not C2). Thus, the effective execution acts like an "competence operator", changing c1 in L(c1)- witch can differ from C2 and c2 (the presumptions about the lesson's effect having only a statistical value). We can optimize (assist) the selections (persons and connected documents) operated in any phase of the concretization chain, if we provide mechanisms for the surveillance of each operation's internal competence equilibrium. These facilities are created by using the same reference systems for the indexation of actors (persons), operations (activities) and instruments (documents) and by defining competences by postures. The rules (equations) that intervene depend on: the procedure's "topology" (Toead= operation, executor, assistant, supporting document, Toea, Toed, Toe, etc.), the concretization order (for instance: first d(o), then a(o,d) and finally e(o,a,d)) and the assistance strategy. For example (see [6]), for an operation requiring a competence level O, approached by a learner having a competence C, supported by a an assistant capable to sustain C1- C2 increases and by a document capable to sustain C3- C4 evolutions, we can observe situations as: *(C1<=C<O<=C2* or *C3<=C<O<=C4*)- any support component is sufficient, or (*C1<C<C3<C2<O<=C4)* - the assistant can lead the executor in the document's efficiency range.

Adaptable mode. Although the mechanisms suggested above are useful in the model definition or adaptation phases, they can also intervene in the execution phase, if concretization liberties have been allowed. This way, run-time "semantic services" are put into function, handled by optimization agents- giving assistance in selecting connectable resources and persons, alerting participants, matching automatically etc.

2.6 TELOS production chain

2.6.1 Core-modifications.

Can be done in the limits imposed by the backward compatibility problems.

Engineers' operations. After launching the system, the TELOS engineers will be able to modify the core structure, handlers or libraries:

1 Modify the core K reference system (domain, technical, administrative)- protecting or recalculating the existent semantic references.

2 Modify the core structure: add new types of resource wrappers, aggregators or cascade fabrications; update the core handlers.

3 Add, suppress or modify knowledge handlers (ontology and conceptual maps editors and viewers, librarian cataloguing and indexing tools etc.) and modify knowledge library structure; modify primary resources libraries' structure and secondary resource handlers (documents, tools, persons and operations wrappers, aggregate editors and executors etc).

4 Modify support (for core- as a global tool).

5 Modify global control layer (eventually updating the kernel).

6 Annotate, produce traces, receive support- eventually operating cooperatively.

7 Save, leave, resume- as for any multiple- session activity, determining the evolution of the TELOS system.

Administrators' operations. A core administrator manages the content of primary, secondary or tertiary resources' libraries. He can act at another actor's request (technologist, designer or learner), supervise this actor or delegate to him the intervention rights.

1 Add, suppress or update knowledge documents (reference domains): ontologies, conceptual maps, thesaurus, catalogues, etc.

2 Add, suppress or update primary resources records (for documents, tools, persons and operations) or secondary resources (wrapped/interfaced, aggregated)

3 Modify tertiary components (a secondary resource or resource handler placed in a LKMS or LKMA library, a LKMS, LKMA or LKMP handler or library structure)

2.6.2 Core's use to LKMS' design

The LKMS design can be managed as a complex aggregate composition. Thus we can signal operations as: finding components, making notes, producing traces, receiving help etc. But we will concentrate on specific aspects.

A Compose an LKMS-template (model)

Technologists' operations.

1 Organize the LKMS knowledge reference system (related to the core's one)

2 Organize the global LKMS structure (definition).

3 Organize LKMS parts: add, suppress or modify primary, secondary or tertiary resources (handlers and documents)

4 Organize support and control for LKMS use- depending on the destination context (embedded, linked, autonomous)

5 Annotate the construction, produce traces and receive support by tools (for example, metafunctions) or by specialized actors (assisting engineers).

6 Save, leave, suspend or resume LKMS multi-session or cooperative constructions

Facilitators' (engineer) operations The composition may imply assistant engineers (observing, guiding, evaluating or replacing the LKMS composer)- intervening synchronously or asynchronously (using eventually a support metafunction).

B Prepare (adapt) LKMS for host installation

Administrators' operations. The LKMS administrator may define particularizations of every executable LKMS instance, adapting it to the use context. This can be done (eventually cooperatively or driven by a "LKMS preparation meta-function") in one or many LKMS instance preparation sessions.

1 Chose a saved LKMS template and create an new adapted instance.

2 Prepare the LKMS instance (extend edition) by parameterization and component concretization, deciding the thin-fat-heavy alternative and the deployment context and finally, installing the instance in the use context.

3 Note, produce traces and receive support.

4 Save, leave, suspend, resume the LKMS multi-session or cooperative adaptation.

5 Activate LKMS for work in embedded or external (linked or autonomous) modes.

6 Analyze the core use and make feed-back propositions.

Facilitators' operations. The adaptation may imply assistants (observing, guiding, evaluating or replacing the LKMS adapter) - intervening synchronously or asynchronously (using eventually a support meta-function).

2.6.3 LKMS' use for LKMA (lessons, support tool etc.) design.

A Use core or LKMS to compose or modify a LKMA template

Having in mind that the authoring instruments equipping the LKMS installed in a beneficiary's context come from the core's tools base - there can be situations when LKMAs are conceived directly on this base. But even then, we may consider an LKMS - consisting only in the used authoring tool.

Designer's operations

1 Organize K layer. Generally, the reference system used in a LKMA already exists in the LKMS or core context. However, sometimes we may provide a local semantic reference system or a local "add-on" completing (adapting) an external reference. The must interesting case is that of an LKMA dedicated to the piloting the use-based evolution of a knowledge domain.

2 Define structure. The central part of any LKMA is its aggregate definition (in the thin case- it may be the only part). It can consist in one aggregation layer (conforming to the collection, fusion, function or other aggregation types), or in a recursive cascade of aggregate definitions.

3 Organize LKMA package. Add, suppress or update the aggregated resources (for a fat LKMA): interfaced documents, tools, persons and operations. Add resources handlers (for a heavy aggregate). Add a LKMA modifier, if a real-time adaptation capacity is desired. Modify the LKMP library and handlers.

4 Organize actor access and control layer. Prepare the control C for the LKMA use- depending on its activation type (embedded, linked, autonomous): the control declaration- for thin LKMA, protocols - for fat LKMA, handlers- for heavy LKMA

5 Organize support for the LKMAs use. Prepare S part of the declaration layer- for thin LKMA, support rules- for fat LKMA, S handlers- for heavy autonomous LKMA

6 Note and produce traces about the composition process. This data is not included in the final aggregate, but can guide the current (or a subsequent) composition experience.

7 Cooperate and receive support for the composition activity (observed directly or by traces) from assisting tools (for example, metafunctions) or from specialized actors

8 Save, leave, suspend, resume a LKMA multi-session or cooperative construction

9 Publish LKMA template, when the construction process is finished.

Facilitators'- operations. The composition may imply assistants (designers, technologists etc) intervening synchronously or asynchronously (using eventually support meta-functions).

B Prepare (adapt) LKMA for host installation

Administrators' operations. The LKMA administrators may operate multi-session or cooperative adaptations (particularizations) for every executable LKMA instance, adapting it to the use context.

1 Chose a published LKMA template and create a LKMA instance

2 Prepare a working LKMA instance (extend edition- parameterization, concretization etc). The amplitude of the adaptation process depends on the flexibility foreseen by the LKMA composer. The distribution of the concretization process between the composing, preparing and using phases respects the "life cycle mode".

3 Annotate, produce traces and receive support.

4 Save, leave, suspend, resume the multi-session or cooperative LKMA adaptation.

5 Activate LKMA for work in embedded or external (linked or autonomous) modes. For some types of aggregates, this step may imply a "compilation", producing an "executable LKMA".

6 Analyze LKMS use and propose structural and behavioral modifications

Facilitators' operations. The adaptation may imply assistants (technologists, administrators or designers) observing, guiding and evaluating the LKMA adapter - intervening synchronously or asynchronously (using eventually a meta-function).

2.6.4 LKMA' use and LKMP generation and administration

Considering the situation analogous to those presented above, I will pass up the textual explications. In their place, I expose here this last use case schema, as it was conceived in the conceptual architecture document ([12]/telos) - to illustrate the function use potential in modeling, demonstrating or managing the TELOS system. The figure reveals the system's target-process: the use of LKMAs (from the central core library, the library of some embedded or separate LKMS or apart) by learners, assisted by facilitators, to improve their competences. In the same time, protagonists produce material results (resources, traces, notes) - that can enhance LKMP libraries.

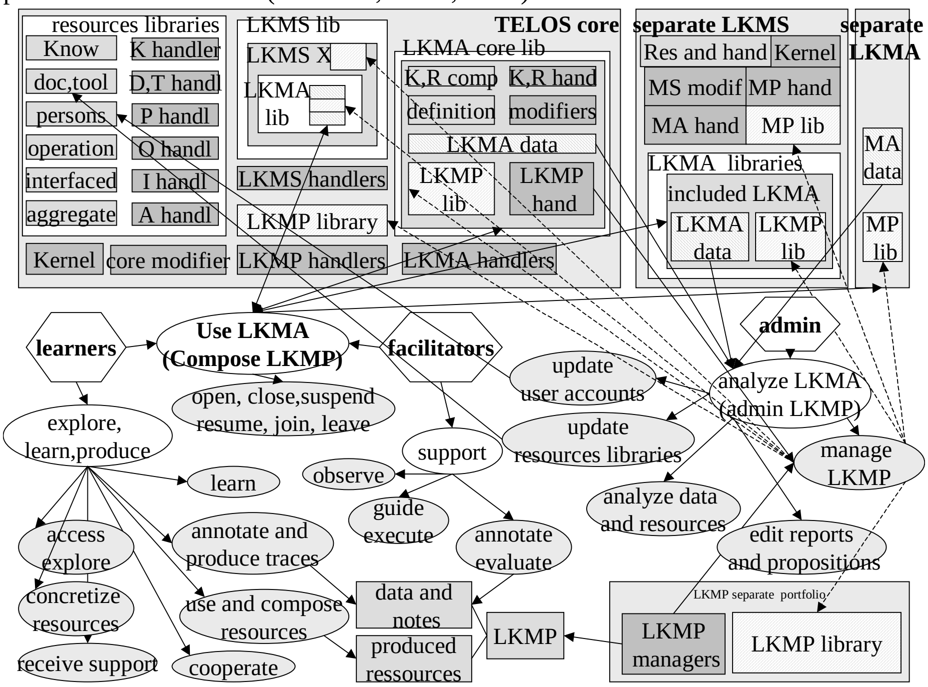

Figure : LKMA' use for learning- and LKMP administration

The second cycle presents the post-operative analysis of the educational sessions' results (made by an administrator - or by other actors) and the carrying out of appropriate reactions (reports, competence profile modification, resource library enrichments). A complex operation is also the LKMP' management, taking place (according to version distribution protocols) between: the autonomous LKMP libraries of a user or user group (portfolios), the core LKMP library, the LKMP libraries of some LKMSs (core embedded or separate) or the library of the producing LKMA (from the core, a core LKMS, a separate LKMS or autonomous).

3 Conclusion: the management of the LORNET project

As we have already seen, the procedures' management through functions can also be used for the description, demonstration, management or orchestration of the system operations forming the TELOS physiology. Furthermore, the cooperative research activity taking place in LORNET was described and could have even been managed through specific functions. An interesting engineering situation was created - in which the instruments produced in a project can be useful to its organization, allowing refined and fertile strategies for ascending the research-development-application spiral. That is why we could adopt and enrich the RUP software engineering method (axed on the "use-cases" management- during all development's phases) [4].

That is also why we have used [7] the GEFO manager to illustrate the desired behavior of the LORNET system. But to realize a coherent management of the project through metafunctions, the meta-ontology of the research activity should be finalized and the cooperation between the implied teams and the resource and ideas sharing protocol- be better organized. The longitudinal management of the TELOS physiology - perhaps its most interesting characteristic - also asks the improvement of the education meta-ontology and of the function manager robustness.

References

1. Bloom, B.S. (Ed.): Taxonomy of Educational Objectives, the Classification of Educational Goals, Cognitive Domain, New York: David McKay Company Inc., (1956)
2. Paquette, G., Rosca, I., De la Teja,I., Léonard, M., Lundgren-Cayrol, K.: Web-based Support for the Instructional Engineering of E-learning Systems, Proceedings of WebNet'2001, Orlando FL, October, W. Fowler, J. Hasebrook (eds.) pp. 981-987.(2001)
3. Rogozan, D., Paquette, G., Rosca, I.: Evolution of an ontology used as semantic reference system in a tele-learning system", Université de Technologie de Compiègne, 243-249, http://archive-edutice.ccsd.cnrs.fr/edutice-00000723, (2004)
4. Rosca, I., Paquette, G., Mihaila, S., Masmoudi, A.: "TELOS, a service-oriented framework to support learning and knowledge Management" E-Learning Networked Environments and Architectures: a Knowledge Processing Perspective, S. Pierre (Ed), Springer (2006- in press)
5. Rosca, I.: Semantic indexation and knowledge propagation, Hermes, (2006- in press)
6. Rosca, I.: Knowledge management instrumentation for a community of practice on the semantic Web, http://pedagogie.ac-montpellier.fr/Disciplines/maths/REF_2005/REF-Rosca.pdf Symposium REF-2005, Montpellier (2005)
7. Rosca, I., Paquette, G.: TELOS research progress, LOR'04 Towards the educational semantic web, Vo1, No4, Dec, http://www.lornet.org/eng/infolornet_vol1_no4.htm#a35 (2004)
8. Rosca,I., Paquette,G.: System orientation principles, TELOS Vision and Orientation http://www.lornet.org/docs/telos.pdf, pp26-28 (2003)
9. Rosca, I., Paquette, G.: Organic Aggregation of Knowledge Objects in Educational Systems, Canadian Journal of Learning Technolog Volume 28-3, pp. 11-26 (2002)
10 Rosca, I., Paquette, G.: The Explora2 system, Congrès TeleLearning, Vancouver (2001)
11. Rosca, I.: Towards a systemic vision of the explanation process; the story of a research on integrating pedagogy, engineering and modeling- PhD thesis, http://www.ioanrosca.com/educatie/these, (1999)
12 Rosca, I.: Technical reports for GEFO and TELOS, www.ioanrosca.com/education