Five years for five perspectives on TELOS

Ioan Rosca
LICEF, Teleuniversity of Montreal, Canada
ioan.rosca@licef.teluq.uqam.ca

Abstract

This paper explains the ideas proposed by the conceptual architecture of the distributed system "TELOS", describing the concerns that led to them, along the MOT- ADISA- EXPLORA- ION- GEFO-LORNET project chain… The struggle between integrating applications and decomposing them in components- usable for recomposing systems with variable geometry… The prototyping of a resource controller supporting the aggregation by "fusion" of secondary resources (which wrap the primary ones)… The introduction of "functional" aggregation, binding resources to operations, tying processes modelling, orchestration and reproduction… The management of system evolution- with "meta functions"… The shift from structural composition to service concatenation, supported by a communication bus and "interface-agents" controlled by a kernel… The treatment of semantic inter-operability, using "knowledge domains"- as reference systems… To finally define an evolving and plastic middleware, combining structural extension (by distribution, recursive aggregation and phylogentic production cascades) with segmentation- on administrative criteria.

1. Introduction

The considerations of the next chapter constitute an introductive survey through the problem-space synthesized in chapter 3. Observing the final form of an intellectual construction is not always enough for the comprehension of its relevance - in relation to the problems it tries to solve. An eloquent explanation of a solution for a complex problem (like the one exposed here) can justify the description of the research's evolution, on the path of the main difficulties, intuitions and choices - seeking a compromise between multiple and contradictory criteria (modularity/integrity, portability/optimality, complexity/flexibility, etc). Thus, I will use the direct rhetoric of narrating my experience in the place of the traditional one- based on the analysis of considerations exposed in the literature. The competent reader will however easily perceive the background of the exposition. I assume this narrative methodology, believing in its expressive virtue in highlighting the "why?" of formulas like: "wrapping secondary resources", "distributing interface- agents", "recursive aggregation", "production cascade", "indexing competences", "emergence- orchestration mix"- on which the conceptual architecture of the TELOS system has been based.

2. History of a research

Figure 1, depicting the adventure of 5 years of research, could have been cut out and presented in steps. I preferred to render the global image, explaining it progressively. The description blends 5 incursions in the physiology of a distributed system in expansion: the structures, the processes, the services, the knowledge evolution and the administrative management of production cascades.

2.1. Between integration, decomposition and recomposition

(2.1a) I began my activity as conceptual architect for the LICEF systems with an integration mandate. I had to find a manner to bind several applications, conceptually interrelated, but developed in parallel projects:

1 Local graph editors (MOT -[1], AGDI, Exploragraph)- used for modelling: knowledge structures to assimilate, scenarios of pedagogical activities, structures of material resources to build, delivery plans, etc.

2 35 metadata editors (named e100, e104, e212, e214 etc) based on templates (Word, Excel, PageMaker) - for the specification (in corresponding data islands: d100, d104 etc.) of the instruction system to built: goals, participants, resources, activities,

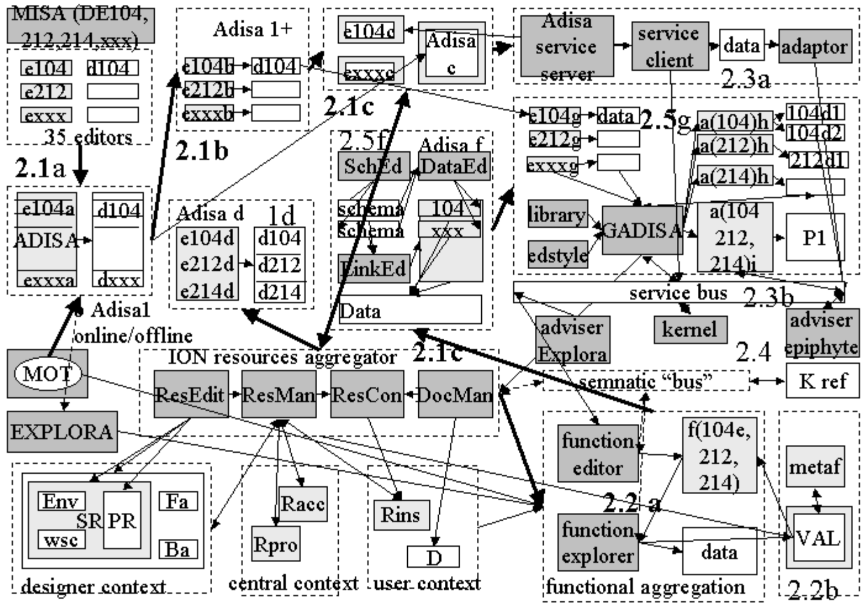

Figure 1.: 5 years in the aggregation quest labyrinth

implementation phases, cost estimation. The flow for a complete plan edition being controlled by the MISA method [2] and exposed as a Word document or as a task graph- rendered with the MOT editor.

3 A platform for virtual campuses (EXPLORA [3])- integrating the management of various categories of participants (administrators, designers, learners, tutors, etc.) with a system for managing guided course explorations- in web format.

I was asked to find a way:

- to bind in a coherent whole the various textual metadata editors (graphical and textual), ensuring the propagation of the data introduced into a documentation element, towards the elements influenced by it (downstream, in the flow of the MISA method)
- to distribute the design activity, considering the fact that a "MISA project" requires the intervention of experts of different specializations (knowledge, pedagogical, material design, delivery etc)

After a double analysis (mediating between the technicians and the researchers teams), we chose a compromise solution, developing the ADISA system [4]. I will only expose here the aspects related to the modularity/unity contradiction (at the operational and data structure levels).

1 All the metadata files (d104 etc) are edited (reedited) in ADISA with IEXPLORE DHTML forms, valorising the interactive features and the adaptability of the interface (to the editors' preferences or reflecting some upstream changes).

2 The data introduced for a documentation element X (for instance with the e104a editor) - is placed in the corresponding xml local file and is propagated to other "xml islands" Y- if recommended by MISA methodologists. There are several types of propagations: "automatic" (the user of Y will take note of them at the next session), "selective" (the editor of the Y island is able to accept or refuse the modifications suggested in upstream) or purely informational (they don't modify the data island Y, but only offer useful information to its editor).

3. The graphs edited using the MOT application (included with a COM object in the DHTML edition process) also produce XML files. From here, the data can be propagated to other islands, through the mechanisms already mentioned. This way, the human ergonomics of the graphical edition is combined with refined data management (allowing appropriate propagations, global analyses and easy retrieval of information introduced graphically).

4. The data of a local project can be edited in a number of offline sessions. Afterwards, it can be propagated towards the central Web server. The server solves the problem of cooperative project edition (by check out/check in - type mechanisms). The "online" edition is also possible, involving the same interfaces and data flows.

(2.1b) Immediately after the integration, in ADISA, of the correlated editors for forms and graphs… the need for modularisation appeared. The conception of a complete instructional project, useful in complex situations, had proven to be cumbersome in others, the efforts to follow the method from the start to the end being unjustified.

We therefore passed to "the extraction" of the 35 editors from the ADISA context, transforming them in autonomous offline/online editors for every informational segment (MISA documentation element- which I will note DE). The realization of the editors in DHTML - XML - facilitated this process. The only difficulty was created by the propagations that had strongly coagulated the system (for example, the forms of the competence editor E214 were built dynamically according to the list of participants - introduced with E104 and of concepts - declared with E212). For realising the autonomy of each DExxx-b (by preserving the compatibility with the corresponding DExxx-a, version, integrated in Adisa) we operated an "externalisation" of the dependencies from the Javascript code, towards a configuration XML (seeking the reorganization flexibility -see a similar idea in [5]). The DHTML editor E214b- for example- reads the contextual relations from its configuration XML, discovering if it must work with its local data structure, actualise the knowledge (participants) list- from certain addresses or receive them from the flow of the incorporating ADISA project.

(2.1c) We also approached the problem of managing the separated Exxx-b editor collection (baptised "Adisa1+") outside an ADISA project context. Being a "collection type" aggregation problem, we initially thought that it would be naturally solved within the framework of the "resource manager" built in another LICEF project. The metadata organization norms used in the Edusource project (to ensure the compatibility between pedagogical resource repositories) allowed enough freedom to declare configurations (working modes, dependencies).

But the difficult problem encountered by the distribution and use of the DExxx-b modules proved *not* to be their declaration and publication (transferring them from the designer context towards the central base of the resource manager ResMan -for the DExxx editors- or of the document manager DocMan -for the corresponding data files); nor their retrieval to the user context (operated by resource manager clients). It was to insure the technical operating conditions in the (variable) local contexts, also solving data dependencies - in the case that the interoperability with other DEs was asked by propagation links. Thus, we have encapsulated the Exxxb resources in "WSC"s (COM objects created "on the fly" by the local controller on the basis of: the primary resource, a XML description of its "external" methods and the javascript code of the "wrapping capsule"- supporting these methods). This solution produced Exxxc "secondary resources", eliminating the need for knowing the local address of the DEs to connect to.

Departing from these experiments, we attacked two broader problems, which proved extremely sharp. The first: how to organize the aggregation of (pedagogical) resources composed by fusion (the global system depending on the relations between the combined parts)? The second, related to it: how to facilitate the process of these aggregations by preparing the resources used as raw material and by using a "resource controller" able to valorise this preparation? I explored these subjects with a group of architects, interested in problems as interoperability, modularisation, orchestration of dynamically established architectures (S. Mihaila, C. Mitocaru, L. Vornicu). Together, we noted the difficulty of combining autonomy, portability and compositionally - a problem treated intensely in software literature. But we were interested in inter-operation and aggregation at the whole application granularity and not at the "components" level (works like [6] being an important source of inspiration).

Respecting the Internet architecture spirit [7], we have searched for interoperation principles between client-server (or multi-tire) applications- on which we could found the Explora, SavoirNet and TELOS systems: from the "connecting user" (which can link applications, chaining the operations that use them) or the "connection through data" (propagated or transferred between documents) to the "integration on the user computer" (taking advantage of the operating system possibilities or of an intermediary shell as a browser, interpreting DHTML), the "connection through servers", or the "cross-communication" (client of application A working with the B server) etc. The problem proved difficult. The result of our investigation was a "manifest" (composed by S Mihaila) regarding the principles of an architecture based on "dynamic and recursive agregability".

For exploring integration formulas we have used the prototype of the distributed resource controller ResCon included in the ION prototype (developed by Val Rosca). The treatment of the Adisa DExxx modules provided the suggestion that all the "primary" resources PR be prepared ("wrapped", "interfaced", "encapsulated" - using a JavaScript file Env, carrying out the external methods declared in a XML - WSC) so that we obtain "secondary" resources SR- easy to manipulate by the resource controller. The prototype also included an editor ResEdit for the definition (in xml) of generic resources (called "resources families"- Fa) with abstract methods (services) formulated in a homogeneous language ("EXPLORA script"). The key issue proved to be the "wrapping" of primary resources in secondary capsules, so that the abstract methods be placed in correspondence with those incorporated in the wrapped objects. Despite the attempts to automate (facilitate) this "binding", the preparation of the secondary resources (interfacing drivers towards the primary code and implementation in JavaScript of the abstract methods) remained the programmers' task.

Once wrapped in the capsule of a "resource family" Fa, a resource (local application, client of a client-server or multi-tier application, etc.) can be: installed by the controller in the local context, placed in communication with other resources, combined in aggregations, replaced by an available resource of the same family, handled (remotely or co-operatively), spied (intercepting actions at the interface level), handled by "batches" Ba of operative, demonstrative or instructive commands - formulated in "Explora script" language. ION provides textual and graphical editors- for the creation of these batches and executors- that pilot the controlled resource and illustrate the advance along the operation workflow.

(2.1d) Finally, we demonstrated that we could dynamically recompose (from the encapsulated autonomous DEs) the complete ADISA system or a coherent sub-ensemble of it- using the ION controller. For example we have composed "E-104-212-214 workshops", capable of editing D-104-212-214 integrated data structures. But the "authoring" operation proved to be difficult and the generalization of the "fusion" aggregation methods and instruments posed principal, technical and strategic problems, exceeding the possibilities of our team.

2.2 Between modelling, orchestrating and reproducing processes

Complex structural aggregations (like ADISA workshops) call for the assistance of their composition or use. The organization of "learning objects repositories" is therefore complemented by the instrumentation of authoring processes and of educational cooperative activities.

The modelling of "concurrent processes" (transfer, cooperation, use etc.) -constituting "events" of a community physiology or interactions between 2 communities- can be complemented with that of long processes (as resource "lifecycles" or global evolutions of systems). Another level at which procedural representations can intervene is that of "phylogenetic" chains binding production cycles extending a system: an object made in a process being used -as instrument or raw material- in another.

(2.2a) I have therefore carefully analysed the use of the MOT+ graph editor, built by LICEF, for the modelling of ontogenetic, physiological, evolutionary and phylogenetic processes. I observed the manner in which MOT models allow the reflection of working, cognitive, cooperative, pedagogical and diffusion procedures [8], respecting the "ergonomics of human interpretation" (visual representations necessary in human- human communication). I also studied the mechanisms necessary to the man-machine communication and to the computer agents' dialog- comparing the MOT pedagogical workflow (learnflow) formulas with similar developments coming from CSCW (or CSCL) and watching interoperability solutions based on norms like EML or IMS-LD [9].

The other important problem approached by the LICEF researchers was the coordination of the actual procedures. To the refined but passive graphical descriptions composed in MOT, corresponded- in EXPLORA- simple task trees, nevertheless capable to orchestrate the planned activities. Trying to eliminate the discontinuance between the models' edition and their exploitation - existing in the LICEF chain- I realized the complexity of the scenario enactment problem (see also [10]) - understanding why it was lagging behind the rest of the IMS-LD like developments.

We explored new formulas for processing procedures with the "function manager" prototype [11]. The VAL editor (that became "GEFO" when it was included in the LORNET project) uses the ION resource manager and controller to bind to the workflow primary, secondary (wrapped) or tertiary (batches) resources. For example, by connecting the corresponding editors to a workflow representing the edition process of the 104- 212- 214 MISA chain and by graphically specifying data propagation bridges, a "dynamic aggregation" is obtained- active at the moment of the function execution (exploration).

The facilities offered by the function explorer are: 1 inspiring the actions' sequencing; 2 declaring and producing exploration data, usable for support reactions; 3 launching and controlling resources, facilitating their manipulation and their procedural aggregation; 4 mediating the participants' communication and coordination; 5 providing retrieval and matching services for the run-time concretisation of the components.

(2.2b) The great asset of the functional aggregation formula is the (recursive) possibility of its management- using "metafunctions". These allow the modelling and the orchestration of the global physiology of the loop formed by the procedural reality and its model. For example, we can manage the evolution of the function that aggregates the f(104,212,214) ADISA workshop with the help of an adequate metafunction, eventually using the fact that the function editor in its whole has been wrapped as a secondary resource (therefore allowing: capture of gestures, injection of commands, distance and cooperative control, demonstrative batches etc).

For the conception of an "expanding distributed system" the "use cases" are not only a method for orientating the construction of new tools. They reflect the physiology of the system in extension, evolving along with it (see a similar approach in [12]). Stable "structures" are replaced by structural (or procedural) aggregation and disintegration mechanisms. If the first version of my vision for TELOS was called "architecture" (trying to emphasize the system "blocks") and the second was called "behavioural design" (being based on use cases), the final name ("conceptual architecture") assumed the indissoluble fusion between structural and behavioural aspects - treating the organization of plasticity.

2.3 Between distributed architectures and the services bus

(2.3 a) As I already mentioned, I realized (in the ADISA - ION projects) that the management of the structural disintegration - aggregation process raises important technological, strategic and principle difficulties- having to resolve contradictory requirements: autonomy versus connectivity, small granulation for flexibility/ large- for operationality etc. Attempting the "componentisation" of the MOT+ editor - realized with MFC / C++ - and of the Explora platform - realized with java applets/servlets - in order to recompose virtual campuses with variable geometry equipped with co-operative scenario exploration tools, we encountered difficulties tempered our enthusiasm...

For this reason we have welcomed tendencies like "web services" and "application servers" that we saw (beyond the technological details of the SOAP-WSDL-UDDI chain) as a reorientation from morphological compositionality towards segmentation for dynamic and supple aggregation.

In the LICEF environment, interfacing the existing applications, so that they can ask and deliver services to partner applications, using a shared language and an adequate communication bus (in the sense of a semantic protocol overlapping the technological ones) - proved to be a valuable idea. Our first tests of "communicational aggregation" were based of the DExxxc editors, taking advantage of their encapsulation as secondary resources; that allowed the mix of manoeuvres on the direct interface with orders transmitted through the resource controller. The "wrapping" of the entire Adisa application as a "secondary resource" with remote manipulability allowed the piloting of the edition process with demonstrative "batches", managed by the ION executor or with "functions", managed by the VAL explorer.

We then organized an "ADISA service server" able to provide relevant information to users equipped with service clients, or to the Exxxb mini-editors of the Adisa1+ range. After the calibration of the "service bus" idea (by the creation of a link between the VAL function manager and the rule-based "adviser" [13] which equip the Explora platform) we began to apply it for realizing bridges between the LICEF applications. We realized, on the same principles, the demonstrative connection of an "epiphyte adviser" [14] to the ADISA2 generated workshops (see 2.5).

(2.3 b) If the interoperation needs had been reduced to the integration of LICEF applications in the Explora2 platform [15], the solution based on "data and service" communication and on the participation of applications in common procedures could have been combined with a structural reorganization. But when Explora became a "middleware" between several virtual campuses, resource repositories and educational application servers distributed on the Internet (in the SavoirNet project), it became obvious that we cannot ask for a profound reengineering of our partners' systems… and, therefore, the "supple" connection- is the only realistic solution. Thus appeared - in the systems' architecture (in addition to knowledge, resources, participants, activities, and support managers and to the resource controller - on which we based the distribution and the aggregation) - a "kernel" - responsible for the management of the "services bus". What was left to do was to decide the organization norms of the active interfaces (contact agents) that would allow to the applications to communicate by this bus, the strategies by which we could equip the applications and the users with these interface- agents.

This was the situation, when I was invited to materialize my ideas- proposed at the Telelearning Vancouver conference- as conceptual architect of TELOS. The LORNET project [16] aims at facilitating interoperation between instruction and knowledge management systems, educational service providers and resources repositories accessible through Internet- reducing unpleasant and costly redundancies (see a similar middleware in [17]). A "distributed pedagogical operating system's" architecture (protocols, base services, strategies that could progressively lead to the interoperability of the entities bound to the system and respecting its norms)- had to be defined. It was a major challenge, reflected in the orientation principles, which I inserted in the vision document [18].

2.4 From component indexation to global knowledge evolution management

In addition to technical compatibility, communication between (instructional) informational systems also requires semantic compatibility. Co-operative processes involving humans and documents require the sharing of meaning, depending on the cognitive relationships between the participating elements and influencing "the semantic aggregation". The knowledge is represented in various reference systems, embodied in humans and explained in messages placed on various documentary supports. It evolves within the framework of the processes it orientates. A major issue for TELOS was the choice of a formula for interlacing the processes (activities) and the evolution of the "knowledge" incorporated in persons and clarified in documents.

The correlation of the knowledge, participants, resources and activities management was the constant preoccupation of the researches at LICEF, as we can observe analysing the MISA method [2] or the organization of MOT [1] and EXPLORA [3] systems. Various principles and mechanisms were explored: the connection of the management "axes" (realized in ADISA), the grouping of knowledge, activity and "mediatic" scenarios (MOT+), the correlation of the advance in the course's structure with the enactment of the pedagogical and of the learning scenarios (in Explora). Also were explored various organization methods for the knowledge domains usable as reference systems (MOT graphs, metadata structures, ontologies [19]) and various scales of "competence" (evaluations of somebody's relation with a certain knowledge)- usable for the observation and the facilitation of the learning process [20].

I tried to combine these developments with my own ideas concerning knowledge and explanation management [21]: stressing on the bipolarity of explanation and the "distributed cognition" vision ([22]). My main interest was to use the synaptic web of the computers network to provide "explicational" retrieval and matching services: finding the available and pertinent resources (human, material) sustaining a certain competence progress on a given subject.

I therefore based the TELOS "emergent" working mode (in which the users search support persons and documents in the accessible repertories, use them - eventually producing other resources- and freely establish the operation chain) on the indexation of all elements relative to the same "knowledge domain" K- usable as reference systems. The "orchestrated" working mode also uses these references systems for indexing the function's abstract or concretised elements- relying its matching services on this indexation. The adaptable procedural models can be managed observing the competence equilibrium conditions around pedagogical operations. The optimisation computations, performed by specialized agents, are based on the declaration of explanatory capacities (comprehension, application, clarification, recommendation) of support elements.

2.5 The administration of aggregation cascades- between categories and roles

(2.5a) The last aggregation attempted for ADISA-type editors allowed the definition, with a DHTML editor (SchEd), of new DEs forms- in XML schema files. On their basis, a generator (DataEd) builds the corresponding DHTML form (for instance a new version of 104) "on the fly". With its help, the corresponding XML data island can be edited. The same technique was used (in the Link Editor) for declaring propagations that define a multi-form editor Ef.

(2.5b) We have also searched for particular aggregation formulas, available in situations when the elements to be combined are not of arbitrary origin, but have resulted from the decomposition of a coherent application. We may compose simplified versions of the mother structure, with variable geometry, but with a physiology included in that of the original one.

This last track gave birth to the GADISA "generator" (developed by C. Mitocaru). It allows the construction of Adisa-type workshops, using a battery of DExxx-g editors- as raw material. Compared to their homologues from Adisa 1 or Adisa1+, these components from the "Adisa2 raw material base" have improvements, facilitating the recombination process and increasing their flexibility: modularisation of each ED, use of a common "utility" library, separation of the data logic from the edition one, externalisation of interpretation and propagation rules etc.

The user of the GADISA2 environment choose the documentation elements that will compose the projected sub-workshop, the configuration and the aspect of each ED and the data propagation links- with a DHTML interface. GADISA2 can generate mono-editor workshops (like a(104)h etc) or complex workshops (like a(104,212,214) etc)- without data or pre-loaded with data extracted from various sources, with the help of the Adisa service server. With these generated workshops, projects can be declared and loaded with data extracted from various sources with the help of the "ADISA service bus" (that includes an adaptor service- for the case of data format differences).

The coherence of the data flow in the generated workshop is based on the fact that all the islands, editors and propagation "pipes" are selected from the pool extracted from the global ADISA system. If components come from the same "mother" source, the freedom of aggregation can be combined with the support for significant combinations - hence "*recombination" can use other formulas than "combination*". This intuition lies at the basis of the extension technique by "genetic cascades", proposed for the TELOS system (chapter 3)

All these experimentations didn't cross the LICEF frontier. In "real life", the management of aggregations, lifecycles and cascades of resource production and use is confronted with the problem of rights, mandates and regulatory norms. To whom does an object belong? Who has the right to fabricate it or use it- and in what conditions?

An ensemble of distributed objects, cooperating in the context of a given physiology and targeting certain objectives - can be partitioned on "ownership" and "rights" criteria. The last major problem that the "conceptual architecture" confront is the consideration of the "pragmatic" aspects - related to the connection of an instruction system to the administrative context in which it is used by its beneficiaries.

The MISA method allocates an important space to aspects related to the organisation principles of an institution, to the definition of "target populations", to the planning of operations and evaluation of their costs. It allows the population of pedagogical and delivery scenarios with abstract "actors" and "instruments"- that may be concretised through participants and resources chosen by the beneficiary institution. Wishing to surpass the rigidity of dedicating instruments to some categories of users (learners, teachers etc), the management of the EXPLORA platform proposes "roles"- that can be flexibly allocated, in function of the necessities. Continuing this orientation, I tried to separate the "categories" that can be used in an institution or community (defined by enumeration or by the description of certain characteristics) from the "roles" used in the functions' models- that can be played by various persons or categories.

Disappointed by the superficial treatment of the "juridical" aspects in the literature dedicated to dynamic distributed architectures (of components, of services etc) and by the slow progress of researches on "digital rights management", I adopted an "in house" formula. It operates a segmentation of the production cascades (recursive aggregation) in order to take into account the passage from an institutional context to another.

3. TELOS structure and physiology

The considerations from the previous paragraph reveal the difficulty of interlacing aspects related to: the structural logic of applications, the processual logic of operations, the logic of geographical distribution of components and of communication between them, the administrative logic of "rights", the logic of signification sharing and the logic of the recursive fabrication process. I have defined [23] the TELOS conceptual architecture taking into account the experiences related in chapter 2- as an evolving middleware, combining structural extension (through recursive aggregation and distribution) with the reproduction of procedures and with the connection of the ontogenetic cycles in phylogenetic cascades- segmented on administrative criteria.

This expanding structure will support the modelling and management of distributed instruction activities: from the emergent to the orchestrated ones. When users prefer the freedom to order (emergently) the operation sequence (resource conception, publishing, retrieval, use, annotation etc), the system offers them retrieval instruments for finding the appropriate resources (support tools and persons, previously "published" in the resource repositories): semantically pertinent, administratively available, and technically operable. In other situations, instead of loosing time to find resources and chain operations, users can rely on "aggregates" [24] edited by an author at a previous stage: "collections"-sets of resources, equipped with management interfaces, "fusions"- unitary systems composed from interdependent components, "operations"- aggregating an action, its executor, support actors and support or target resources, "functions"- orchestrating resources connected to the operations chain.

TELOS will facilitate technical and semantic inter-operation between its (distributed) users and modules and those of external systems. To do this it will use a microkernel design pattern: a "communication bus" coordinated by a "kernel" that deploys (distributes) and connects the communication agents (see also [25]), coordinating the chaining of the declare/ ask/ deliver/ receive phases- in order to satisfy remote micro-service requests. All the TELOS modules must be "pluggable" to this "communication bus". The external systems wishing to participate must be wrapped (statically or dynamically) with the TELOS interface-agents. A user asks/delivers a service through the agent-interface allocated by the kernel: directly to him (as a visual interface) or to a system that he is using. The TELOS core will also contain a general resource controller (delegating the manipulation of any resource to the appropriate handler) and an import-export module- allowing the communication with systems based on other norms.

Therefore, the TELOS "Core" (logically unified- physically being distributed on a network) contains:
1 The area of the coordinating kernel (distributor of applications and clients for remote access and of interface- agents for connection to the "service bus").
2 The area for the management of knowledge and competence reference system (operational, technical and administrative)- handlers and documents (ontologies, taxonomies etc).
3 Repositories (directories) and handlers for the management of "primary resources" (persons, tools and documents, operations)- based on metadata records.
4 Secondary resource library (interfaced or aggregated: collections, systems and orchestrations) along with the instruments (handlers) required for their manipulation (edition, use etc)
5 Handlers and libraries required for the management of the main fabrication chain of system tertiary resources (LKMS, LKMA, LKMP).
6 Systems (LKMS, LKMA, LKMP) external to the TELOS administration core, but resulting from its fabrication process.
7 Delegated "interfaces" (system agents, user agents, native application clients, TELOS clients) for coupling external entities to the TELOS communication network

Using a "4d" vision, we will talk about an extensive plastic structure TELOS(t). Only the kernel has a greater stability, while the "core" is continuously extended, enriching its central resource libraries (primary, secondary, tertiary- see explanations below) and distributing, in various external contexts, systems resulted from its production cascades

LKMS produced from core elements. With the help of specialized editors, technicians use core secondary resources as raw material (or as design tools) to build "learning and knowledge management systems"- placing (installing) them in the core dedicated library ("embedded") or in an external beneficiary's context ("separated"). As any aggregate, a LKMS can pass trough concretisation (adaptation) from the model state to the state of executable instance. These LKMS can (linked case) or cannot (autonomous case) maintain dependences trough the core. That will influence the "completeness" of their structures: starting with the "thin" cases (containing only definitions and data), going through the "fat" case (containing LKMAs produced and managed by the LKMS) and ending with the "heavy" situation (containing the handlers, for autonomy).

LKMA produced with LKMS Using an LKMS, designers can construct "learning and knowledge management applications", 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 of some core services), but it can also gain autonomy trough the enrichment with appropriate handlers (becoming "heavy" LKMA). The simplest form ("thin") of an LKMA may contain only its definition (pointing to the resources and the handlers placed in the core or LKMS libraries) - and its data (the exploration results).

LKMP produced with LKMA. The material results (traces, annotations, user constructions) of an LKMA use (accompanying learning) can be managed in the context of the generating LKMA, be placed in the libraries of the core or of the support-LKMS, or in personal portfolios (see [26]). System's feed-back loops (repositories' enrichment, resources' evaluation, competences' evolution, knowledge reference modification)- can originate from the LKMP analysis.

Apart the continuous extension of the core, the universe of the externalised TELOS products also can extends "genetically". The "phylogenetic" character of production cascades shows up following the "circulation" of a certain element. 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. TELOS practice the longitudinal management of long process chains- with the help of the function manager.

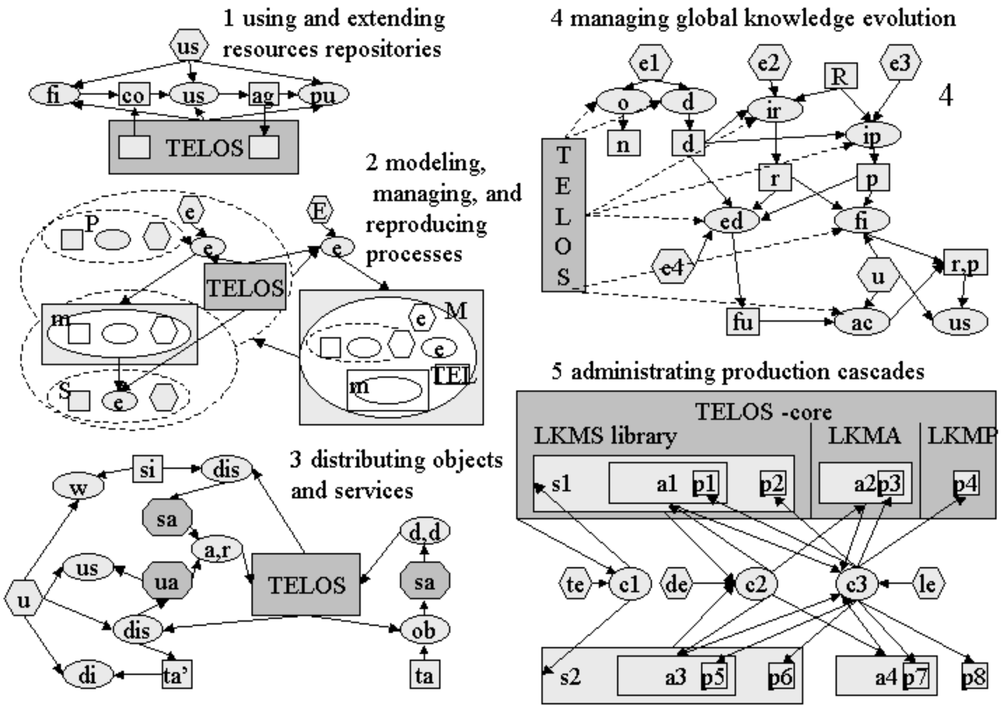

Figure 2: Five perspectives on TELOS

4 Interlaced description

Figure 2 recalls the five perspectives discussed in this paper.

1 TELOS allows the users *us* to continuously extend a resource base, supporting them in the research *fi* of the necessary components *co*, then in their use *us* for various purposes, among which the aggregation of new resources *ag*, followed by the publication *pu* in system repositories.

2 TELOS allows the reproduction of primary processes- P that inspire the edition -e of their models-m used in the production of derivate processes- S. This chain can at its turn be modelled, the metmamodel-M… being usable in its management.

3 TELOS allows to a user *us* to obtain services 1. From the target resource *ta*, through direct action *di* on replicas *ta'* distributed (*dis*) in the user context 2. Asking and receiving (*ar*) services delivered to the *ua* agents distributed to the users 3. Through the *si* systems that they work with (*wo*). In this last case, the requestor systems *si* also obtain (*dis*) interface agents *sa* similar to those used by the systems that declare and deliver (*dd*) these services.

4 TELOS supports the global evolution of a system involving knowledge, using as reference system domains *d* respecting a norm *n* (organized and defined by experts *e1*). On their grounds - the *e2*, *e3* experts - accomplish the *ip* indexation of the participants *p* and the *ir* indexation of the resources *r* catalogued in the system's repertories *R*. Before being used, the resources *r, p* can be found directly (*fi*) by a user *u* (emergent mode) or accessed (*ac*) through procedural models *fu* that are indexed- by the experts *e4*, during edition (*ed*) - in order to support the competence equilibriums.

5 The system allows the segmentation of some production cascades on administrative "discontinuity" criteria. Starting from core modules *te* technologists fabricate (phase *c1*) LKMS *s1* embedded in the core library or *s2* placed in the context of external beneficiaries (autonomous or linked). Using LKMS designers *de* can build (*c2*) LKMA placed in the core library (*a2*), the library of the LKMS constructor (*a1*,*a3*), or separate (*a4*). The use of LKMA by learners *le* modifies their competences and can produce LKMP deployable in divers contexts (core LKMP library-*p4*, LKMS product libaries-*p2*,*p6*,LKMA product libraries-*p1*,*p3*,*p5*,*p7*, personal portfolios-*p8*).

5. Prospecting the negotiation management

As I have already mentioned, the choice of a resource to use or to include in an aggregate (function, etc) depends on technical, semantic and administrative criteria. The process of establishing the rights for accessing objects, persons and activities (negotiation) may have various forms: service shopping sites, demand and offer databases, auction systems etc.

During the last year, I have studied these aspects, directing the development of a prototype, based on Java J2EE technology, dedicated to the negotiation of learning services (inline and offline support, resources allocation, access and participation rights etc). We are experimenting now the binding of this module to the others composing the TELOS1 prototype (GEFO, etc) - using Web services connections and respecting digital rights management principles.

6. References

[1] Marino O,, de la Teja, I., Lundgren-Cayrol, K. and Paquette, G. (2005).Transposing MISA Learning Scenarios into IMS Units of Learning *Journal of Interactive Media in Education* 2005(13).[jime.open.ac.uk/2005/13]

[2] Paquette, G., C. Aubin et F. Crevier . MISA, A Knowledge-based Method for the Engineering of Learning Systems, *Journal of Courseware Engineering*, vol. 2, August, 1999

[3]Paquette, G. Construction de portails de télé-apprentissage Explor@ : Une diversité de modèles pédagogiques », *Science et Techniques Educatives*, vol. 7, no 1, p. 207-226 (2000).

[4] 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)

[5] Andrade L, Fidairo J L, Gouveia J, Koutsoukos G Separating computation, coordination and configuration, *Journal of software maintenance and evolution : research and practice*, 353-369,2002,John Wiley & Sons ed

[6] Sessions R. *Software Fortresses: Modeling Enterprise Architectures* Addison-Wesley Professional; 1st edition (February 24, 2003)

[7] Fielding R, Taylor R., Priccipled design of the modern web architecture, *ACM transactions on Internet Technology*, Vol2,No2, May, pp 115-150, 2002

[8] G.Paquette, I. Rosca, Modeling the delivery physiology of distributed learning systems. *Technology, Instruction, Cognition and Learning* (TICL) ,v1, No2, 2003

[9] Koper, R., Manderveld, J., Educational modelling language: modelling reusable, interoperable,, rich and personalised units of learning, *British Journal of Educational Technology*, Vol 35 No5, pp 537-551, 2004, Blackwell Publishing Ltd

[10] Vantroys T, Peter Y: Cow, a flexible platform for the enactment of learning scenarios, *CRIWG proceedings*,Autrans, France, 2003, Springer Verlag

[11] Rosca, I, Rosca V. Pedagogical workflow management with functions, LOR'04 congress, Montreal, http://www.lornet.org/i2lor/pps/Rosca.pps, 2004

[12] Garcia-Cabrera, L, Rodriguez- Fortiz, M.J., Perets-Lorca, J. 2002. Evolving hypermedia systems : a layered software architecture. *Journal of software maintenance and evolution : research and practice*, 389-485, 2002

[13] Lundgren-Cayrol, K., G. Paquette, A. Miara, F. Bergeron, J. Rivard and I. Rosca (2001). Explor@ Advisory Agent : Tracing the Student's Trail, WebNet'01 Conference.

[14] Dufresne, A. The exploraGraph advising system : an ergonomical evaluation of the editor, TICE'2002, Lyon, France.

[15] Rosca, I., Paquette, G.: The Explora2 system, Congrès TeleLearning, Vancouver (2001)

[16] Rosca, I., Paquette, G.: TELOS research progress, LOR'04 Towards the educational semantic web, Vo1, No4, Dec (2004)

[17] Lee Y, Chong Q.: Multi-agent systems support for Communities-Based Learning, *Interacting with computers* 15 , 33-55, 2003, Elsevier Science

[18] I.Rosca, G.Paquette., System orientation principles, TELOS Vision and Orientation http://www.lornet.org/docs/telos.pdf, pp26-28 (2003)

[19] G Paquette, I Rosca An Ontology-based Referencing of Actors, Operations and Resources. *eLearning Systems SW-EL*, 2004

[20] Paquette, G. (2002). Modélisation des connaissances et des compétences, pour concevoir et apprendre, Sainte-Foy, Presses de l'Université du Québec.

[21] I.Rosca, Towards a systemic vision of the explanation process; story of a research on integrating pedagogy, engineering and modeling PhD thesis, Montreal, http://www.ioanrosca.com/education/these, 1999

[22] Hollan J., Hutchins E, Kirsh D, Distributed cognition: toward a new foundation for Human-Computer Interaction research, *ACM Transactions on Computer-Human Interaction*, Vol7 No2 ,174-196, 2000

[23] 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)

[24] Rosca, I., Paquette, G.: Organic Aggregation of Knowledge Objects in Educational Systems, *Canadian Journal of Learning Technologies*, Volume 28-3, pp. 11-26 (2002)

[25] Cianacarini P, Rossi D, Vitali F., Designing a document-centric coordination application over the Internet, *Interacting with computers* 13 677-693, 2001, Elsevier Science

[26] MassonR Pegler C Weller M: E-portfolios: an assessment tool for online courses *British Journal of Educational Technology*, v35, no 6, 2004