Adaptable models for procedure orchestration and competence matching services

Ioan Rosca

Researcher and architect LICEF, Teleuniversity of Montreal, Canada
Telecommunications, information and instructional systems engineer
ioan.rosca@licef.teluq.uqam.ca

Abstract. While developing the GEFO prototype, I explored the fluid passage from modelling process to orchestrating and reproducing them - proposing the "function" formula. The services obtained by their use (information, declaration, facilitation, coordination)- valorise the efforts made in the edition phase. They can also sustain the matching of relevant elements, available at execution time, based on the semantic indexation of potential participants and resources- relative to knowledge reference systems. The definition of "postures" allows for the formulation of "competence conditions" around activities- seen as competence operators. Using metafunctions we can globally manage the procedure-model loop and the "lifecycle" of functions. The selection services- used by the authors of the orchestrating scenarios- during the gradual concretisation of the elements-use "competence equations". But each concretisation of an entity or action… modifies the downstream equations. The distributed agents following the equilibrium must (cooperatively) solve a chain of optimisation problems, in continuous redefinition.

Keywords: procedure modelling and orchestration, function and metafunction, competence indicators, conditions, services and agents; explicative matching; optimisation problems for the progressive concretisation of models, GEFO

1 Introduction

Figure 1 map the ideas flow of this paper- for the orientation in the problem-space approached by of the research synthesized in it.

We start (in chapter 2) with the problem of process representation. After the exposure (in chapter 2.1) of the “4d” ontological vision assumed in the text and the modelling technique used in the GEFO prototype (2.2), we present problems and proposals related to knowledge modelling and its evolution (2.3) and to the modelling of the explanatory co-operation (2.4).

We then pass, in chapter 3, at the reciprocal problem of the models' use for the implementation of actual procedures After the presentation (3.1) of the need for bundling multiple mechanisms for the orchestration of man-machine groups- using communication and co-action, we describe (3.2) the formulas centred on the pragmatics of the services realised in GEFO. We then propose a manner for preparing the models in order to support the "matching" of the relevant components at execution-time- a facility of the "orchestrated working" method corresponding to the retrieval mechanisms of the "emergent" one.

Chapter 4 studies the global physiology of the system formed by the process-model loop. The "meta- functions" technique is exposed, then the mode in which it allows the reproduction of procedures (4.2). The technique of adapting instances through a progressive concretisation of the elements, controlled by "competence equations" (4.3) emphasizes the "conceptual web" exposed previously (the management of the evolution of competences, the use of the functions and meta-functions, etc) The problems of global optimisation problems enounced at the end (4.4) also take advantage of the holistic vision adopted and reveal the interest for the use of "agents"- which cooperatively solve the web of evolving problems of operational research problems, created by the desire of optimising competence management.

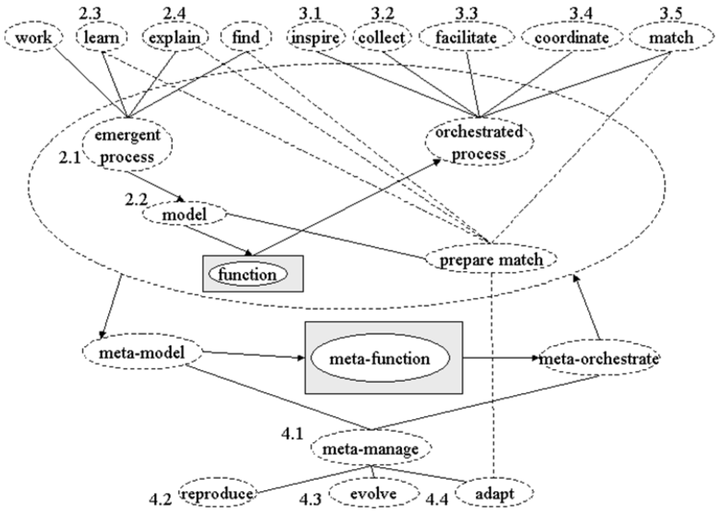

Figure 1 : Map of the problem space approached in this paper

2 Processes and their modelling

2.1 Structure and process, existence and becoming, 4d ontology

As Mizocouchi signals [1], each of us have his own position face to the primitives of thought: space, time, matter, entity, relation, conscience etc. For the partisans of a "4d type" existential vision, the "entities" and the "concepts" that reflect them- evolve continuously- on a "trajectory" that determines their flowing existence. That is why the dynamic of systems- on one hand- and their history- on the other- cannot be separated from their essence. The interest for the general theory of systems (processes) [2] and my experience as telecommunication, information and instruction system engineer have led me to a holistic "4d" vision..

What attracted me in the pioneer works of the (cognitive) living theorists, like Varella and Maturana [3] is the stress put on the integrity of the "autopoetical" systems (having the morphology continuously re-modelled by a physiology dedicated to the identity conservation). I have reinforced, in time, as engineer of electronic regulatory loops, the bio-cybernetic vision initiated by the meeting with Odobleja's [4] works.. Thus, I observed the physiology of the total system, formed by a primary external reality (objects, persons, process), its internal reflections (evolutionary cognitive spaces), the external reflection of the conceptual spaces (language, reference system, representations, models, messages) an so one…An infinitely recursive game…

2.2 Primary process and processes of its modelling and communication

The problem of mixing process and structure modelling can be tackled in various modes (see an example in [5]). Resorting to biological analogies, I used the term "function"- to designate the reproduction formulas of the procedures on the basis of active models. The specification of the "physiological" colour given to the word is necessary for the comparison with the "functional representations" used in technical fields - when the interlacing between structures and process - around a teleological axis- is desired (see examples in [6] and an analyse of these representations in [7]).

Process modelling was approached in a multitude of domains - with very diverse agendas and methods, rather difficult to coagulate. Some approaches treat the reflection of the processes for a better comprehension of complex phenomena [8] and socio-technical systems [9]. Others seek the production of new process, driven by "scenarios" [10]. Some use them in the orientation of the production of support instruments ("use-cases" case- as in [11]). Others use them as execution support (operation sequencing, co-operation coordination, resource handling, etc (as [12] see also cap 3).

Some (task/action oriented) domains are interested in the management of productive processes [13]. Others are axed on the reflection and influence of cognitive processes [14]. Another category studies communication or collaboration processes [15]. The educations sciences synthesise these preoccupations, applying them to instructional processes. The organization processes of informational and retrieval spaces [16] are also examined in several domains (information science, computer science, AI, etc.) We can approach "short" processes (collaboration relations, the ontogenesis of object fabrication), "long" processes (objects lifecycle, systems evolution -[17], “phylogenetic” production cascades) or even interlaced processes, forming the physiology of complex systems.

The participants to processes being persons, objects and "computer agents", the communication and co-operation man-man, man-machine and machine-machine problems interfere, leading to representations addressed to human or machine interpretation and to a range of problems that goes from the ergonomics and semiotics of perception and human dialogue to the definition of some inter-operation norms between technical systems.

The difficulties encountered in the coagulation of a unitary model of the explanation phenomenon starting from the vast and dynamic scientific activity signalled before, determined me to choose a pragmatic approach of the procedure-modelling problem. Working on the Explora2, SavoirNet and TELOS architectures [18] I compared the LICEF pedagogical workflow (learnflow) modelling formulas with similar developments coming from CSCW (or CSCL) analysing on one hand- the human logic (and ergonomics) of the representations [14] and- on the other hand- the inter-operability of the resulting data, based on norms like EML or IMS-LD [19].

Concluding that there are natural similarities between the different representation conventions, I adopted the MOT symbolization (with some nuances), proffering to concentrate on the problem of the pragmatics of representations' use. In order to deepen the research on the physiology of the ensemble formed by the procedural reality and its orchestrating model, I have piloted the prototypal development (by Val Rosca- whom I wish to thank) of a "function manager" [20]. This instrument was then used and refined in the context of the LORNET project [18].

Figure 2 summarizes the formalism and the method used in GEFO:

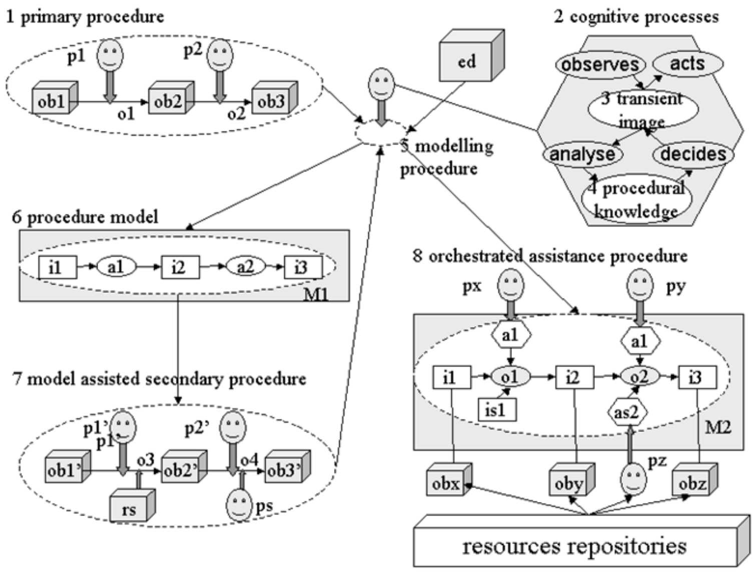

Figure 2: Processes and their modelling …processes

A primary procedure (1) has a dual character, as "structure in process". Structurally, it is formed by interconnected "components": persons (the executants *p* of certain actions and their assistants- ps) and objects (resources *obs*- to be used or produced- or support resources *rs*). Processually, it consists of a chain of operations (steps, phases - *o*). Combining the structural and processual approaches we arrive to decomposition in interlaced "threads" (roles). Procedure aggregation can continue in both directions: the actions can break up in sub-procedures and the procedure as a whole, seen as unitary operation, can be part of an including meta-procedure. Someone can participate to a real procedure (as the one suggested by that dotted oval) or may wish to refer to it, using a symbol, a name or a model.

The observation of a procedure (by a participant or external observer) is accompanied by **cognitive processes (2)**: observing and reacting, analysing and deciding etc) leading to a certain evolution of the **transient image (3)** and finally of the **procedural knowledge** (4 : memorised image). These cognitive processes can themselves be represented procedurally, if their exposure is useful.

The modelling of a procedure (5)- is also a process, which creates a symbol structure M1 (or M2), reflecting, or imagining a concrete (with specified elements) or abstract (with generic elements) system-in-process. It uses representations for the components reflected in the model "mirror": actors (hexagons) - which can designate generic participant categories or specified persons, instruments (rectangles) - which can designate concrete resources or generic classes, operations (ovals) - designating particular or generic processes, realized or to be realized. Some models can comprise other symbols too, such as those dedicated to the possibility of "branching" (ordering) the actions, depending on data obtained in a concrete realization of the procedure.

To assist the execution, to present or teach a procedure - even a simple model of the operation chain can be useful. The participant using a model as an inspiration tool (**model assisted secondary procedure-7**) can interact with support resources and persons, although this assistance is not planned in the model. The *pedagogical management of a procedure* is a flexible solution, but it can create organization difficulties (finding support etc)

The orchestrated assistance procedure (8-*pedagogical procedure's management***)** supposes the explicit representation, in the model, of support actors and instruments, reducing the freedom of the pedagogical arrangement, but assuring the conformance to the pedagogical intentions of the model's author..

2.3 The management of the invisible part - the "K" level

The previous paragraph points to "the invisible part" that accompanies the processes in which humans cooperating and communicating through messages placed on various supports intervene: the knowledge processing. Concepts evolve together with the persons incarnating them. The document space (involving or explaining concepts)- is also continuously changed. Activities produce modification of knowledge incorporated in the involved persons and resources. The "knowledge domains" can be used as reference systems (a concept being identified by its address). But such "coordinates" are not sufficient to observe and facilitate learning. The characterization of someone's relation to a concept requires "competence indicators" [21]. The indexation of persons, documents, and activities on a common reference system can produce a "semantic aggregation" [22].

The approach of the knowledge management and the representations' granularity depend on the goal pursued by their explicitation. Prior to cooperating or communicating, the partners must equip, find and agree with one another. The computer network infrastructure can provide contact, contract and management services, forming a "synaptic" infrastructure for the collective brain's physiology.

Activity coordination systems elaborated in CSCW (CSCL) could be enriched with matching facilities so that they facilitate the retrieval and the selection of the participants that can perform (optimise) the ongoing operations' chain. The developments in "distributed cognition" ([23], [9])- stress on the fact that that the individual cognitive metabolism is "situated" in that of the community's. The passage from the “semantic Web” vision to a “pragmatic Web” approach (explained in [24]) represents a shift of the attention from the knowledge representation structures to the physiology of their use- in the context of community cognitive evolution..

2.4 Cooperative explicative (support) processes

As we have seen in chapter 2.2, the interest for assisting the procedure execution (see chapter 3) interfere with that for representing assisting procedures. But the concept of "assistance"- covers a large range of significations. The advanced "support" systems allow the combination of these possibilities, the choice being adapted to the users' needs. After many years of practice and meditation, I have reached the conclusion that the concept of *explanation* can facilitate the fusion between instruction and assistance and that the foundation of the "instructive-productive management" should be a unitary theory of procedure modelling- including material and cognitive aspects. I have exposed the objectives and principles of such a science in my doctoral thesis [25] trying to conceive a model for the (instrumented) explanation phenomena.

The problem's complexity forced me to resign myself to elaborate partial models (morphological and physiological), to structure a "map of my perplexity" and to enounce principles that have subsequently guided my research. From those, the observation that "explanation" is based on the cognitive consonance lived by a human pair- is crucial. Synchronous or asynchronous, realized through communication, sharing or co-operation- the explicative relationship between an "expert" and a "novice" is essentially a bipolar phenomenon based on the collaboration between two decision centres, involved simultaneously, jointly, asymmetrically.

The difficulties of modelling the individual cognitive processes are amplified in the case we want to describe the phenomenon of explanatory co-operation. In the case of "communication" [15] intervene a number of delicate problems studied in psychology ,cognitive sciences, communication and information sciences, semiotics and multimedia, logics and epistemology etc. In the case of “co-operation”, the subtle dialectic of the circular relationship between doing (to know) and knowing (to do) is added- and studied in the science of education, action theory, negotiation and decision theory, CSCW, etc (an example in [13]). This allows for the didactics of the work in double command: the explanation of a procedure through the sharing of an action :the expert E does because he knows, the novice N knows progressively- because he is helped to do. The modelling and management of distributed triangular actions (expert- computer- novice)- which I explored in a series of projects (“Triple controlled explanation”, “TaxiNet”, etc.) proved even more difficult.

It deepened my interest for the orchestration of co-operative procedures.

3 Orchestrating procedures with their models

The problem of procedures' management (orchestration, coordination, support, reproduction) cannot be separated from that of their modelling, even when the models are not directly used as instruments in their realization. On one hand- because the opportune intervention over a complex phenomenon requires its comprehension. On the other- because the intervention is, in its turn, a "modelisable" process… But I will not enter yet in the meander of these recursive relations (treated in chapter 4). In this chapter, we will concentrate on the models used as active management tools.

The solutions come from fields traditionally interested in the orchestration and management of co-operative procedures involving objects, people and computer agents (CSCW - [12], CSCL-[19], etc), DSS [26], CE-[27] etc.).

A necessary fusion is that between process modelling and model use. As the studies dedicated to the "enactment" of scenarios [28] show, the passage of a model from the posture of passive reflection of a phenomenon in that of an instrument for its coordination is not at all a trivial step. I faced this difficulty while studying the problem of transforming MOT (a LICEF-conceived editor for the management of procedural knowledge, pedagogical scenarios and resource diffusion plans [29])- towards the posture of a collaborative editor for cooperative procedures' orchestration scenarios. Unsatisfied by the separate treatment of "modelling" and "orchestration", I adopted a fluid approach, organizing the progressive transition of a graph model from the hypostasis of phenomenon image to that of interface for its implementation (coordination).

The "function" mechanism which coagulates experiences condensed in concepts as: "workflow", "flowchart", "task model", " scenario", "floor control"- in a coherent formula for the orchestration of mixed ensembles formed by people, objects and computer-agents - operating on the basis of knowledge. The fundamental criterion that I adopted is pragmatic: the development of multiple services for procedure assistance, which should bring significant benefits with reasonable preparation efforts.

The following facilities resulted, explained by extracts from the respective use-cases (from the LORNET documentation)

3.1 Inspirational function exploration

A user observes a function that inspires him for producing an analogue procedure.

Use case 1: "**1** A user opens an instance of the function **2** He observes the visible graph and navigates in the sub graphs **3** He obtains graphical details of some (all) nodes (fish-eye facility) **4** He may consult the accessible attributes of a node (for example the involved knowledge and competence) **5** He may open some documents attached as information resources **6** He executes some of the represented operations, acting in the real world (parallel to the function) **7** He closes the exploration process

3.2 Declarative function exploration

A user declares advancement on a representative function.

"**Use case 2:** "**1** A user opens a new instance of the function **2** (He can do all the actions described in Use case 1) **3** He successively declares the executed operations and acts accordingly in the parallel world of the reproduced procedure **4** He observes the function reaction (acceptance– if the function chain is respected or refusal- if not) **5** He may obtain task completion advices automatically, when an elementary or composed operation is declared **6** He may be obliged to answer some prepared questions before continuing the advancement. These evaluations may verify the competence suppositions **7** He may ask for some task advices before executing the next operation **8** He may declare some details about real operations (notes edited as instance attributes) **9** He may edit some attributes for a source-operation that are propagated and observable in a receiver-operation **10** He may publish task related documents as attached resources to an instrument-node **11** He produces a trace of his advancement, that he (or another acceptable user) can observe at a later date **12** He can analyse the advancement in other instances of the same function **13** He may manually rollback (undo) some operations, restricted by the chain's logic **14** He closes the exploration process

3.3 Facilitating function exploration

A user (partially) controls the execution of a procedure, using the function as an interface for manipulating some resources

Use case 3: "[] 3 He can launch some resources, connected to the function as target instrument concretisations and controllable with functions or through their own interface. They may be: - server resources with client interface - local resources, downloaded "on the fly" or in a previous phase by the resource controller used by the function **4** He can continue to manipulate an open resource using its own interface and manually declaring the advancement on the function graph. **5** He can manipulate some resources using their own interface and intercepting the actions at the function level, thus obtaining help from the monitoring agent **6** He can manipulate some open resources using the function interface (asking the execution of some prepared scripts, thus obtaining help from a manipulating agent) **7** When he launches a "collector resource", he exploits the propagation of components and parameters from a "source-resource", realizing the dynamic aggregation of the resources **8** The function may automatically launch some script executions if the advancement conditions are met (help from a self-controlled manipulative agent) **9** He continues the manipulations (steps 2-9) until the exploration is finished or postponed for a later session **10** He may undo some machine operation declarations; the rollback of the actual operations is only done on rollback-prepared resources **11** He closes the exploration process."

3.4 Coordinative function exploration

Several users coordinate themselves in the execution of a cooperative procedure described by a function

Use case 4: "**1** A team member opens an executable instance of the function **2** (He can do all the actions described in Use cases 1, 2 and 3: explore indications, concretise components, assist the executor, execute operations in a private world, declare advancement, manipulate resources – if appropriate rights are satisfied) **3** Instance sharing. Other team members (co-executors, assistants, observers) open exploration sessions on the same function instance. Every participant has his own graph exploration window and can act (to reproduce the represented procedure) in a private collateral world **4** Navigation. The inspirational exploration (sub graphs, details, attributes, guides) of every member (observer included) is not restricted by the other sessions (with the possible communication of the others' position in the sub-graph tree, used as an awareness tool) **5** Resource concretisation. A guiding document or a helping agent connected by a participant to an instrument node (as a resource concretisation) is propagated to the coordination server and may be consulted (used) by the other participants (document and transferable resource sharing) **6** Shared advancement. For the untied operations (without functional manipulation of resources), a legitimate executor can declare advancement on his navigation window. The accepted declarations are propagated to the other participants. The generated advices are visible only for the declarer. The required advices are adapted to the common advancement. **7** Actor (executor and assistant) concretisation and floor control. If the coordination mode is set to 1 ("token request") an executor/assistant can not declare/support an execution, even if he is on the approved list for that operation, before asking for floor control (declaring the intention to execute/support, reserving, concretising the actor to himself) Observation**:** Don't confound the floor control between several legitimate executors, with the floor control between operation members (executor, assistant, agent) and the floor control between operations (chain logic) **8** Auditing. The instance common trace reflects the contributions of the participants (executors, assistants) to the advancement. The active participants and the observers may consult it. **9** Resource sharing. For the tied operations (with connected resources), the manipulation can be launched by every legitimate participant (mode 0) or by the one who has received the floor control (mode 1). Sometimes the resources are local to the participant. In other cases, they are on the server and several executors (or assistants) can manipulate the same function-controlled resource (see use case 3), successively launching prepared execution scripts. **10** Communication between team members. A participant may introduce some local information into the attribute fields of a node. This information is accessible to the other participants, representing an asynchronous message (task related annotations). If several users share the same attributes window, they can communicate synchronously (internal node chat) **11** Support. The assistant of an operation can communicate, synchronously and asynchronously, with his assisted executor. The support may consist of a resource concretisation (document or agent). It may also consist in the demonstrative manipulation of the operation target. **12** Mixed cooperation (assistant and agents). If the conditions for an automatic (agent) execution are met (see use case 3), the scripts are launched and announced to all the active participants. Special flow control rules may apply for triple actor operations (executor, agent, assistant) **13** Signalling jams. If the operation chain cannot be continued because of the absence of an executor or assistant, the function may make the appropriate invitations (inline signalling, sending emails etc.) **14** The steps 4-13 are repeated by an active participant until he decides to finish his session **15** When the last active participant finishes his session, the instance is closed; it may be reopened and continued at a later time **16** After the completion of the chain, the function responsible may close the exploration of the instance

3.5 Matching function exploration

The function assists the connection of the appropriate participants and resources for an instance execution

Use case 5: "[] 3 **Unsupported participant (executor) concretisation** (mode 2, operation O1). When a participant wants to reserve an execution (declaration, manipulation) for an operation without support (actors and instruments), he may ask for an analysis of his competence legitimacy, to see if he has sufficient competence for every knowledge attached to the operation :$p>=o$ **4 Resource assisted participant (learner) concretisation** (mode 2, operation O2) When a inscribed participant (with a competence p) wants to reserve an execution for an operation (competence o) with planned execution actors (competence a) and concretised support resources (capable of assisting an i-f competence shift), he may ask for a competency analysis to see if: $i<=p=a<o<=f$ (the participant has the desired under-qualification and can operate using the assisting resource); $i<=p=a<f<o$ (the participant has the desired under-qualification but the resource provides insufficient assistance); $i<f<=p=a<o$ (desired under-qualification but the resource is useless inferiorly); $p=a<i<o<=f$ (has the desired under-qualification but can not approach the resource); $p=a<i<f<o$ (desired under-qualification with insufficient and inapproachable resource); $p=a<o<=i<f$ (desired under-qualification with superior useless resource); $i<=p<a<o<=f$ (the participant has extra-under-qualification but can operate using the assisting resource); $i<=p<a<f<o$ (extra- under-qualification and the resource provides approachable yet insufficient assistance); $i<=p<f<a<o$ (extra- under-qualification and the resource is insufficient); $p<i<=a <o<=f$ (extra under-qualification that makes the resource inapproachable); $a<i<=p<o<=f$ (reduced under-qualification that allows the utilization of a theoretically insufficient resource); $a<o<=p$ (the non-desired qualification of the participant allows for a non-assisted execution) etc....**5 Human-assisted participant (learner) concretisation** [] **6 Abstract assisted participant (learner) concretisation** [] **7 Support concretisation for a participant** []8 **Support concretisation for an abstract actor []** 9 **Multiple assisted operations** (mode 2, operation O7) It is possible that multiple resources (or human assistants) be necessary for ensuring a competence condition for an operation. In this case, after every support concretisation, the competence equations are modified. The general condition became: "the reunion of the (i-f) intervals of all the support tools should contain an interval that contain the levels o and p" 10 **Using the resource support** After the reservation of a resource supported operation, the chosen participant may use the supporting resource (read a document, watch a demonstration etc) before executing 11 **Using the human support** After the reservation of a human supported operation, the chosen participant may communicate asynchronously with his assistant, before executing 12 **Executing an operation** The execution is possible for a participant who has the right to act (mode 0) or has obtained the token (mode 1) for the reservation (actor concretisation). It may consist (see use case 3) only in a declaration (the real resource is not connected to the function) or of a function-interfaced manipulation. It may depend on some planned support (learning situation) and on the accomplishment or on the support conditions (assistant and guide concretisations). It may be partially done by a support actor (assistant o agent) []"

4 Meta-process of process management

4.1 The reality-model loop and metafunctions

The time has come to observe (see figure 1) the loop formed by the two relations established between a reality and a model, with major behavioural consequences. In chapter 3 we saw that the modelling (description) of manifested or imaginary phenomena can orientate future implementations in accordance with certain values and goals. In chapter 2 we mentioned that the reality must be observed and understood (modelled)- even if we wish to conserve it or equip it.. When the phenomenon's "model" is used as an instrument by its participants, the reality and the model form a global system, the physiology of which deserves being understood, modelled and optimised.

The importance of perceiving this holistic and cybernetic reality is shown by the increasing interest for problems like: reflection of community life [8], discovery and instrumentation of their creation or physiology [30] "narrative" descriptions and "scenario based designs" ([31], [10]) evolving systems ([32], [17] and "shared understanding through cooperative design" [33] etc.

As I signalled in par 2.2, the evolution of the reality- model relationship is also a process, belonging to the class of long-term evolutions (also see the considerations in [17]) The modelling of "short" processes involving several entities in inter - relation, constituting "events" of community life, can be complemented with that of the long process of evolution of the entities and structures ("lifecycle") and of the complex process blend defining the "physiology" of the system. Another level at which procedural representations can occur is that of the "phylogenetic" chains - that bind the cycles of production by which a system is extended: an object made in a process being used (like instrument or raw material) in another.

We therefore are in a recursive situation: we want to observe and possibly influence (orchestrate) metaprocesses - with variable demonstration formulas - through which a series of process that occur are modelled or are implemented using the model. The key idea of the GEFO management is the "resort to the method"- through the use of "metafunctions".

We arrive this way to the following "hypostases" (forms) of a process P's representation: *operations* o(P) (which describe the basic actions and the involved elements), *functions* f(P) (which describe the decomposition in phases and roles) and *meta-functions* F(f(P)) (which describe the relationship between a real procedure and its model).The synthetic "operation" form is useful in model management, being treated as any other resource: indexed, placed in "operations" repositories, retrievable, executable or usable as composition element in procedural aggregations (functions). The "metafunction" form establishes the reproduction formula for the process P.

The coupled use of functions and metafunctions is interesting for the engineering of the evolving chains P-F(P) as suggested by Garcia-Cabrera & others [32] who organized the observation of the evolution for the conceptual, presentation and navigation sub-systems- in parallel with their metamodels. The price paid for adopting this "circular" methodology is a certain epistemological (methodological, organisational) perplexity- but it is exactly this cybernetic dialectic which opens the way to refined and fertile strategies for ascending the research-development-application spiral.

Observing the last "use cases" (6-10) conceived for the GEFO manager (and demonstrated at various occasions) - dedicated to the edition of functions, we can observe both the advantages and the complications brought by the recursive treatment The edition of a function based on the observation of a demonstrative execution combined with the capture of some actions (through sensors placed in the space of the modelled phenomenon) - must continuously refer to the execution. The edition of a function controlled by a metafunction also has an execution character- for the used metafunction. And the edition of a metafunction by capturing the demonstrative edition of a function- posed serious problems, not only to the development of the prototype, but even to the conception of the respective use-case...

4.2 Object and process (re)production

Procedures can be "aggregated" in several manners, in order to compose increasingly complex procedures. At the basis of the connection of procedural chains is the concatenation between two operations o1 and o2: simple (temporal, precedence), mediated by the transmission of a parameter, or realized through the use in o2 of a resource produced in o1, or through the implication of the same actor in both procedures. We can obtain chains of any length by repeatedly coupling the (orchestrated) production of systems that prepare the (orchestrated) production of systems that … A special type of relationship is established between a procedure and the metaprocedure that controls it. The representation of the meta-process of reproducing procedures by modelling them and using these models to create more or less similar phenomena (procedure "phylogenesis" see figure 3)- is the key of the GEFO prototype's use in the management of the TELOS system.

1. Modelling. A primary procedural phenomena P occurs 2. The procedure P is observed (imagined) by the designers, who edit its model (function x) (and concretize the resources and participants). 2a. The "interception" of the normal or demonstrative actions (with the help of some "captors") can sustain the generation of pre-models, continued by human edition **3.** Reproduction. The primary phenomenon P is reproduced in a number of secondary phenomena S, through executions of the model (see inspiration, declaration, manipulation, coordination and matching facilities- chapter 3) 4. Meta- modelling. Observing (imagining) the primary process 2 of the model's editing (or the 1-2-3 chain of procedure reproduction with the help of the model), some technologists can edit meta-models for explaining or supporting the modelling process 4a Observers, demonstrators or captors watching the edition process can generate metafunction pre-models. **5.** Meta-reproduction. Using edition meta-functions the process 2 of function editing can be reproduced (with variations) in secondary editing processes- producing functions usable in the reproduction processes x- step **6**. **7** Meta-meta-modeling. Observing (imagining) the process 4 of the metamodel's editing (or the 1-2-4-5-6 chain of manging the procedures reproduction with the metafunction), some engineers can edit core-functions explaining or supporting the metamodelling process **8** The editing metafunction may be modified (re-edited) using the core function

An so on….

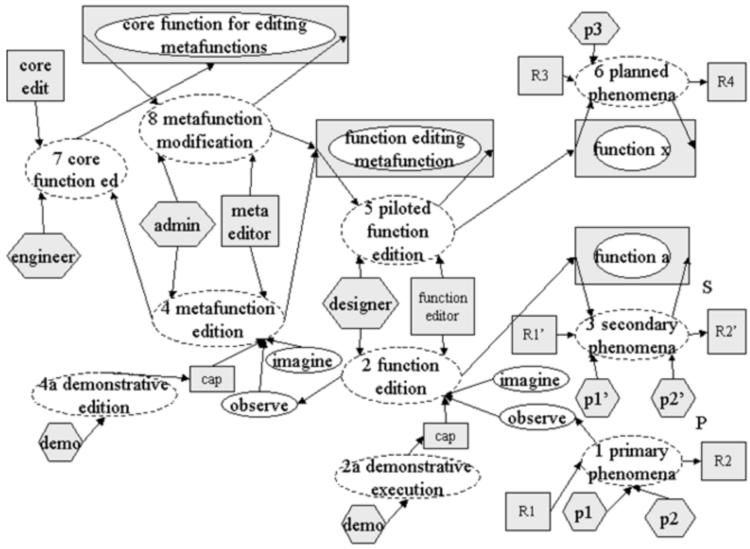

Figure 3: procedure reproduction

4.3 Lifecycle and life mode in emergent and orchestrated processes

Functions can also be used for describing and managing the "lifecycle" of resources with various structures (extensible "collections", adaptable "fusions") and with various evolution formulas. A procedural model can describe the "emergent" establishment of the process chain or establish a scenario for its actualisation. In this last case, the emergent evolution of the modelling-orchestration chain can be described with “metafunctions” (that can control similar modelling cycles)..

A typical sequence for the emergent mode is: 1 **Publishing.** The participants having this right (mandate) declare new resources (documents and persons) in the appropriate repositories. 2 **Retrieving**. The resource users exploit the retrieval instruments. Some facilities for adapting the selection according to the users' competences may also occur 3 **Cascading.** Operations are chained freely, according to the users' necessities (initiatives). 4 **Reacting.** As a result of using some resources, data (traces or annotations) can appear.

The evolution of the "orchestrated" mode is explained above:

1 Edition A real procedure is observed by an author, who conceives a model based on it, using a "function editor". The operations and elements (actors and instruments) are declared abstractly, allowing liberties for further concretisations, in the limits specified by the author (expertise required for the assistants etc) **2 Concretisation.** The function definition process can be continued, starting with the base (class) model. Concretisation can mean specifying the final components or just restricting the selection criteria (administrative, technical, domain) for the connectable elements. An arborescence of increasingly particular "derivate" models can be obtained this way. **3 Execution.** Is accomplished according to the scheduling or after free instance retrieval. The execution's results (data, annotations, traces, produced resources)- are stored aside. **4 Reaction.** Is based on result analysis and can include the modification of competence profiles and resource indexation or even the re-organization of the knowledge reference system.

The meta-procedure exposed above is treated as a metafunction (figure 4 is a print screen of a demonstration made with the GEFO prototype) That gives us the possibility to define and manage the "life mode" of a function that characterizes the liberty space of this derivation process -conforming to a *system evolution typology*. Using metafunctions we also can manage the concretisation cascades that lead from an abstract model to the realization of actual procedures. On this occasion, we can also treat the problem of organizing the semantic matching services.

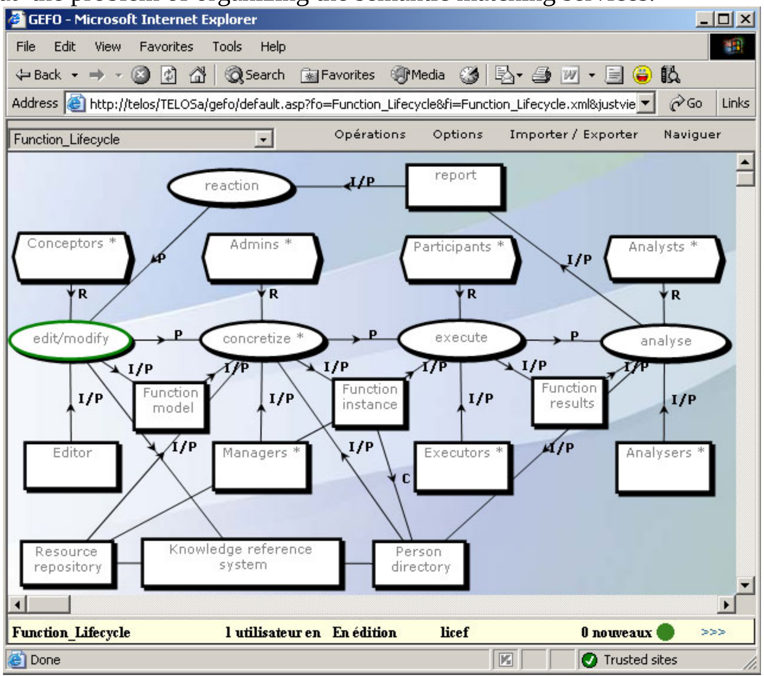

Figure 4: management of a lifecycle with GEFO metafunctions

4.3 Matching services for progressive concretisation

The use case reproduced in chapter 3.5 signals the various situations that can appear during the exploitation of a function, around each operation, according to the topology of that respective operation (executor or groups of executors, support persons and documents). When the execution of an operation occurs, several mechanisms can intervene (access filtering, analyses, alerts, etc) supporting the optimisation of the competence evolution (knowledge aimed and obtained through execution). The "agents" techniques may be applied for resolving this kind of problem (examples in [34] [35] [36] [16] [30])

If concretisations liberties are allotted to the execution phase, refined matching services trough the available resources can come into action- allowing to the "adaptable orchestrated" mode to fructuously combine the advantages of using "scenarios", of resource finding liberties and of the use of the computer network synaptic (based on semantic indexation). These mechanisms and "matching" facilities are similar to those available to the editors involved in the progressive edition of functions. The considerations that follow therefore apply to the whole chain of concretisation.

The agents that must solve "the competence equilibrium optimisation" around an operation have a difficult mandate. They must be able

- to deal with problems like: the use of the several norms for the organization of knowledge and competences references, the "merging" of two ontologies, the modification (versioning) of the reference ontologies to adapt to the granularity of concepts optimal for a given situation and to solve the problem of the connection between competences relative to a certain concept and those relative to others with which it is related in an ontology (particularly relative to its sub- concepts).
- to cooperate, in order to find out what are the availabilities at a given time, adapting its recommendations to the current situation
- to be able to work in "planning" mode (on the basis of appropriate information) and to ensure the reliability of the transactions on which the realization of a given scheduling depends
- to consider (eventually providing means for verifying presumptions) the fact that a generic operation O that envisages a competence progress (C1- supposed level, C2- aimed level) is a vague (statistical - probabilistic)"operator"- passing the real participant from a level c1 to c2 - even when all the requirements are met.
- to solve a wide range of optimisation problems (according to various criteria and topological situations)
- to "commutate" the formulation of the problem according to the modifications appeared in "upstream" (concretisation of certain elements replacing the abstract instruments).

Even if we resume ourselves only to the last aspect, we obtain a fearsome problem (see explanations in [6]) - as figure 5a signals - in which the empty figures represent abstract elements: the planned operation - O, the imagined executor - E, the projected support document - D, the human assistant- foreseen - A and those full representing actualised elements (o, e, a, d)- having real competence characteristics (c-1 supposed for the assisted beginner, c2- aimed) more or less close to the planned ones (C1, C2). "The competence equations" will depend on the concretisation order, multiple

alternatives for going through the "state machine" that progressively fixes the topology being possible:
- O, OE, OEDA, OEDAa(d), OEDAad, OEDAead, OEDAoead
-O, OE, OEe, OEDAe, OEDAea(d), OEDAead, OEDAoead etc

The rules (equations) that intervene depend on: the procedure's "topology" (Toead= operation, executor, assistant, supporting document, Toea, Toed, Toe, etc.) and on the position of the enactment in the concretisation chain. For example (see the hardened line in figure), for an operation *O* requiring a competence level C1O, approached by a real learner *e* having a competence c1e, supported by a concretised assistant *a* capable to sustain c1a- c2a increases and by a real document *d* capable to sustain c1d-c2d evolutions, we can observe situations as: *(c1a<=c1e<C1O=c2a* or *c1d<=c1e<C1O<=c2d-* any support component is sufficient), or (*c1a<=c1e<c1d<c2a<c2d<C1O<=c2d)* - the assistant can lead the executor in the document's efficiency range).

However, as I have said, the concretisation of *O* in *o* can bring surprises….

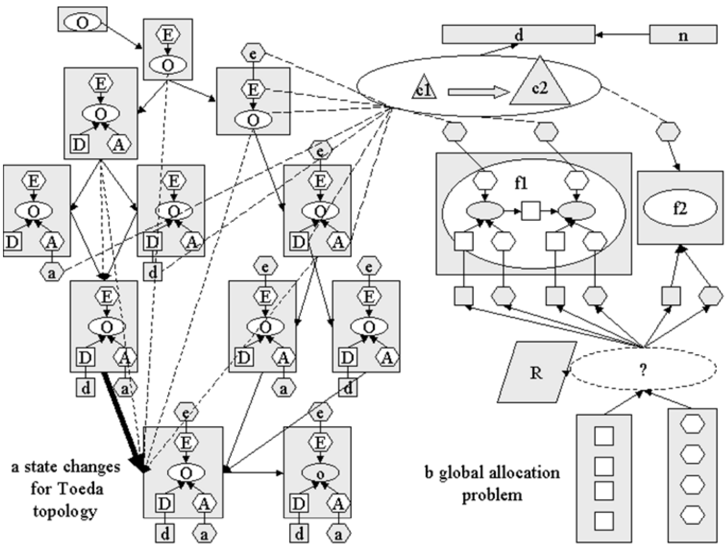

Figure 5 : Competence optimization problems

5 Conclusion: the challenge of global optimisation

The experiments made with the GEFO prototype could only highlight the existence of a "problem space". It is obvious that the moment for a better organization of it has come, in order to be able to at least estimate solvability - and I am taking steps in this direction. For the moment I signal the opportunity of a interdisciplinary encounter (the dynamic system theory, distributed cognition, semantic Web, operational research, workflow management, agent orchestration, etc) for attacking problems like the one exposed above. To which many others can be added, very difficult as that exposed in figure 6b: by what methods can we optimise the allocation (distributed in time and space) of the support documents and persons that can sustain an ensemble of users using a certain group of functions for the orchestration of procedures that should produce a certain modification of the global competence situation?

Treating such a question requires, among other things, a global vision concerning the unitary metabolism of knowledge… represented in reference structures… incarnated in participants… whose competences evolve… as a consequence of executing procedures with instructive effects…which exploit the potential competence leap of the explanations… incorporated in documents and provided by assistants. That shows the interest for using metafunctions in management (modelling, explaining, orchestrating, adapting and reproducing) a knowledge system's physiology.

References

1 Mizoguchi, R. Introduction to Ontological Engineering. In *New Generation Computing*, 21, pp. 365–384, 2003. Ohmsha Ltd and Springer Verlag.

2 Le Moigne, J. L., La modelisation des systemes complexes, Dunod, Paris 1990.

3 H. Maturana and F. Varela: Autopoiesis and Cognition: The Realization of the Living, Boston: D. Reidel, 1980

4 Odobleja S., *Psychologie consonantiste.* Librairie Maloine, Paris (1939), second edition E. S.E Bucuresti (1979)

5. Oberweis A. An integrated approach for the specification of processes and related structured objects in business applications, *Decision Support Systems* 17, 31-53, 1996, Elsevier

6. Chittaro L, Tasso C, Toppano E, Putting functional knowledge on firmer ground Applied Artificial Intelligence, 239-258, 1994

7. Kitamura, Y., Kashiwase M., &others; Deployment of an ontological framework of functional design knowledge, *Advanced Engineering Informatics*, 18(2):115-127, 2004

8. Beeson I, Exquisite variety: computer as mirror to community. In *Interacting with computers* 14 643-662, 2002, Elsevier Science

9. Herrmann T, Hoffmann M, Kunau G, Loser Kai-Uwe, A modelling method for the development of groupware applications as socio-technical systems *Behaviour & Information Technology* V 23, No2, 119-135, 2004, Taylor & Francis ed

10. Diaper D. Scenarios and task analysis, IN *Interacting with computers* 14 379-395, 2002

11. Bustard D.W., He.Z, Wilkie F.G., Linking soft systems and use case modelling through scenarios, IN *Interacting with computers* 13 97-110, 2000, Elsevier Science

12. Isenhour P.L., Rosson M.M., Caroll J.M, Supporting interactive collaboration on the Web with CORK *Interacting with computers* 13 655-676, 2001, Elsevier Science

13 Pinelle D.,Gutwin C., Greenberg S.: Task analysis for groupware usabiliy Evaluation: Modeling shared workspace tasks with the mechanic of collabortion, ACM Transactions on Computer-Human Interaction, Vol 10, No4, Dec 2003, 281-311

14 Korner C.; Sequential processing in comprehension of hierarchical graphs, Applied Cognitive Psychology, 18, 467-480, 2004, John Wiley &Sons Ltd

15 Allwood J, Traum D, Jokinen K.: Cooperation , dialogue and ethics, Int J Human-Computer studies, 53,871-914, 2000

16 Miyahara, K. ; Okamoto, T., Collaborative information filtering in cooperative communities, *Journal of Computer Assisted Learning*, 14(2), 100-109, 1998

17 Lehmann, M.M, Kahen G., Ramil J. F., 2002. Behavioural modelling of long-lived evolution process- sonme issues and an example. *Journal of software maintenance and evolution : research and practice*, 335-351,2002,John Wiley & Sons ed.

18 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)

19 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

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

21 Rosca, I.: Knowledge management instrumentation for a community of practice on the semantic Web, *Symposium REF*-2005, Montpellier (2005)

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

23 Hollan J., Hutchins E, Kirsh D: Distributed cognition: toward a new foundation for Human-Computer Interaction research, *ACM Transactions on Computer-Human Interaction*, Vol 7 No2 ,174-196, 2000

24 De More A., Patterns for the pragmatic Web, 13th *Int. Conference on Conceptual Structures*, http://www.starlab.vub.ac.be/staff/ademoor/papers/iccs05_demoor.pdf, 2005

25 Rosca, I.: Towards a systemic vision of the explanation process; the story of a research on integrating pedagogy, engineering and modeling- PhD thesis (1999)

26 Raghunathan S A structured modelling based methodology to design decision support systems. In *Decision Support Systems* 17, 299-312, 1996, Elsevier Science B.V.

27 Noelle,T, Kabel,D Luczack.H.: Requirements for software support in concurrent engineering teams, Vol 21 No 5, pp 345-350, *Behaviour and information technology*, 2002

28 Vantroys T, Peter Y: Cow, a flexible platform for the enactment of learning scenarios, CRIWG roceedings,Autrans, France, 2003, Springer Verlag

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

30 Paliouras G., Papatheodoru C, Karkaletsis V, Spyropoulos C, Discovering user comunities on the internet using unsupervised machine learning techniques. In *Interacting with computers* 14 761-791, 2002, Elsevier Science

31 Caroll, J.M. Five reasons for scenario based design, *Interacting with computers* 13 43-60, 2000

32 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, John Wiley & Sons ed.

33 Arias E, Eden H, Fischer G, Gorman A, Scharff E: Transcending the Human Mind - Creating Shared Understanding through Collaborative Design. ACM Transactions on Computer-Human Interaction, Vol7 No1 84-113,2000

34 Stathis K, De Brujin O, Macedo S: Living memory: agent- based information management for connected local communities, *Interacting with computers* 14 663-688, 2002, Elsevier

35 Witkowski M, Neville B, Pitt J, Agent mediated retailing in the connected local community, In *Interacting with computers* 15 , 5-32, 2003, Elsevier Science

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