ORGANIZATIONAL FUNCTIONS AND ENTERPRISE SELF-MAINTENANCE: A FRAMEWORK FOR INTEGRATING MODELLING, MONITORING AND LEARNING

David Aveiro[1], José Tribolet[1,2]
[1] *Organizational Engineering Center, INESC.*
[2] *Department of Information Systems and Computer Science,*
Instituto Superior Técnico,
Technical University of Lisbon.
{david.aveiro,jose.tribolet}@ceo.inesc.pt

Process orientation is a major concern nowadays. For this reason there have been great advances in business process modelling as a means to better understand and facilitate organizational operation and evolution. However, organizational functions like logistics, quality, finance, etc. still exist and strongly influence aspects like change processes, or the structuring of organizational units even if a "process culture" is established. This paper proposes a framework for integrating modelling, monitoring and learning aspects associated with the functional dimension of the enterprise. This framework is based on: (1) an ontology proposed for the functional dimension of an organization; (2) the fundamental concepts: entity, role and activity, that are the base for current research work by other authors, on modelling other dimensions or concerns of an enterprise architecture. Also an example of application of the framework is presented as a proof of concept.

1. INTRODUCTION

This paper proposes a framework for integrating the modelling, monitoring and learning associated with the functional dimension of the enterprise. This purpose comes in the context of research work done at Centro de Engenharia Organizacional (CEO) of INESC INOV, whose current view of organizational modelling is synthesized in the work of (Sousa et al., 2005), where key concepts for modelling the organization's enterprise architecture using the Unified Modelling Language (UML) are discussed. The authors argue that "modelling the multidimensional aspects of the enterprise should be organized into five architectural components: Organization, Business, Information, Application and Technological architectures. It is our purpose to further complement the above mentioned work, to incorporate the functional dimension of the enterprise, being this paper a second step in that direction. As a first step, and for the clarification of the function concept, in another paper we bridged knowledge from diverse areas like management, biology and

philosophy to arrive at an ontology that describes the main constructs of the functional dimension of an organization and their interrelation. Findings of this work important for the purpose of this paper are presented on section 2. In section 3, subsection 3.1, we introduce the fundamental concepts entity, role and activity, that are the base for current research work by (Sousa et al., 2005), on modelling other dimensions or concerns of an enterprise architecture. These concepts also serve as a basis for the framework proposed in this paper – in subsection 3.2 – with the purpose of integrating Sousa et al.'s current modelling methods, with the monitoring and learning aspects of the functional dimension of an enterprise. Subsection 3.3 clarifies the frameworks components and interrelations with an application example that serves as a proof of concept. Conclusions and future work based on hereby presented work can be found in section 4.

2. ORGANIZATIONAL FUNCTION: AN ONTOLOGY

In management/organization literature, the essence of the organizational function concept is not clear, existing several views and incoherencies in the use of this word. We can however infer one pattern: this word – *function* – is used to describe the aggregation of several processes into one more abstract macro-process, usually executed by an organizational unit (like marketing, logistics, IT, etc. But what's the added value of calling this macro-process a function? What properties emerge in a process that groups other processes, that justify calling it a function? To find what these properties could be, we analysed perspectives on the function concept from biology and philosophy of biology and found that some seem to apply to organizations. In this section we cover a summary of the ontology that resulted from the thorough analysis of the several perspectives studied.

We proposed an ontology for the organizational function concept, anchored on the fundamental concept of activity (we used the equivalent name: *process*). The word *function* has its root in the Latin word "functio" which means functioning, which in turn means "doing well in a regular way". *Functions* are process interdependency relations that determine the nature of organizations as viable (cohesive) systems. *Processes* in an organization are all interrelated or interdependent in some way due to the intertwined nature of its operation. Changes in the characteristics of inputs or outputs of *processes* in an organization will affect in some way (dysfunctional or not) the operation of other *processes* elsewhere – process interdependence (Christensen, 2002) and non-locality (Chauvet, 1993).

So that the organization can survive – maintain a capability of recursive self maintenance (Bickard, 2001) – certain *processes* (or parts of processes) will have to serve the capability of maintaining the conditions on which the organization is able to maintain its cohesion and survive (primary goal of any organization). That is, an organization has to have processes (or parts of the processes themselves) which are dedicated to monitor if key state variables are in the state of "doing well in a regular way" – or in other words, respecting the established *norm* (Bickhard, 2002) or *goal* (Kueng and Kawalek, 1996) for them – and work as (or invoke elsewhere) *resilience* dynamics (Christensen, 2002), that is reaction mechanisms of the system to perturbations that risk a departure from the *norm* and entrance on a state of dysfunction (converse of function).

If in a certain *process* there is a departure from the *norm*, this can be due to: an *expected exception* – the case in which previously existing *business rules* (Eriksson, 2000) (BRG, 2000) – methods embedded in the *process* – or other *processes* (invoked due to the *exception*) will work as a *resilience* mechanism to try to restore the *norm*; or an *unexpected exception* (Mourão and Antunes, 2005) – the case where the organization will enter in a dynamic space where *microgenesis* processes will occur (Bickard, 2001). They consist in heuristically, trough trial and error tentatives, based on past similar problem situations and analysis on its results, construct or change existing *processes* and/or *business rules* to circumvent or solve the problem that caused the *exception.* These *microgenesis* processes can be highly complex and unpredictable, depending on the degree of complexity of the problem that caused the *exception* and amount of *knowledge* necessary to investigate and understand necessary aspects of the problem. Just as the necessity of specialized *knowledge* (and its coordination) is at the genesis of the organization (Weeks, 2003) (Chandler, 1980), also specialized *knowledge* is at the heart of the micro construction (or change) of the organization, as a way of adaptation (realized in new resilience processes/rules) to new environmental conditions.

Summarizing, the ontology of an organizational function and its dynamics is depicted on Figure 1. Given a *process* X, we can elicit an *organizational function* by stating the following interrelated artefacts: (1) a *norm* (goal value) for a certain state variable of the process; (2) which other *process* Y (or processes) depend on this *norm*, in order to remain *functional*; (3) the set of *business rules* (embedded in the process itself, or other processes) that work as *resilience* mechanisms to *expected exceptions* and try to re-establish the *norm* to the *process functioning,* either by internal measures, or by invocation of other processes that will make inputs of process X change in a way it returns to normal functioning (4) set of specialized *knowledge* related to *process* X and Y's domain, that is, information about business rules that: (i) worked or not in previous real (microgenesis) situations in the organization (ii) are best (or not so good or even bad) practices from theoretical (or real) situations accumulated in the memory of the agents executing the microgenesis processes (note that all these "theoretical situations" are patterns that had to be derived and elicited from previous real situations in real organizations as a result of a microgenesis process). This knowledge is valuable input for the *microgenesis* process of treating *unexpected exceptions* that will select new successful business

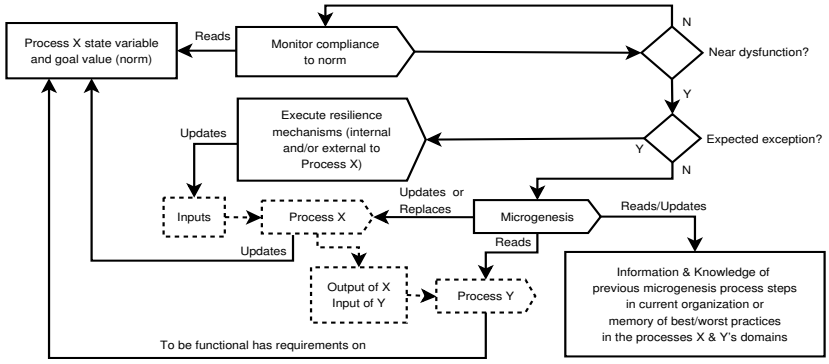

Figure 1 - Depiction of an organizational function's artefacts and dynamics

rules to be integrated in process X, or ultimately replace it.

3. FRAMEWORK FOR INTEGRATING MODELLING, MONITORING AND LEARNING

3.1 Enterprise architecture fundamental concepts

In this section we present the fundamental concepts – or primitives – for enterprise architecture modelling, that are defined in (Sousa et al., 2005): "we define the fundamental concept of entity and activity and that of role. These three concepts allow complex interactions of entities to be abstracted".

Entity – An organization is composed of entities. Entities are nouns that have a distinct, separate existence, though it need not be of material existence. In enterprise modelling, an entity can be a person, place, machine, concept or event that has meaning in the context of the business, and about which some information may be stored because it is relevant for the purpose of the model. An entity is characterized by its attributes and methods. These features can be either intrinsic or extrinsic. Intrinsic features describe the entity in isolation, while extrinsic features arise from the relationships with other entities.

Role – A role is the observable behavioural of an entity in the scope of a specific collaboration context. Hence, a role represents the external visible features of that entity when it collaborates with a set of other entities in the context of some activity. Each role represents a subset of its external or extrinsic features in the context of a specific collaboration defined in a role model.

Activity – An activity is an abstraction representing how a number of entities collaborate through roles in order to produce a specific outcome. Similarly to an algorithm, an activity aims accomplishing some task which, given an initial state, will always end in finite time and in a recognizable end-state. An activity may also be functionally decomposed into a finite set of further activities, thus add detail to the specification.

3.2 Framework

In the context of the previously presented fundamental concepts, an *organizational function* is a special activity *F* that corresponds to the collaboration of the entities described in the following paragraphs.

An activity *X*, relates to a state variable *a*, on which an activity Y has requirements in terms of respecting a certain norm *n* that restricts accepted states or values for *a*, so that *Y* can remain functional. *X* can have an internal mechanism of resilience such that in the occurrence of an expected exception that endangers compliance to *n* (provokes an approach or entrance in a state of dysfunction), a certain set of business rules *Ri* will try to handle the exception and avoid dysfunction.

If we are in the presence of an unexpected exception that endangers compliance to *n*, the resilience mechanism can enter in a microgenesis (learning) dynamics that can change existing (or generate new) business rules that try to handle the exception

and if successful, can be integrated in *Ri*. In case *Ri* cannot handle the exception (due to lack of knowledge or authority of the actors present in X's collaboration), actor *W*, responsible for monitoring *a*, will invoke activity *Z*, which constitutes an external resilience mechanism and implies a shift of control flow to the collaboration realized by *Z*. So that *X* can get functional again, *Z* may be able to handle and resolve the exception either by being able to change *X*'s inputs, or by also entering in microgenesis and for example, changing *X*'s own nature – algorithm, giving other kinds of inputs, etc.

Collaborations that execute microgenesis dynamics are, by nature, highly unpredictable and unstructured, but an important aspect to facilitate microgenesis dynamics is to keep a microgenesis log of steps taken (trial and error, justifications of decisions, etc.) until successful business rules are put in place that can resolve the exception. By nature (Bickard, 2001), this information is valuable for future microgenesis dynamics working on the dysfunctional activity. An essential role in microgenesis dynamics is of the knowledge of best/worst business practices associated with *X* or *Y*'s domains, that can be removed or instantiated according to the organizational scenario.

In failing to avoid dysfunction of *X* (*Z* itself is in a state of dysfunction), actor *W*' responsible for monitoring *Z* will have to invoke activity *Z*' and successively this pattern will have to repeat (hence the recursive nature of the organizational function concept) – eventually reaching the highest and ultimate (still internal) activity in the organization that can totally reconfigure the system (the *governance* activity) so that *X* can again return to a state of function, where the organization stabilizes (is able to maintain itself). In the case of (the *governance* activity) failing to do so, dysfunction can propagate to several other activities in the organization, eventually leading to its total collapse. In Table 1, these entities are further described, along with their roles in the collaboration that realizes the activity organizational function *F*.

Table 1 – Entities in the collaboration that realizes an organizational function F

Entity	**Description**	**Role**
Activity *X*	The collaboration on which an organizational function (a collaboration itself, named *F*) is being elicited. From the point of view of the functional dynamics of *F*, *X* is an entity.	monitored
Activity *Y*	A collaboration that depends on good and normal outcome of Activity *X*.	dependent
Observable state *a*	A state variable *a* whose value depends on behaviour of Activity *X*, and on which activity *Y* has requirements, so that activity *Y* itself respects requirements on one or more state variables (*a'*, *a''*, etc. whose value, in turn, depend on behaviour of activity *Y*) on which other activities will have requirements	observ-able state
Normative require-ments on a: *n*	Allowed and/or disallowed states/values/variations for *a* that respect the norm of functioning in terms of requirements of Activity *Y*	business goal
Internal	Set of business rules established for Activity *X*,	internal

Entity	Description	Role
resilience mechanism *Ri*	that work as an internal resilience mechanism, that is, contribute to the maintenance of normative requirements on *a*	resilience
Actor *W*	Entity that is monitoring Activity X. More than one Actor can be doing this (for example, the actor executing the activity can, himself be monitoring *a*)	monitor
Activity *Z*	Collaboration that is invoked (by Actor W) in case there is a departure from *a*'s norm that Activity *X*'s internal mechanism of resilience cannot withstand (activity is near, or already in the state of dysfunction), and therefore, constitutes an external mechanism of resilience	external resilience
Microgenesis Activity *M*	Collaboration with capacity to repair/replace Activity *X* (or other activities on which *X* inputs depend and are causing the dysfunction) in the case of unexpected exceptions	micro-generators
Microgenesis Information *I*	This entity has two aspects: (1) Microgenesis log, that is reasons why current internal and external mechanisms of resilience were chosen/selected (results obtained in face of other mechanism tried); (2) Memory of best/worst practices in Activity *X* and *Y* domains of knowledge	Organizational learning and knowledge

The invocation of resilience mechanisms can be characterized as *internal* or *external* and can be *proactive* or *reactive*. *Internal* is when activity *X* itself has embedded business rules as an internal resilience mechanism and actor with the role of executing it also has the role of monitoring *a*. *External* is when it's the activity organizational function *F*, executed by an actor with the role of monitoring *X* that detects dysfunction and activates resilience mechanisms. *Proactive* is when the exception that is (or is going to be potentially) disturbing normal functioning didn't yet provoked a departure from the norm and resilience mechanisms act immediately to avoid dysfunction. *Reactive* is when dysfunction has happened in *X* and is provoking an exception in *Y* (that depends on *a*, in turn dependent on *X*'s behaviour) and *Y*'s organizational function is invoking a certain collaboration context that will, for example, correct *X*'s behaviour. To clarify the presented framework and it's concepts, we now introduce, as a proof of concept, an example to show practical application and it's potential usefulness.

3.3 Application example

Consider a retail company with the following activities: Activity X – Transport sunflower seeds; Activity Y – Sell sunflower seeds. If *X* is in a context of eventually failing to respect the norm *n* (arrive before 9h in the day after) for state variable a (estimated time of arrival) due to a severe traffic jam (expected exception) between store no. 4 and store no. 32, actor *W* (truck driver) with the role of executing *X*, and

also the role of monitoring *X*, can, for instance (activate business rules belonging to *Ri*), accelerate the truck, or go trough a shortcut or alternative route to arrive on time (organizational knowledge that can or should be included in training or available to the actor). This is also an example of an internal and proactive invocation of a resilience mechanism.

If *W* encounters more traffic jams and keeps delaying (*Ri* not able to handle the exception), he can call the logistics centre, referring that he will not be able to arrive on time. This is an example of proactive invocation of an external mechanism of resilience, Activity *Z* – Manage fleet, now with the collaboration of Actor *W'* operator of the logistics centre of the company.

One outcome would be *W'* giving some advice on alternate routes for the truck driver to try (case of giving new inputs to *X*, whose executer *W* didn't have access to). Another outcome is *W'* to activate an alternate *X* - Send sunflower seeds by express mail - FedEx, (change in *X*'s nature, with another instantiation). Another external proactive invocation would be the case of truck driver to be unreachable by mobile phone, but the logistics agent could be monitoring the truck by GPS system and detect the impossibility of arrival on time and activate alternate *X* as stated above.

Examples of application of microgenesis concept follow. If in store no. 32, *Y* approaches a state of dysfunction (*a* is number of packs in stock, *n* is have a minimum of 30) due to dysfunction on activity *V* - Replenish stock (in turn affected by a very unexpected exception of a devastating migratory hamster attack in the supplier's terrain), the salesman *W* will (proactively) invoke process *Z* – Manage stocks, which has the role of external resilience mechanism for *Y*'s organizational function.

The inventory manager, executer of *Z*, enters in microgenesis mode, due to the unexpected nature of the exception and tries to construct alternative activities to solve or circumvent the problem. He contacted alternative suppliers, but they were also victims of the hamster attack. He then had the idea of checking if there was stock available on other stores. When he worked on other company five years ago, there was a process of shifting products between stores which wasn't a practice on his current company (case of application of tacit knowledge from the actor that corresponds to the current organization "learning") and then he had the idea to establish this activity, which turned out to be new business rules for activity *Z* that reflected on the construction of an alternative activity *V*, that previously only transported items between suppliers and stores and now transports items between stores. An example of using a microgenesis log would be registering the reason for choosing FedEx for alternative *X*: for example, a table with prices, showing it's much cheaper to send the packs by FedEx, than sending by UPS company. In a situation of necessity of changing *X*'s nature this information can be useful for future decisions.

4. CONCLUSIONS AND FUTURE WORK

The proposed framework allows a separation of concerns between the modelling of operational aspects of collaborations as presented in (Sousa et al., 2005) and the interdependences, monitoring and learning aspects of collaborations inherent to the functional dimension of the enterprise. As an innovative contribution to the enterprise architecture modelling field, this framework, can be used as a basis to capture the dynamics of monitoring and evolution (or adaptation) of the enterprise, in face of its environment.

To capture and model organizational functions of activities is way to capture knowledge about resiliency solutions, that is, how to manage resources while facing disruptive deltas in certain processes that otherwise would remain tacit in the minds of human actors of organizations. It's also possible to capture information on which steps made current business practices successful (in situations of unexpected exceptions) and allow a more conscious process of evolution or adaptation to new unexpected environmental perturbations. That is, information of the process of learning of an organization can collected and reused in other learning situations.

Trough the elicitation of networks of interdependences between activities, it may also be possible to (automatically or semi-automatically) detect which processes are more vital and establish priorities for creating proactive mechanisms of resiliency that can prevent and avoid propagation of dysfunction in a timely manner.

To further develop current work, a graduate and an MsC thesis are currently assigned to start in September. One has the purpose of extending a modelling tool (System Architect, Casewise, MooD or other) to allow a coherent application of the proposed framework to model functional dependencies between processes. The other thesis aims to create an intuitive tool for capturing in real-time, or after situations of unexpected exceptions, structured information about dynamics of selection of new business rules (microgenesis dynamics) and, at the same time, to provide informational support to actors participating in these dynamics.

5. ACKNOWLEDGMENT

Research work that led to results presented in this paper was possible thanks to the financial support of a PhD scholarship (Ref.: SFRH / BD / 13384 / 2003) subsidized by "Fundação para a Ciência e a Tecnologia - Ministério da Ciência, Tecnologia e do Ensino Superior" of the Portuguese government and by the European Social Fund (FSE), in the scope of the "III Quadro Comunitário de Apoio".

6. REFERENCES

1. Bickhard M. H., The Dynamic Emergence of Error Avoidance and Error Vicariants, Journal of Experimental & Theoretical Artificial Intelligence, Taylor and Francis Ltd, 2001
2. Bickhard Mark H., Process and Emergence: Normative Function and Representation, Department of Philosophy, Lehigh University, 2002
3. Business Rules Group, Defining Business Rules - What Are They Really?, 2000
4. Chandler Alfred D., The Visible Hand: The Managerial Revolution in American Business, Belknap Press, 1980
5. Chauvet G. A., Non-locality in biological systems results from hierarchy. Application to the nervous system., Institut de Biologie Theorique, Universite d'Angers, 1993
6. Christensen W. D. and Bickhard M. H., The process dynamics of normative function (The Philosophy of Biology), Monist, Hegeler Institute, 2002

7. Eriksson, H., Penker, M, Business Modeling with UML: Business Patterns at Work, OMG Press, Wiley Computer Publishing, 2000
8. Kueng P., Kawalek P., Goal-Based Business Process Models: Creation and Evaluation, Informatics Process Group (IPG), Department of Computer Science, University of Manchester, UK, 1996
9. Mourão, H. and P. Antunes, A Collaborative Framework for Unexpected Exception Handling, Groupware: Design, Implementation, and Use, Heidelberg, Springer-Verlag, 2005
10. Sousa P., Caetano A., Vasconcelos A., Pereira C., Tribolet J.Enterprise Architecture Modeling with the Unified Modeling Language2005
11. Weeks J. and Galunic C., A theory of the cultural evolution of the firm: the intra-organizational ecology of memes, 2003