Enterprise dynamic systems control enforcement of run-time business transactions

Sérgio Guerreiro[1], André Vasconcelos[2,3], and José Tribolet[2,3]

[1] Universidade Lusófona de Humanidades e Tecnologias, Escola de Comunicação, Artes, Arquitectura e Tecnologias da Informação, Campo Grande 376, 1749-024 Lisbon, Portugal,
[2] CODE, Center for Organizational Design & Engineering, INOV, Rua Alves Redol 9, Lisbon, Portugal,
[3] Department of Information Systems and Computer Science, Instituto Superior Técnico,Technical University of Lisbon, Portugal
sergio.guerreiro@ulusofona.pt,andre.vasconcelos@dei.ist.utl.pt, jose.tribolet@inesc.pt

Abstract. Business transactions models prescribe the design freedom restrictions of the business transactions dynamics but *per se* do not guarantee that organizational actors perform them accordingly. Enterprise dynamic systems control (EDSC) guarantees that the prescriptions are followed in the operation by performing a continuously cycle of observation, decision and control action. Control action actuates with a change in the business transaction models prescription to avoid the recurrence of unintended operations or a change in the control rules if the deviation from prescription is recognized as being innovative. This paper proposes a full DEMO-based ontology to enforce EDSC in the run-time operation of business transactions. The EDSC proposal is exemplified using *(i)* DEMO business transaction ontological specification, *(ii)* actor's qualification mechanim integrated with business transactions and *(iii)* business rules enforced in the actor's *ex-post* acts.

Key words: Business transaction, Control, DEMO, Governance, Model, Ontology, Operation, Organization

1 Introduction

An enterprise is a large, complex and non linear dynamic system whose dynamics encompasses both the business transactions and the organizational structure's systemic definition. To deal properly with the systemic definition of an enterprise, scientific foundations with new theories and methodologies that enable the understanding, design and implementation of the organization are needed. Business transactions occur at run-time across a distributed network of actors that constitutes an enterprise. Business transactions establish the design freedom restrictions prescribed for the organization, but they do not guarantee that the business actors perform the transactions accordingly. Business actors while playing an active role in business transactions are simultaneously the essentials

elements of the enterprise dynamic systems control, because they individually and/or collectively observe the reality and autonomously produce new control actuations that change it. Without actor's activity there are no performed acts thus the organization does not exist. To guarantee that the prescribed business transactions are respected it is necessary to continuously control the misalignments between the prescribed business transactions and the observed operation in the organization. The foundational concepts from dynamic systems control (DSC) theories, which are defined and used successfully in the Engineering fields for decades [1][2][3], are applied in this paper to the enterprise dynamic system to control its outcome. The emerging field of Enterprise Engineering (EE) [4] introduces capabilities to deal rigorously with the dynamic aspects of the business transactions using ontological models. An important indication of compatibility between the EE existing laws of organizational dynamics and the case studies developed with success in the industry is pointed in [5]. The integration of the DSC concepts with the EE concepts allows the understanding, designing and engineering implementation of the enterprise dynamic systems control.

The problem addressed by this paper - to design and implement EDSC capabilities in business transactions operating in a run-time organizational environment, taking in account the misalignments between operational conditions and prescribed references as defined by the organizational models - will be solved taking into account the scientific contributions of DSC and EE. The DEMO theory and methodology [6] that lies at the core of EE will be used in full to solve this problem. The solution consists in enforcing control in run-time business transactions using a bottom-up and top-down integrated approach. Organizational control is a continuous orchestration of combined low-level and high-level control actions taken by the organizational actors. At operational level, the solution checks misalignments between prescribed business transactions models and observed operations in the organization. These observations are used to trigger actions by the organizational control. Organizational control checks if the historical business transactions comply with the prescribed business rules. The control action results in changes in the prescribed business transactions models or changes in the business rules. In the first case of control action, the prescribed business transactions are changed to avoid the recurrence of unintended operations. In the second case of control action, innovation is recognized as positive and the deviations from the prescribed transactions models are incorporated in the new prescription. In other words, the observed misalignment is valuated as being a more innovative way of operating the organization and thus it is used to define a new organizational prescription, in what constitutes in fact continuous organizational learning. Organizational control appears in different forms and acts at different levels throughout the distributed enterprise, *e.g.*, actor's qualification or business rules implementation, but in all cases, repeating a pattern of observation, decision and control actuation.

The organizational control procedures in our solution are enforced using the concepts of business transaction and control.
A business transaction is a model representation of a given organizational real-

ity that is valid within a specific timeframe. As proposed in Enterprise Ontology (EO) [6][7], a business transaction involves *(i)* actor role definitions, in order to specify who is responsible for each part of the transaction, who initiates it and who executes it, *(ii)* a transition space definition, and *(iii)* a state space definition. Hence, actors are positioned at the core of this solution. Actors, which might be carbon-based or silicon-based, are autonomous and act according with their desires and interpretations of reality, for instance, in terms of personal interpretation, environmental change, requirements change or legal change. EO distinguishes the production acts from the coordination acts. A production act (P-act) contributes to bringing the goods and/or services (material or immaterial) that are delivered to the environment of the enterprise. A coordination act (C-act) enters into and complies with commitments towards each other regarding the performance of the production acts. The P-acts and C-acts have effect in the correspondingly two separate worlds: the production world (P-world) and the coordination world (C-world). Moreover, the actors dynamically change the operational reality of the organization while they perform their activities. For this reason, the enterprise is considered as a dynamic system. When a business transaction is instantiated, at a single instant in time, it is called run-time. When the models are designed is called design-time.
Control consists in the ability to drive, with a bounded effort, the operation of the enterprise towards a stable state whenever changes or perturbations occur. In the scope of an organization, control is performed implicitly and the overall organizational system maintains its stability involving all the actors, without a clear identification of which parts are controlled and which parts are controllers. Thus, our solution states that control *(i)* is not always either explicit or analytical, as cybernetic control approaches are, *(ii)* it is not possible to predict all the business transactions conditions to control before execution due to the high number of possible combinations that could happen at run-time and *(iii)* the identified deviation from the stable state is sometimes incorporated as organizational innovation. Control requires the capabilities of observation, decision and control actuation. The definition for decision is similar to the definition of enterprise governance proposed in [8][4]: *'...is the organizational competence for continuously exercising guiding authority over enterprise strategy and architecture development, and the subsequent design, implementation and operation of the enterprise...'*. The same concerns of guiding an organization based in references and acting in the subsequent design, implementation and operation exists in the enterprise governance and in the decision counterpart of DSC theory.

For clarification, this paper defines the boundary of the DEMO business transactions to enforce enterprise control. The research regarding enterprise dynamic system control must follow to other dimensions of the organization. The remaining of the paper is organized as follow. Section 2 defines the followed research methodology. Then, section 3 details the ontology to control the operation of the business transaction in an organization using the DEMO ontology. Section 4 discusses the obtained ontology. And finally, section 5 concludes the paper and points to future work.

2 Methodology

Organizational control is not a recentc concern, early in 1965, Emery and Trist [9] refer that a company is a open system where its behavior is only explained when analysed in conjunction with its interactions with the surrounding environment. Later, Hofstede [10], proposes that control should be formed and evaluated as a homeostatic[1] model rather than a cybernetic model. Much more recently, Tribolet and Magalhães [11] also state that agility and real-time reconfiguration capability [12] are requirements to the maintenance of the organizations. Also recently, Hoogervorst [8] states that the increasingly complexity is characterised by an increase that follows the size of organizations. Thus, precise models that are able to deal with this complexity without exploding in terms of size are needed. By other words, the models should be able to follow, at the same pace, the increase in the complexity size of companies [13].
This research is based in a simplification of the design-science research (DSR) as proposed by Hevner *et al.* [14] and Winter [15]. The following steps were performed so far: *(i)* identify the problem in the Enterprise Engineering domain using a simplified case study [16]; *(ii)* demonstration that no full solution exists for the case study supported by state of the art review with the following main focus [17]: Access control models [18], DEMO ontology [6], dynamic systems control [19] and Enterprise Governance [20]; *(iii)* usage of Enterprise Ontology to model a solution in the scope of Information Systems and Computer Engineering. Firstly, designing and implementing an EDSC solution. Then, verify and validate the design and the implementation of the solution using modal logics, simulations and case studies. Then, disseminating the contribution in the community, using technical reports, presentations and publications. And finally, learning from the interactions and restarting the research flow to EDSC redesign.

The remaining of this paper presents the obtained EDSC design after a series of iterations of this research approach.

3 Enterprise Dynamic Systems Control using DEMO

This section proposes an ontology to control the run-time business transactions modeled in the DEMO methodology [6]. The solution starts by identifying the dynamics of the business transactions in terms of their functional and constructional perspectives. The concepts of prescribed (or ex-ante[2]) model, observed (or ex-post[3]) operation, observation act, control act, time, control cycle and control competence layers are considered in the design of the control for the enterprise.

[1] Homeostatic control model considers that there are a large number of interrelated cybernetic systems within an organization, executing different business services and working side-by-side which usually involves communication between Humans and machines.

[2] The term *ex-post* is defined by the expression: "after the event".

[3] The term *ex-ante* is defined by the expression: "before happening".

The functional perspective is concerned with the behavior of a given function of a given system. Constructional perspective is concerned with the details that are included in a system that, in the end, allows the demonstration of its behavior. Using the two perspectives in complement, it is possible to have the identification of the desired behavior for the control and also the details that are needed to enforce it. The observation and control of the actor's activity in the scope of the business transactions is enforced using access control concepts integrated with the DEMO business transactions concepts. The transition space and the state space of the solution are described using the DEMO ontology. The implementation is independent from the definition, representation and meaning of the mandatory concepts to control the operation of the organization.

DEMO methodology encompasses 7 steps: first 4 steps are obtained by analysis and the last 3 are obtained by synthesis. The sequence is relevant because the methodological process is strongly dependent on the previous steps. In the end, the essential ontology that is obtained allows a true and complete traceability between the enterprise's concepts that could be shared by the different organizational actors. The deliverable of the DEMO methodology is a set of diagrams that are able to summarize the essential of an organization. Next subsections include the Transaction Result Table (TRT), then the most relevant Action Rules Specification (ARS) are pointed, afterwards the Object Fact Diagram (OFD) specification and finally the Organization Construction Diagram (OCD) is depicted and explained.

3.1 Transaction Result Table

Table 1 presents the transactions types and the correspondingly result types that are obtained when each transaction is completed successfully. These transactions are ontological. The infological and datalogical references are not considered at this stage. Four sets of transactions are identifiable with respect to DSC domain:

1. *Ex-ante* **definitions** regarding the business rules, the business transaction models and the accesses: T01, T02 and T03,
2. **Management** of the business rules, business transactions models and accesses: T11, T12 and T13,
3. **Control** regarding the actual session, the accesses and the business rules: T05, T06 and T07,
4. **Observation** of the actual session: T04.

Each transaction is valid regarding a specific period of time. It is interesting to notice that even at this early stage of the ontological design, the identified transactions consists in defining, observing, controlling and managing. Which are typical tasks performed in the scope, for instances, of the management of the software development process [21][22][23][24]. Figure 1 relates the different result types that are specified in the TRT. From right to the left, and from top to bottom, the following considerations are made. The result type R11 is related with result type R01 because of the management result of the business rules that implies the definition of a business rule. The same happens between the

Transaction Type	*Result Type*
T01 Business rule definition	**R01** Business rule BR has been defined for period P
T02 Model definition	**R02** Model M has been defined for period P
T03 Access definition	**R03** Access A has been defined for period P
T04 Observation of run-time session	**R04** Session S has been observed for period P
T11 Business rule management	**R11** Business rules BR have been managed for period P
T12 Model management	**R12** Model M has been managed for period P
T13 Access management	**R13** Access A have been managed for period P
T05 Run-time control	**R05** Session S has been controlled for period P
T06 Run-time access control	**R06** Access A has been controlled for period P
T07 Run-time business rule control	**R07** Business rule BR has been controlled for period P

Table 1. Transaction result table presenting the transactions types and the obtained result types.

R12 and R02 and also with R13 and R03. The result type R04 which is the observed session is related with the controlled session and consequently with the access controlled (R06) and the business rule controlled (R07). The result R01 is related with the result R07 because the control of the business rules only occur when the business rules have been defined. The same happens between the R03 and R06 regarding the access. The same does not happen with R02 because run-time control is not concerned in controlling if the instances of the DEMO models are complaint with the DEMO transactional patterns specification. Indeed, this work is the research conducted regarding the DEMO processor by Kervel [25].

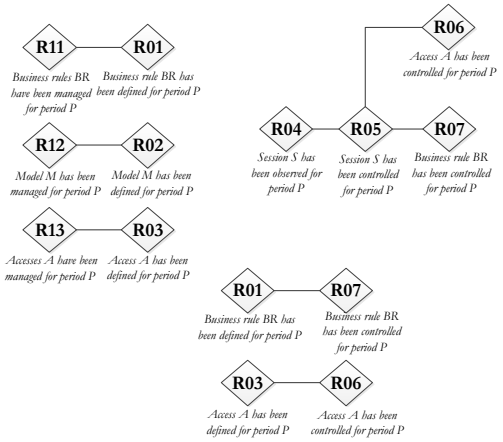

Fig. 1. Result structure of the result types obtained in the TRT of Table 1.

Within the EG boundary, in Figure 3, exists the responsibility for the observation of the operation of the enterprise and to act when needed. In detail, the business rule manager (A01) is responsible to self-initiate the business rule

definition transaction (T01) and also to initiate the business rule management transaction (T11). Similarly, the model manager (A02) is responsible to self-initiate the model definition transaction (T02) and also to initiate the model management transaction (T12). And finally, the access controller (A03) is responsible to self-initiate the access definition transaction (T03) and also to initiate the access management transaction (T13). The idea of having a self-initiate transaction is to enable the definition of the controller *ex-ante* references to be made by someone. If no reference are established then no control can be performed. References are considered herein as the control bootstrap. The actuation is performed by initiating one of the following transactions: business rule management (T11), model management (T12) or access management (T13). From the EG kernel functional perspective, the control of the run-time business transactions is performed by observing all the process steps operated by the users (CA04) and when needed T11, T12 or T13 are triggered.

The observation initiates the *observation of run-time session transaction* (T04). An user might be a person or a machine. Ontologically, transaction T04 means the connection of the EG kernel with the operation of a given business transaction that is complaint with a DEMO model. In practice, it means that observability is a mandatory requirement to enable EG. Once, the observation over the DEMO artifacts[4] is established then control is a matter of evaluating the observations and deciding which is the correct action to be taken. Then, every business transaction execution that we want to control must be connected with the boundary of the EG. The idea, is that the coordination and production, acts and facts, keeps being executed outside the EG. Therefore, EG is observing the operation and acting accordingly with its predefined definitions that were made by the business manager, model manager and access controller.

Moreover, the business transaction can only advance if EG allow it to advance. By default, a negative-policy is enforced. The ontological solution for this requirement is based in the wait link available in the DEMO process model aspect. To further clarification at this stage, the transaction T04 is similar to the technological interceptor concept proposed by Sun Microsystems [26]: *"a thin layer of software that verifies every bean transaction but only acts over it whenever a precoded situation happens"*.

The elementary actor role Interceptor (A04) initiates T05 which is named as the *run-time controller* and that creates a new fact in the ontology that is *"Session S has been controlled for period P"*. The run-time controller executes T05 initiating two parallel transactions:

1. **run-time access control** (T06), control if the access to the sessions should be granted or revoked,
2. **run-time business rule control** (T07), control if the *ex-ante* defined business rules are satisfied.

[4] DEMO artifacts are considered, in this paper, as the basic constructs defined by the DEMO ontology.

Listing 1. Action rule for the promise of transaction T06

```
1  when run−time access control of [Session] is promised
2  with user of [Session] is [User]
3  and role of [Access] is [Role]
4  and permission of [Access] is [Permission]
5  and staticconstraint of [Access] is [StaticConstraint]
6  and dynamiconstraint of [Access] is [DynamicConstraint]
7  and current time frame on [Period]
8  if [Role] is ElementaryActorRoleverified in [Role,DEMO:ElementaryActorRole]
9    valid within the current time frame on [Period]
10 and [User] is UserRoleverified in [User,Role] valid within the current time frame on [Period]
11 and [Permission] is Permissionverified
12 in [Role,Permission,DEMO:FactType,DEMO:ActionRule]
13   valid within the current time frame on [Period]
14 and [StaticConstraint] is StaticConstraintverified in [Role]
15   valid within the current time frame on [Period]
16 and [DynamicConstraint] is DynamicConstraintverified in [Role]
17   valid within the current time frame on [Period]
18 then run−time access control of [Session] must be executed
19   run−time access control of [Session] must be stated
```

The following DEMO artifacts are controlled: *Elementary actor roles* as specified in the ATD, *Transaction kind* as specified in the ATD and TRT, *Action rule* as specified in the ARS, *Fact type* and *Object class* and *Result type* as specified in the OFD. Transactions T06 and T07 are performed by the correspondingly actors: the run-time access controller (A06) and the business rules engine (A07).

3.2 Action Rules Specification

The promise steps regarding T06 and T07 are presented herein using the SBVR specification [27]. The actual observation of the T04 is only able to evolve *if and only if* all T06 and T07 are accepted successfully by T05. To start with, A06 is responsible to execute the run-time access control that is enforced between the user's activity in the enterprise reality and its privileges restrictions that are *ex-ante* defined. Considering an ontological specification, the access control of the DEMO model instances is enforced with five consecutive verification steps. The access to the artifact of the DEMO model instance is only granted *if and only if* the user fulfills all these five steps successfully as depicted in Listing 1.

The elementary actor role A07 is related with the capability of executing a set of business rules of the kind: *if...then...else*, as presented in Listing 2. Each rule encompasses a rule condition and a rule action. If the *rule condition* is TRUE then the *rule action* is triggered otherwise it is not triggered. In the case of more than one *rule condition* being TRUE at the same time, then the *priority* is used to only execute one rule action. The *rule condition* are calculated using the DEMO artifacts presented in the session.

3.3 Object Fact Diagram

Regarding the applicability of the obtained state model, remark that the state model is not a database relationship model, neither an entity-relationship diagram or an UML class diagram, but rather a conceptual presentation that would be used to be shared among the enterprise's actors in order to exist an unique

Listing 2. Action rule for the promise of transaction T07

```
1  when run−time business rule control of [Session] is promised
2  with priority of [BusinessRule] is [Priority]
3  and rulecondition of [BusinessRule] is [RuleCondition]
4  and ruleaction of [BusinessRule] is [RuleAction]
5  do for all [RuleCondition] in [BusinessRule]
6    if TriggerCondition in [Session] valid within current time frame on [Period]
7  od
8  then run−time business rule control of [Session, RuleAction] with higher [Priority]
9  must be executed
10   run−time business rule control of [Session, RuleAction] with higher [Priority]
11 must be stated
```

conceptual schema of a given world. In other words, the goal of a state model is to specify, in the OFD (Object Fact Diagram), the subjective interpretation of a given world using two concepts: *(i)* individual concepts and *(ii)* concept type. From the variables available at the TRT depicted in Table 1 the following fact types are identified: **Business rule**, regarding a rule that is enforced in the run-time business transactions to react to an *ex-ante* condition; **Model**, regarding the DEMO business transaction model that is supposed to be followed. The model is used to enforce the security concerns; **Access**, regarding the *ex-ante* security concerns that are supposed to be followed and the *ex-post* restrictions that the security concern should satisfy; **Session**, regarding all the activities that are attempted by the users in the reality of the organization. All these fact types are related with another variable which is the period of time. The fact type **Period** is thus considered to bound the result type execution within the desired period of time. Figure 2 depicts the proposed overall OFD, it is based in the previous definition of the TRT, ATD and ARS.

3.4 Organization Construction Diagram

Figure 3 depicts the OCD of the EG ontology. Some few considerations are made regarding this model. An information link exists between A12 and T13 to obtain the access configuration from the DEMO model definition during the configuration phase of the references. The A07 has three information links to the T11, T12 and T13. They correspond to the control action that are taken when any misalignment occur. T11 corresponds to a self-change in a business rule, T12 corresponds to a change in the DEMO model to be controlled, it is expected to be performed by the model manager which is a person and T13 that corresponds to a change in the access management. A06 has an information link with the T13 because of the static and dynamic constraints that need to change the access management.

4 Discussion

The main driver for this research started with a single statement that is followed by many practitioners and researchers: enforcing control in the activity of the

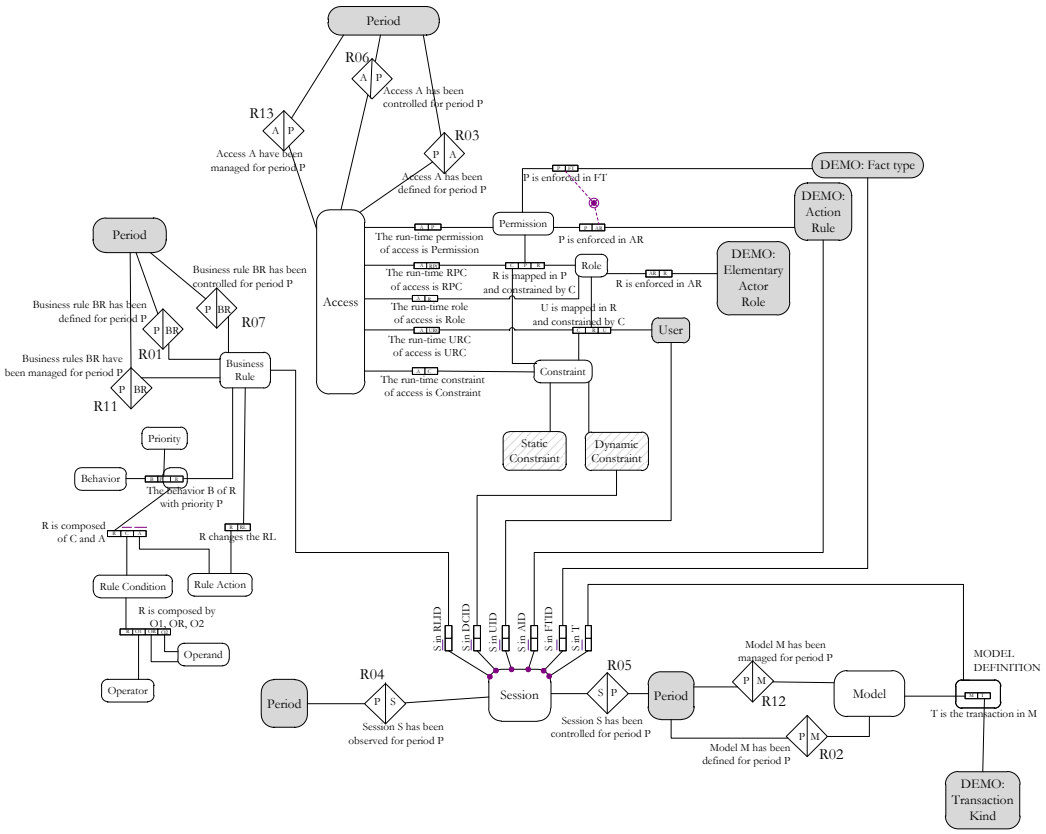

Fig. 2. Object Fact Diagram for governance enforcement in DEMO ontology: the overall state space representation.

organizations. In general terms, it is related with the ability to drive, with a bounded effort, the operation of the enterprise towards a stable state whenever occur changes or perturbations. This ability by its turns already exists in the DSC field. A DSC approach is composed of two mains parts: the controller and controlled process. Many times, control is considered as an isolated organizational component that reacts accordingly with the behavior of the system to be controlled. In practice, due to the organizational complexity, this approach is insufficient as long as the dynamic of the system to be controlled could not be fully specified. Having this problem in mind, our solution is inspired in the DSC but with a specialized view, which is how to enforce control in the transactions that operates in a real run-time organizational environment, in order to face the misalignments between the operational conditions and the references defined by the organizational models. In this way, this research effort is bounded to offer well defined scientific developments using a constructive and incremental approach. Moreover, we consider that control is a matter of awareness, where

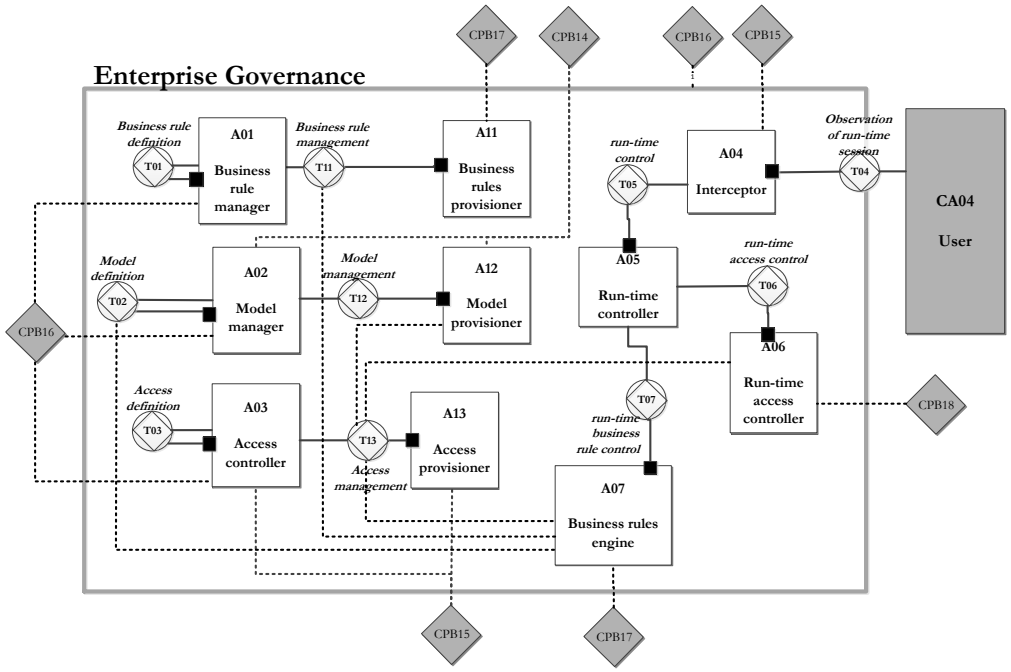

Fig. 3. Organization Construction Diagram for governance enforcement in DEMO ontology.

individual actors and collective actors creates the sense of awareness towards the need of controlling, and then, the execution of their actions reveals the specific control actions that they perform in the reality. Usually, control in organizations is strictly considered as a black-box perspective that lies inside the actors capabilities, and the models of the business transaction specifies what the actors should do. Then, the organization trust that control is tacitly implemented. Again, this paper, identifies that this perspective is valid but only represents a partial coverage of control in organizations. In a black-box perspective, control is thus only identified by the result of the actors in the reality which can only be seen as a whole but not by its parts. Thereafter, in this scope, this paper defeats that besides the capability of the individual and collective actors to decide and take control actions, also autonomous mechanisms of observing and acting should be completely understood by ontologies and then enforced in the operation of an organization.

It is true, that actually, the capability to fine-grained control the access to the artifacts of an organization, or even, the capability to define and implement business rules, are most of the times, decoupled from the enterprise design. The practical consequence of this decoupling, is *(i)* the duplication of effort in the control and models design counterparts and *(ii)* with the designed models not

aligned with control. Nowadays, a change in the control requires a change in the model design, and *vice versa*. Integrating the access control at the models design enables a fine-grained access control to the artifacts directly in the design with a perfect alignment that enables the continuous changes throughout time. Moreover, this integration enables a full observability of the operation of the enterprise and thus allows the enforcing of business rules that are able to react in run-time based in the actual and historical observations. As a consequence of this, the business rules are kept as directions that are truly followed by the organization. Also, the separation of concepts between the controller and the controlled process should also be considered and precisely ontological specified if a real implementation is expected. If so, it is possible to *(i)* continuous observe the design restrictions of the run-time business transactions from the inside and then *(ii)* actuate with a change in the business transaction models when needed. In the limit, parts of the control could be performed by automatic systems rather than exclusively performed by actors.

DEMO is used in this paper to argue about the enterprise's ontology own proposal and also to specify a solution that could be used in any DEMO organizational model. Despite this strong foundation in DEMO community we state that are not extending the DEMO meta-model but rather increasing the knowledge regarding control in the DEMO models. To summarize the reasons for our ontology are: **Automatically configure the accesses**, using the DEMO models itself. Avoiding all the effort related with the *ex-ante* access configuration, as presented in Listing 1 and by *Access* fact type in Figure 2. Only the users need to be related with their role(s), which from our understanding is the specific responsibility of each organization. Moreover, it is identified that EO do not cope with the concepts of access control. It is postponed for the implementation phases, however, access control is not an *add-on* that could be later added, as stated by ACM community, but rather something that should be included in the essential description of the organization, The business rules **enforces high-order actuation** in the operation of the organization, as presented in Listing 2 . That is true, if exists a set of values in the *Operand* fact type that are derived from the *Session* fact type. Then, they could be used to validate the *Rule condition* and to trigger the *Rule action*.

The objections are the following: the difficulty in designing ontological models for the enterprises, and the demand to use more case studies taken from the industry to full validate this EDSC proposal. To conclude, from our point of view, the ontological modeler should always have in mind that besides using the ontology to express the essence of the organization between different stakeholders, it should also be used to strongly found the software development. Moreover, a complete traceability between normalized EO and normalized software engineering [28] would optimize the deliverable of software cycles in enterprises.

4.1 Implementation aspects

Besides the presented ontological discussion, the implementation of EDSC is at this stage of the paper envisioned. The *Session* fact type plays a central role in

the capability of observing what is being done in the operation of the enterprise. The implementation of an universal identification to all DEMO artifacts kept by *Session* allows a fine-grained controllability of each DEMO artifact involved in DEMO business transaction steps performed by the actors. Hence, a new transaction (T08) is proposed to deliver a result type of "R08 Identification ID has been controlled for period P". This proposed implementation is quite similar with a class concept developed in any object-oriented programming language, such as JAVA or C++. The instantiation of such a class is the act of giving an unique memory allocation, with its own data space where the properties of that object are stored. No other object is equal with the first one in terms of identity. Universal identifiers are postponed to this implementation phase, because they represent the instantiation of Figure 2 and not the EDSC concepts.
Other issue that should be taken into account in the implementation is the large amount of data that is obtained from the observation (T04) of run-time business transactions. It is useful to decide upon the correct control action to take but places a huge computational problem. Future research should be done regarding this issue.

5 Conclusions and future work

Our DSR approach represents an effort for conceptualizing the control patterns that should be included in the design of the real systems that supports the organization. Moreover, the control concepts presented herein are directly related with the EG area regarding the lower level of governance for a business transaction. For other aspects of the organization, other control layers should be further considered. The solution of this paper consists in enforcing control in run-time business transactions using a bottom-up and top-down integrated approach. Organizational control is a continuous orchestration of combined low-level and high-level control actions taken by the organizational actors. Run-time business transactions control is defined as the delegation capability of assuring that the responsibilities, competencies and authorities are being followed by the actors by the mean of the accountability and actuating in the predefined models. Accountability capability must be explicit considered in the design of the control of the organization, either considering it explicit or considering it implicit within the actors but with a clear separation of concern. Using DEMO, the confidence that is obtained is thus far more significant than with the approaches that are subjective and that leads the designer to a set of own decisions. In most of the cases, the results that are obtained suffer from mistakes and errors that should then be only identified while in development time.

Regarding EG, we emphasize that it is not only a matter of guiding the authority over the enterprise strategy but also an extremely important matter of creating consistency between the design and the operation of the enterprise. Focusing in this principle, we are able to present a solution that *(i)* clear identify the competence of the different atomic concepts that are included in a organizational model either Human or machine based and, *(ii)* enforce the key activities

of responsibility and accountability in that models and then that *(iii)* use the authority to change the models when needed.

Lastly, besides the development of an EG approach to be used by the large enterprises, we believe that an open solution for the EG regarding the small and medium enterprises (SMEs) is a benefit that today it is not available at all. The effort needed to implement a complete solution encompassing business process, business rules and security enforcement is not reachable by many of the SMEs. Hence, the economical and financial impact of the knowledge introduced by this paper would be of great benefit to this kind of companies.

6 Acknowledgments

This work was supported by a PhD scholarship: SFRH/BD/43252/2008, and also by a national project: PTDC/CCI-COM/115897/2009, MOBSERV, Sistemas Facilitadores da Utilização de Serviços por Dispositivos Móveis, from Fundação para a Ciência e a Tecnologia, Ministério da Ciência, Tecnologia e Ensino Superior, Portugal.

References

1. Ribeiro, M.I.: Análise de sistemas lineares. Volume 1 e 2. IST Press (2002)
2. Kuo, B.: Automatic Control Systems. seventh edn. Prentice Hall International Editions (1995)
3. Franklin, G.F., Powell, J.D., Emami-Naeini, A.: Feedback Control of Dynamic Systems. Second edn. Addison-Wesley Publishing Company (1991)
4. Albani, A., Dietz, J., Hoogervorst, J., Mulder, H.: Enterprise engineering: the concise manifesto. In CIAO network meeting minute, version 7 (2010)
5. Ven, K., Verelst, J.: The adoption of demo: A research agenda. In Aalst, W., Mylopoulos, J., Sadeh, N.M., Shaw, M.J., Szyperski, C., Albani, A., Barjis, J., Dietz, J.L.G., eds.: Advances in Enterprise Engineering III. Volume 34 of Lecture Notes in Business Information Processing. Springer Berlin Heidelberg (2009) 157–171 10.1007/978-3-642-01915-912.
6. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer (2006)
7. Dietz, J.L.G.: Architecture: building strategy into design. Nederlands Architectuur Forum. Academic Service (2008)
8. Hoogervorst, J.A.: Enterprise Governance and Enterprise Engineering. The Enterprise Engineering Series. Springer Science (2009)
9. Emery, F., Trist, E.: The Causal Texture of Organizational Environments. Systems Thinking, Frederick Emery edition (1965)
10. Hofstede, G.: The poverty of management control philosophy. The Academy of Management Review **3**(3) (July 1978) 450–461
11. Tribolet, J., Magalhães, R.: Engenharia Organizacional: das partes ao todo e do todo às partes na dialéctica entre pessoas e sistemas. Editora Fundo de Cultura. In: Ventos de Mudança, Brasil (2007) in Portuguese.
12. Santos, C.A.L.d.: Modelo Conceptual para Auditoria Organizacional Contínua com Análise em Tempo Real. PhD thesis, Universidade Técnica de Lisboa, Instituto Superior Técnico (2007)

13. Mannaert, H., Verelst, J., Ven, K.: The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability. Sci. Comput. Program. **76**(12) (2011) 1210–1222
14. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Q. **28** (March 2004) 75–105
15. Winter, R.: Design science research in europe. Eur J Inf Syst **17** (2008) 470–475
16. Guerreiro, S., Vasconcelos, A., Tribolet, J.: Adaptive access control modes enforcement in organizations. In Quintela Varajão, J.E., Cruz-Cunha, M.M., Putnik, G.D., Trigo, A., eds.: ENTERprise Information Systems. Volume 110 of Communications in Computer and Information Science., Springer Berlin Heidelberg (2010) 283–294
17. Guerreiro, S., Vasconcelos, A., Tribolet, J.: Enforcing control in the run-time business transactions. In: Proceedings of 5th International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS 2011), ISBN: 978-87-91831-42-3, Center for Industrial Production, Aalborg University, Fibigerstraede 16, DK-9220 Aalborg, Denmark. (2011)
18. Sandhu, R., Ferraiolo, D., Kuhn, R.: The nist model for role-based access control: towards a unified standard. In: Proceedings of the fifth ACM workshop on Role-based access control. RBAC '00, New York, NY, USA, ACM (2000) 47–63
19. Ogata, K.: Modern control engineering. Prentice-Hall, Inc. (1997)
20. Hoogervorst: Enterprise architecture: Enabling integration, agility and change. International Journal of Cooperative Information Systems **13**(3) (September 2004) 213 – 233
21. Pressman, R.S.: Software Engineering, A practitioner's Approach. Third edn. Mc Graw Hill Book Company Europe (1992)
22. Sommerville, I.: Software Engineering. 8th edn. Addison -Wesley (2007)
23. Kandt, R.K.: Software Engineering Quality Practices. Auerbach Publications (2005)
24. IEEE15939, C.S.: Ieee standard adoption of iso/iec 15939:2007, systems and software engineering, measurement process. Technical Report 15939, New York, USA (2007)
25. Kervel, S.: Enterprise ontology driven information system engineering. In: Presentation given at CIAO! (November 2009)
26. Sun, M.I.: Interceptors requirements jsr 318 enterprise javabeans tm. v3.1 Proposed Final Draft (February 2009)
27. OMG: Semantics of business vocabulary and business rules. `http://www.omg.org/spec/SBVR/1.0/PDF` accessed in June 2011 (January 2008)
28. Herwig, M., Verelst, J.: Normalized Systems: Recreating Information Technology based on Laws for Software Evolvability. Koppa (2009)