José Tribolet and Pedro Sousa and Artur Caetano

The Role of Enterprise Governance and Cartography in Enterprise Engineering

Enterprise artography is fundamental to govern the transformation processes of an organisation. The artefacts of enterprise cartography represent the structure and dynamics of an organisation from three temporal views: as-was (past), as- is (present), and to-be (future). These views are dynamically generated from a continuous process that collects operational data from an organisation. This paper defines a set of enterprise cartography principles and provides an account of its role in understanding the dynamics of an organisation. The principles are grounded on control theory and are defined as a realisation of the observer and modeller components of the feedback control loop found on dynamic systems. As a result, an organisation can be abstracted as a dynamic system where a network of actors collaborate and produce results that can be depicted using cartographic maps.

1 Introduction

This paper explores the role played by enterprise cartography and enterprise governance within the enterprise engineering discipline. Enterprise governance relates to enterprise transformation since the change of operational processes, resources and business rules define new management boundaries (Hoogervorst 2009). Enterprise architecture contributes to enterprise transformation as it enables modelling the organisation's structure and dynamics along with the underlying restrictions and design principles (Lankhorst 2013; Op't Land 2009). Transformation is often seen as the set of initiatives that change the organisation's domain from the current *as-is* state to an intended *to-be* state. These two states describe organisational variables at different moments in time. The *as-is* state is defined by the variables that changed due to past events, while the *to-be* state specifies an expected state configuration of the organisational variables. Between these two events, the organisation reacts to other events that are triggered by the operation of the transformation processes.

The issues we address in this position paper focus in the ability to observe and govern the organisation during such transition. This is important because during each transformation initiative an organisation has to react to events. Some of these events may be unrelated to the transformation initiative but may impact the transformation process and therefore deviate the organisation from achieving the planned future state.

This paper presents two contributions. The first is defining enterprise cartography as a function of the *observer* and *modeller* roles as defined by the enterprise's dynamic feedback control loop. Enterprise cartography is not associated with the enterprise design, but with the abstraction and representation of the enterprise reality. Although this differentiates enterprise cartography from enterprise architecture, it may be correctly pointed out that cartography is part of enterprise architecture. But given the relevance of cartography to understand the dynamics of the feedback control loop of an organisation, we opted to discuss the concerns of cartography separately from those of enterprise architecture. The second contribution of the paper is stating the empirical principles that ground the design of the cartography process to play the role of the observer and modeller in the enterprise dynamic feedback control loop. Dynamic systems and enterprise governance are described in Sect. 2 and Sect. 3. Section 4 presents enterprise cartography.

2 Dynamic Systems

The application of systems theory to systems engineering has been discussed since the 1970s (Eriksson 1997; Moigne 1977). Systems theory relates to organisational systems mainly through the principles of dynamic systems, especially control feedback loops (Abraham et al. 2013; Santos et al. 2008). These concepts can be further combined with classic management theories as a means to clarify how feedback loops interact with different organisational views, such as governance, management, and operations (Fig. 1).

In control theory, the *modeller* presents a system view that specifies its current *as-is* state (Levine 1996). The current state makes possible to estimate a future state of the system in the absence of unexpected events. To handle the potential deviations that occur from such events, control theory introduces the concept of *controller*. The *controller* analyses the continuous stream of events and modifies the system's controllable variables as a means of keeping the system behaving as planned (Fig. 2). This is similar to the control of a physical body moving toward a target: the modeller determines the current position and speed of the object and feeds it to the controller; if an unexpected event occurs, then the controller corrects the movement of the object by applying the necessary forces and thereby ensuring that the target is reached.

We argue that the relationships between enterprise governance and enterprise cartography can be established using the principles of dynamic systems feedback control, where cartography plays the aforementioned roles of observer and modeller. These relationships are explained in the next two sections.

3 Enterprise Governance

An enterprise is a network of independent actors. Actors collaborate with other actors along time and thus create a dynamic collaborative network. Actors also produce autonomous behaviour that may change the overall state of the system. Actors can be classified as*carbon-based actors*, i.e. humans, and *silicon-based actors*, i.e. computers. This network runs within a domain where the independent actors behave towards a future state of affairs, and thus produce events, some of which may be unexpected. Therefore, all enterprise domain state changes are a consequence of the individual behaviour of an actor or of the composite behaviour that derives from the actor collaborations. These collaborations may occur between actors that are enclosed by the organisation's boundary, or between an actor that is external to the organisation and one internal actor. So, the behaviour of an enterprise "is" a result of what "it does". An enterprise can therefore be regarded as a large "bionic" distributed network of carbon-based and silicon-based actors that are continuously interacting and producing behaviour.

The current technological advances make possible near real-time, transparent and ubiquitous interaction between people and systems. As such, the boundary between manual, semi-automated and even some automated operations becomes blurred. This means that the actions performed by people cannot be easily separated from those of people supported by a network of computers, and from those of networks of computers. These collaborations can be abstracted as the result of a single network that operates in (near) real-time. The actors that interact within this network act autonomously.

Autonomous behaviour is evident from how a person acts within an organisation since the state change produced by a human actor can only be observed *after* the action is concluded. But the same phenomenon is also observed on information systems because one can only assert what

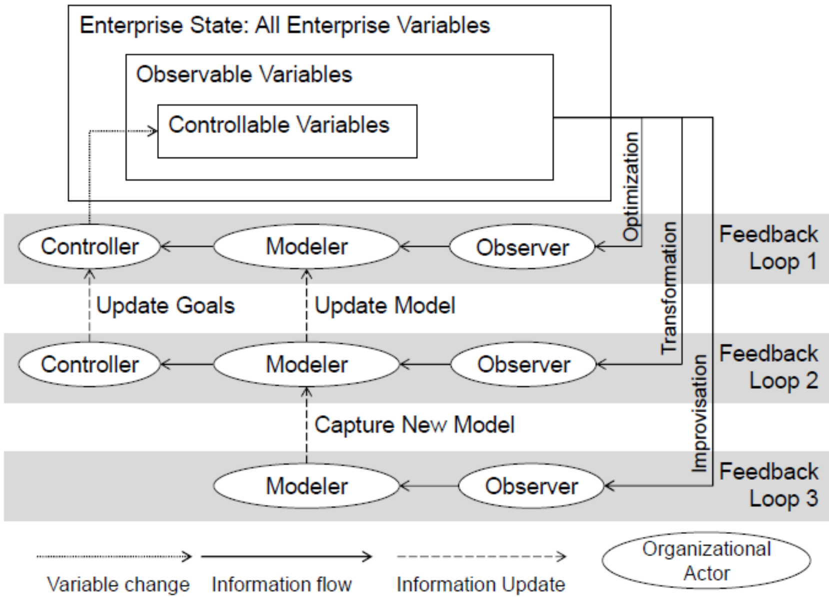

Figure 1: Organisational views and feedback loop, adapted from (Abraham et al. 2013).

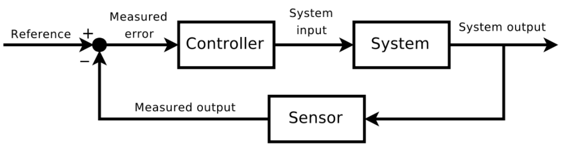

Figure 2: A single-input, single-output feedback loop.

a computer actor has produced after the actual action is performed. The degree of predictability of automated computer actions is potentially higher than that of humans. But achieving certainty is not feasible due to a number of factors. On the one hand, a system may not behave as expected due to faults or failures. And even in the absence of faults of failures, the system may be misaligned with the business. On the other hand, the interaction between multiple systems can produce emergent behaviour, meaning that the overall behaviour of the system may not be the linear sum of each individual behaviour unit. As a result, there is a potential gap between the results that derive from planned actions and the actual results. This makes it impossible to fully estimate the outcome of the interactions in such a network.

This reasoning supports the conclusion that enterprises are dynamic systems. Enterprises are actually a system of systems, composed of and part of other dynamic systems. As such, there is an opportunity to try to understand an enterprise as a complex system through the lenses of systems theory, in particular through the body of knowledge of systems theory and dynamic systems control. However, this application must always consider the intrinsic bionic nature of an enterprise, as people cannot be dissociated from its essence. We defend that all this body of knowledge is directly applicable to enterprises through enterprise engineering. The fundamental purpose of engineering is to provide humans with artefacts that augment their individual and collective capability to deal with specific situations. Engineering helps humans to understand reality and to pro-actively and purposefully transform it as idealised by individual and collective goals. This is the primary purpose of enterprise engineering (Dietz et al. 2013).

How do systems theory and dynamic systems control relate to enterprise engineering? Well, let us start with the “bionic state machine” metaphor presented earlier. According to systems theory, this model can be abstracted as two separate subsystems: a *feed-forward action system*, which is combinatorial in nature and transforms inputs into outputs, and a *feed-back cybernetic system*, which uses as input the state observations and results provided by the feed-forward action system. The feed-back system uses this information to continuous estimate the current state of the system. This is accomplished by contextualising the observations, i.e. by situating the observations into the semantic model of the system. Based on these observations, the feed-back cybernetic system then decides on the actions that all the actors of the system must perform in order to keep the system on a trajectory that achieves its goals. This process is continuously performed. These concepts have been extensively applied to most engineering areas for at least half a century (Andrei 2005).

In this paper we hypothesise that the application of control theory is useful to help understanding enterprise engineering. The next hypothetical principles characterise enterprise governance as a dynamic systems theory problem.

Principle 1 Actions performed by people are enacted by the *feed-forward action* system. People play multiple actor roles within an enterprise such as operational, middle management, knowledge work, auditing, advisory, governance or executive roles. If an enterprise is abstracted as a layered system, all these actions occur at the operational layer, where actual operations are performed by actors. People are abstracted as actors playing roles within well specified semantic domains that uniquely define their contexts of individual action and interaction (Caetano et al. 2009; Zacarias et al. 2005). An actor is capable of playing several roles simultaneously.

Principle 2 A person can be abstracted as a system of systems whenever its actions and interactions occur within the enterprise network. This means that the roles played by people are subject to the rules of the dynamic systems control model. The actions of a person are the result of a combinatorial procedure: a person observes the world, attempts to contextualise and understand its meaning, and then performs an action. This procedure corresponds to the role of *controller*. By acting as a controller, the person can correct the deviations between the current state and the intended state. As such, to achieve goals an human actor operates his own local feed-back subsystem. These actions do not occur at the operational layer but at an higher layer that plans and controls the operations (Abraham et al. 2013).

Principle 3 An enterprise is more than the sum of its actors and resources. Organisational factors such as culture, values, power, and hierarchical structures are elements in defining an enterprise. We abstract these “soft” factors as quality requirements that constrain and parametrise the operating system of an human actor. They are key determinants to the way a human interprets the observations of reality, as well as he reads these observations through his own models of the world, based on which his own sense making operates. These factors have impact on the actions of a human actor since they change how it plays the controller role.

Principle 4 Enterprise self-awareness requires the specification of the domain of action. This is the realms of enterprise governance. Governance actions are distinct from executive, managerial, and operational actions, because they are geared towards the preservation of the enterprise self-awareness. Hence, governance focuses on the design rules and principles that constrain the enterprise actors, along with their actions and interactions.

Principle 5 Maintaining the enterprise as a single entity requires actors to dynamically maintain a view of the actual state of the enterprise.

The previous principles state the relationships between an individual actor and its own dynamic control system. But how do the multiple actors, either carbon or silicon, interact and produce composite behaviour? Using a metaphor: what makes a group of heterogeneous and autonomous musicians become a musical ensemble? Why is this collective entity more than the linear sum of its individual parts? So, what defines the boundary of an enterprise? What forces bind together its autonomous actors as a single entity? We believe that the answer to this question lies in the enterprise's "semantic model of itself". We call this *enterprise self-awareness* (Abraham et al. 2013; Potgieter and Bishop 2003; Santos et al. 2008). This means that if an enterprise has a common semantic model of its actors then in becomes a single collective entity. If there is no common semantic model then the actors are unable to be self-aware of their context and as a result no single collective entity can be defined. This semantic model is a shared dynamic model that is constantly updated by all its active components. It is precisely this shared semantic model that defines a musical ensemble: each musician has its own role, but both individually and as a whole they are self-aware that they share the goal of playing the same piece of music according to a set of rules.

The systemic nature of an enterprise and its cybernetic attributes stress the need for having engineering artefacts to support the collective understanding of its changing reality. Enterprise cartography is fundamental to support this task. Furthermore, this enterprise engineered augmented capability is essential to support the increasing challenges of enterprise governance, which are essential to preserve the integrity of an enterprise as a collective entity. The next section describes the goals of principles of enterprise cartography.

4 Enterprise Cartography

Cartography is the practice of designing and creating maps. It is based on the premise that reality can be modelled in ways that communicate information effectively. Enterprise cartography deals with providing up-to-date model-based views of an enterprise architecture and its goal is facilitating its communication and analysis. We have been successfully applying enterprise cartography concepts to enterprise architecture projects (Caetano and Tribolet 2006; Caetano et al. 2009, 2012b; Sousa et al. 2007, 2009) and developing computer-based tools to support enterprise cartography (Caetano et al. 2012b; Filipe et al. 2011; Sousa et al. 2011). Currently, the principles described here are implemented in a commercial tool that is being used in several medium and large scale enterprise architecture projects[1]. This section describes some empirical findings that we have observed in these cases.

The concept of abstracting reality through representations is not limited to engineering disciplines. Cartography itself is an established discipline that has played a major role in the development of mankind. Cartography is an abstraction process that systematically and consistently transforms an observation of reality into a map or a graphical representation. The production of a map embraces many different concerns, including scientific, technical, and purely aesthetic. Enterprise cartography denotes the discipline that deals with the conception, production, dissemination and study of the maps of an enterprise to support its analysis and collective understanding.

Classic cartography is usually associated with the representation of static objects, as in the case of geographic maps. Modern cartography deals with the representation of both static and dynamic objects and is commonly grounded in information science, geographic information science and geographic information systems. Cartography must also provide multiple consistent views of the same system. For example, geographical maps often combine different views, such as political boundaries, topographic features and several other features. This entails defining

[1] http://www.link.pt/eams/

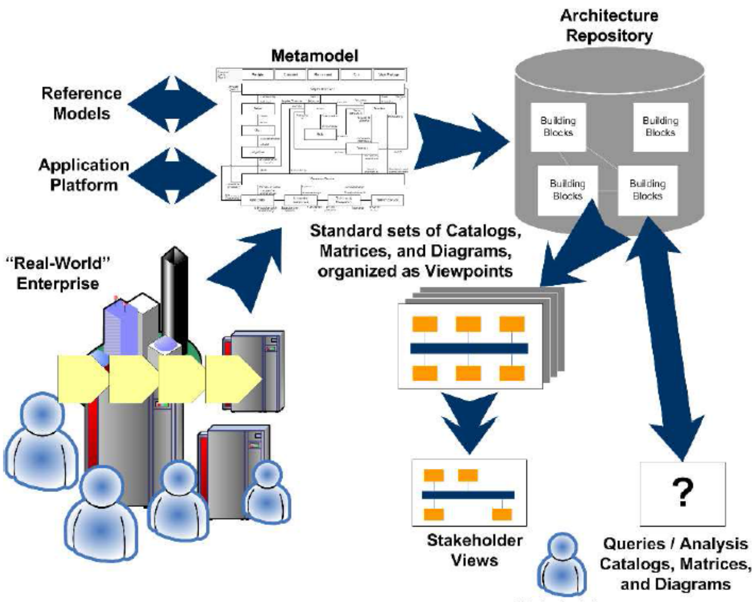

Figure 3: Relationships between meta-model, views, viewpoints, diagrams, and stakeholders, adapted from (The Open Group 2009).

abstraction rules and classification mechanisms so that all of views are consistent. The cartography of dynamic objects also requires to abstract the rules that constrain how objects change and relate to each other over time.

Enterprise cartography deals with the dynamic design and production of architectural views that depict the components of an organisation and their dependencies. It shares its constructs with enterprise architecture, such as meta-models, models, views, repositories, frameworks, and design rules. However, its goal is descriptive. A *view* expresses the architecture of a system from the perspective of concerns defined by its stakeholders. Views are defined by *viewpoints*, which establish the conventions for the construction, interpretation and use of architecture views (ISO/IEC/IEEE 2011; The Open Group 2009). Figure 3), taken from the TOGAF 9 specification, illustrates the basic relationships between these concepts. The following principles distinguish cartography from enterprise architecture.

Principle 1 Enterprise cartography uses observations to produce the representations of an organisation. The process of organisational data collection is a core concern of enterprise cartography. Data collection is not a concern of the mainstream approaches to enterprise architecture.

Principle 2 Enterprise cartography focus on the dynamic description of an organisation. It does not deal with the processes or governance of organisational transformation. The purposeful transformation of organisations is addressed by enterprise architecture.

Principle 3 Enterprise cartography keeps up-to-date architectural views. This implies automated or supervised data collection and view creation. Ideally, these tasks should be performed at the same frequency as that of organisational change. Enterprise architecture techniques do not aim to provide systematic support for data collection nor the automated design and creation of views, meaning these tasks are usually manual and creative.

4.1 Approaches to Enterprise Cartography

There are several approaches to generate organisational models from the data extracted from enterprise systems. Configuration Management Databases (CMDB), as defined by ITIL (Adams 2009), manage the configurations and relationships of information systems and technological infrastructure. To populate a CMDB, some solutions provide auto-discovery techniques that detect nodes, virtual machines and network devices to create infrastructural views. Auto-discovery is actually a cartographic process and requires that the type of the concepts to be discovered is specified in advance (Filipe et al. 2011). The resulting CMDB instance will contain a partial model of the organisation's infra-structure. This model can be communicated through different but consistent visualisation mechanisms, such as textual reports or graphical models that are designed according to a symbolic notation and design rules (Lankhorst 2013).

At the business and organisational layer there are several cartographic techniques defined by business process management (Dumas et al. 2013) and process mining (Aalst et al. 2012). These techniques make use of event logs to discover process activities, control and data flows, as well as organisational structures (Aalst 2011; Aalst et al. 2012; Agrawal et al. 1998). In this case, discovered processes correspond to actual instances of processes, not to the designed processes. Model analysis can also be used to assess the conformance of processes against constraints (Caetano et al. 2012a; Molka et al. 2014). Another example of enterprise cartography is the inference of inter-organisational processes based on EDI event logs (Engel et al. 2012). Semantic technologies, such as ontologies, can also be used to analyse enterprise models (Antunes et al. 2013, 2014). Business intelligence techniques that collect data from organisation systems to produce reports and dashboards are another example of cartography (Negash 2004). Business intelligence actually supports the feed-back control loop by providing managers with a model of the organisation that allows them to ground their actions and decisions.

Enterprise cartography is already a reality in several domains. However, handling dynamic objects, time and change is not explicitly addressed by most approaches. We aim at a generic and systemic approach, very much in line with the concept of "Enterprise Architecture Dashboard" (Op't Land 2009), that displays the enterprise current and future states, its performance and the directions of the organisation transformation process.

4.2 Principles of Enterprise Cartography

This section describes a set of principles that define Enterprise Cartography. These principles use the following definitions.

Project is an transformation process designed to achieve a goal specified by a *to-be state*.

Organisation variable references specific information or a value associated to an organisational artefact.

Organisation state contains the values of a subset of *organisation variables* at a given point in time.

***As-was* state** is the set of all *organisation states* observed in a specific point in the past.

***As-is* state** is the set of *organisation states* as observed in the current point in time.

***To-be* state** is the set of *organisation states* that are predicted to occur in a specific point in the future.

Principle 4 The *as-is* state is defined by the *as-was* and *to-be* states.
Memory of the past state (*as-was*) and the future state (*to-be*) define the behaviour of an organisation. The *to-be* state specifies the goals of transformation projects. Without the *to-be* state the transformation processes cannot be executed or measured since no project goals are defined.

Principle 5 The definition of the *to-be* state always precedes the definition of the *as-is* state.
Organisational artefacts must be always defined as goals in the *to-be* state before being captured in the *as-is* state. This means that the organisational artefacts are not created incidentally but always as the result of a transformation project.

Principle 6 All organisational artefacts can be classified as being in one of four invariant states.
Gestating is the state that describes an organisation artefact after it is *conceived*, i.e. after it starts being planned, designed or produced. At this state, the artefact does not yet exist as an active element of the organisation in the sense it is not yet able to produce behaviour but can be passively used by organisational transactions and processes.
Alive is the state that an artefact enters after *birth*. Birth is the event that signals the moment when a *gestating* artefact enters the *alive* state. This means that the artefact is now able to produce behaviour as part of the organisational transactions and processes.
Dead is when a *gestating* or *alive* artefact is inactive in the sense it is no longer able to play a role in the organisational transactions and processes. This state is the opposite of *gestation* that brought the artefact into existence. However, a *dead* artefact may still have impact on the organisation. For example, an application or server enter the *dead* state when they stop operating and will remain in that state until they are fully retired from the organisational infrastructure.
Retired represents the post-death state where the artefact is unable to further interact with other artefacts.
Organisational artefacts exist first in the *to-be* state and only then in the *as-is* state. This applies to each state transition of the artefact's life-cycle. Artefacts are conceived as the future result of a project, thereby entering the *gestating* state. They remain in this state until the project successfully completes. After that the artefact becomes *alive*. An *alive* artefact *dies* when a decommissioning project completes. A *gestating* artefact can also die if the project is cancelled or not completed. A *dead* artefact is *retired* when a retirement project explicitly removes the artefact from the organisational structure. Therefore, all state changes applying to an artefact are the result of a transformation project. As such, the *to-be* state always precedes the *as-is* state (Sousa et al. 2009).

Principle 7 Organisation models and projects plans are fundamental artefacts.
Organisation models and project plans must be observed as variables whose values are captured during the *as-is* state assessment. This also means that architectural views, viewpoints, models and other architectural artefacts should be regarded as organisation variables. For example, the repository of a UML modelling tool holding the specification of a system under development must be an organisation artefact because it contributes to the specification of the *to-be* state. In contrast, a project is often regarded as an organisation artefact. For instance, both TOGAF and ArchiMate explicitly consider the concept of project Work Package. However, organisational models, viewpoints and views are not explicitly regarded as artefacts by enterprise architecture modelling languages. Nonetheless, system architecture guidelines such as ISO 42010 point out the importance of considering these elements as system artefacts (ISO/IEC/IEEE 2011).

Principle 8 The *to-be* state is sufficient to plan a transformation project.
For the purpose of planning a transformation project the current *as-is* state is not required because the *to-be* state must fully specify the organisational goals.

4.3 Discussion

Figure 4 depicts a time line and a series of events in time (T0-T5). T0 represents the current moment, therefore indicating the instant the *as-is*

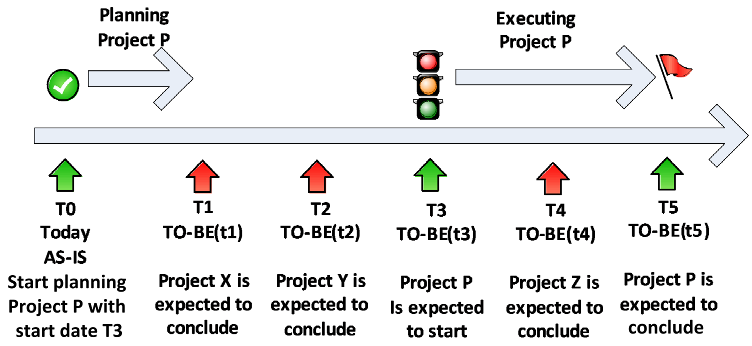

Figure 4: Project planning and execution.

state was captured. At T0 the project P is conceived and enters the gestating state: this project is planned to start at T3 and to be completed at T5. Events T1, T2, T4 signal the completion of projects X, Y and Z, respectively. Therefore, T1, T2, T4 also indicate that the artefacts that were produced by these three projects became *alive*. Since project P is planned to start at T3 the organisation requires knowing about its state at state *to-be*(T3) and not at state *as-is*(T0) although planning is actually taking place at T0. This happens because the completion of projects X and Y at T1 and T2 may interfere with the execution of P at T3. Furthermore, the organisation also requires knowledge about its state at T4 because the changes resulting from project Z may also interfere with project P.

To plan a transformation initiative an organisation needs to be aware of the set of *to-be* states while the project is being executed. A description of the *as-is* state for planning purposes is actually of limited use because there is often a temporal gap between project planning and project execution. On the other hand, other projects conclude and change the organisation state while the project stands between planning and executing. These observations minimise the relevance of the *as-is* state as a means to design the transformation processes of the organisation.

As an example, consider an organisation that plans the replacement of a system in 6 months time and starts today the corresponding project plan. The project planning phase must have an understanding of the dependencies between that system and other systems, as well as to the business processes it supports. If no state changes occur in the next 6 months, then the organisation can indeed rely on the *as-is* state to plan the replacement project. But if the organisation is performing a set of additional transformation projects that will change the organisation's state during that period, then planning the system replacement project will require knowing about the sequence of *to-be* states during the next 6 months and during the actual execution of the replacement project. Otherwise, it will not be possible to plan according to the actual network of dependencies between the system to replaced and other organisational artefacts. Therefore, for the purpose of project planning and execution, the current *as-is* state will often not mirror the organisation's reality. In fact, the relevance of the *as-is* state is inversely proportional to the number of projects being completed per unit of time. At the limit, all dependencies of the system to be replaced may change between the planning and execution phases, meaning that all *as-is* state variables will become irrelevant for planning purposes.

Nevertheless, the knowledge about an organisation's current state is a fundamental asset for its operational management. At operational level, actions and reactions are based on near real-time observations and events, meaning that planning and execution occur in close sequence. However, the requirements of the near real-time operational level of an organisation should not be intertwined with the medium to long-range requirements required for organisational transformation and governance.

5 Conclusions

Organisations do plan and execute projects, regardless of not having a full or accurate representation of the *as-is* or *to-be* states. Such an accomplishment implies that projects include to some degree an assessment of the impact of change between and during planning and execution.

An organisation that does not have a representation of its *to-be* state will be unable to create a detailed plan of project P as depicted in Fig. 4. This means that parts of the plan must be postponed until T3 to minimise the gap between the planning and execution of P. This reality is commonly observed in many organisations despite having impact on the project costs and risk, and staff assignment. It also interferes with the planning of other projects, thereby having negative impact on the organisation's agility. To remedy this issue, enterprise architecture projects often attempt to obtain a complete and accurate representation of the *as-is* state. As a result, the primary goal of these projects is an attempt to keep an organisational repository updated with an observation of the *as-is* state. This approach is often justified by statements such as "knowing in detail where we stand today is a pre-requirement to any transformation project." Although this sounds wise, this is a demanding task in terms of effort and time. Moreover, and as discussed before, the rate of organisational change will make the *as-is* state obsolete for the purpose of transformation planning. Therefore, we posit that organisations should reassess the actual value of enterprise architecture projects that aim capturing the *as-is* state as an enabler of transformation planning.

This dilemma is found in many organisations: the contrast between the notion that an *as-is* assessment is a valuable asset for organisational transformation, and knowing at the same time that achieving such continuous task is demanding. This paper defends that an organisation does not need to have a full and accurate depiction of the *as-is* state but of its *to-be* state. The *to-be* state is specified according to the specific goals of projects, that are required for planning purposes. This contrasts with the *as-is* state that requires observing the variables of all organisational artefacts that are not retired. Consider a project that aims creating a new system that will interact with an existing legacy system. Planning this project requires collecting information about the legacy system as well about the design of the new system. However, the task of collecting information about a legacy system for the purpose of project planning is actually contributing to extending the knowledge about the current state of the organisation. This is a potential avenue to sort out the dilemma stated earlier because a representation of the *as-is* state can be built incrementally by specifying the *to-be* state(s) that are required to plan the multiple projects of an organisation.

This position paper has presented a general framework that provides representations of dynamic organisations in the context of enterprise engineering. It specifically describes a set of principles grounded on dynamic systems theory that provide guidelines on how to represent a cartographic representation of an organisation. Such representations facilitate the planning of organisational transformation.

References

van der Aalst W. (2011) Process mining: discovery, conformance and enhancement of business processes. Springer van der Aalst W., Adriansyah A., de Medeiros A. K. A., Arcieri F., Baier T., Blickle T., Bose J. C., van den Brand P., Brandtjen R., Buijs J. et al. (2012) Process mining manifesto. In: Business process management workshops. Springer, pp. 169–194

Abraham R., Tribolet J., Winter R. (2013) Transformation of multi-level systems–theoretical grounding and consequences for enterprise architecture management. In: Advances in Enterprise Engineering VII. Springer, pp. 73–87

Adams S. (2009) ITIL V3 foundation handbook Vol. 1. itSMF International, Office of Government Commerce, ISBN: 978-0113311972

Agrawal R., Gunopulos D., Leymann F. (1998) Mining process models from workflow logs. In: Lecture Notes in Computer Science Volume 1377. Springer, pp. 467–483

Andrei N. (2005) Modern Control Theory: A historical perspective. Center for Advanced Modeling and Optimization, Romania

Antunes G., Caetano A., Bakhshandeh M., Mayer R., Borbinha J. (2013) Using Ontologies to Integrate Multiple Enterprise Architecture Domains. In: Abramowicz W. (ed.) Business Information Systems. Lecture Notes in Business Information Processing Vol. 160. Springer, pp. 61–72

Antunes G., Bakhshandeh M., Mayer R., Borbinha J., Caetano A. (2014) Using Ontologies for Enterprise Architecture Integration and Analysis. In: Complex Systems Informatics and Modeling Quarterly 1

Caetano A., Assis A., Tribolet J. (2012a) Using Business Transactions to Analyse the Consistency of Business Process Models. In: 45th Hawaii International Conference on System Science, pp. 4277–4285

Caetano A., Tribolet J. (2006) Modeling organizational actors and business processes. In: Proceedings of the 2006 ACM Symposium on Applied computing. ACM, pp. 1565–1566

Caetano A., Silva A. R., Tribolet J. (2009) A role-based enterprise architecture framework. In: Proceedings of the 2009 ACM Symposium on Applied Computing, pp. 253–258

Caetano A., Pereira C., Sousa P. (2012b) Generation of Business Process Model Views. In: Procedia Technology 5, pp. 378–387

Dietz J. L., Hoogervorst J. A., Albani A., Aveiro D., Babkin E., Barjis J., Caetano A., Huysmans P., Iijima J., Kervel S. J. V. et al. (2013) The discipline of enterprise engineering. In: International Journal of Organisational Design and Engineering 3(1), pp. 86–114

Dumas M., Rosa M. L., Mendling J., Reijers H. A. (2013) Fundamentals of Business Process Management, 2013 Edition. ISBN 978-3642331428. Springer, p. 350

Engel R., van der Aalst W. M., Zapletal M., Pichler C., Werthner H. (2012) Mining inter-organizational business process models from EDI messages: A case study from the automotive sector. In: Advanced Information Systems Engineering. Springer, pp. 222–237

Eriksson D. M. (1997) A Principal Exposition of Jean-Louis Le Moigne Systemic Theory. In: Cybernetics & Human Knowing 4(2), p. 42

Filipe J. A. C. B., Coelho M. F. P., Ferreira M. A. M., Figueredo I. C. (2011) Social Responsibility and Environmental Sustainability: The Case of Caixa Geral de Depósitos (Portugal) and Vale (Brazil). In: Editorial Board Members 10(5), pp. 352–375

Hoogervorst J. A. (2009) Enterprise governance and enterprise engineering. Springer

ISO/IEC/IEEE (2011) 42010. Systems and Software Engineering: Architecture Description

Lankhorst M. (2013) Enterprise architecture at work: Modelling, communication and analysis. Springer

Levine W. S. (1996) The Control Handbook Levine W. S. (ed.). CRC Press

le Moigne J.-L. (1977) La theorie du systeme general. Theorie de la modelisation. Collection les classiques du reseau intelligence de la complexite

Molka T., Redlich D., Drobek M., Caetano A., Zeng X.-J., Gilani W. (2014) Conformance Checking for BPMN-Based Process Models. In: Proceeedings of the 29th ACM Symposium

On Applied Computing

Negash S (2004) Business Intelligence. In: Communications of the Association of Information Systems 13, 177â195

Op't Land M. (2009) Enterprise architecture: creating value by informed governance Springer (ed.). Springer

Potgieter A., Bishop J. (2003) The Complex Adaptive Enterprise: Self-Awareness and Sustainable Competitive Advantage using Bayesian Agencies, In: The Journal of Convergence, UCT Graduate School of Business

Santos C., Sousa P., Ferreira C., Tribolet J. (2008) Conceptual model for continuous organizational auditing with real time analysis and modern control theory. In: Journal of Emerging Technologies in Accounting 5(1), pp. 37–63

Sousa P., Pereira C., Vendeirinho R., Caetano A., Tribolet J. (2007) Applying the Zachman Framework Dimensions to Support Business Process Modeling. In: Digital Enterprise Technology. Springer, pp. 359–366

Sousa P., Lima J., Sampaio A., Pereira C. (2009) An approach for creating and managing enterprise blueprints: A case for it blueprints. In: Advances in Enterprise Engineering III. Springer, pp. 70–84

Sousa P., Gabriel R., Tadao G., Carvalho R., Sousa P. M., Sampaio A. (2011) Enterprise Transformation: The Serasa Experian Case. In: Practice-Driven Research on Enterprise Transformation. Springer, pp. 134–145

The Open Group (2009) TOGAF, the open group architecture framework, version 9. Van Haren Publishing

Zacarias M., Caetano A., Pinto H., Tribolet J. (2005) Modeling Contexts for Business Process Oriented Knowledge Support. In: Althoff K.-D., Dengel A., Bergmann R., Nick M., Roth-Berghofer T. (eds.) Professional Knowledge Management. Lecture Notes in Computer Science Vol. 3782. Springer Berlin Heidelberg, pp. 431–442

José Tribolet

Department of Computer Science and Engineering
Instituto Superior Técnico
University of Lisbon
Av. Rovisco Pais
1049-001 Lisboa
Portugal
jose.tribolet@tecnico.ulisboa.pt

Pedro Sousa

Department of Computer Science and Engineering
Instituto Superior Técnico
University of Lisbon
Av. Rovisco Pais
1049-001 Lisboa
Portugal
and
Link Consulting
Avenida Duque de Avila 23
1000 Lisboa
Portugal
pedro.sousa@tecnico.ulisboa.pt

Artur Caetano

Department of Computer Science and Engineering
Instituto Superior Técnico
University of Lisbon
Av. Rovisco Pais
1049-001 Lisboa
Portugal
and
INESC-ID
Information Systems Group
Rua Alves Redol 9
1000-029 Lisboa
Portugal
artur.caetano@tecnico.ulisboa.pt