Towards the Model Driven Organization
Introduction to the 1st AMINO Workshop
Held at the ACM/IEEE 16th International Conference on Model Driven Engineering Languages and Systems MODELS 2013 Miami Florida USA

Balbir Barn[1], Tony Clark[1], Robert France[2], Ulrich Frank[3], Vinay Kulkarni[4], and Dan Turk[2]

[1] Middlesex University, London, UK
[2] Colorado State University, Colorado, USA
[3] Universty of Dusibirg-Essen, Essen, Germany
[4] Tata Consultancy Services, India

Organizations such as banks and public sector institutions can be thought of as being structured in three key layers. The *strategic* layer defines what an organization must achieve in terms of its high-level *goals*, the *tactical* layer defines how an organization plans to behave and thereby achieve its goals, and the *operational* layer defines the day-to-day running of the organization in a manner that is consistent with the organization's plans. The operational layer of a modern organization is implemented in terms of a collection of inter-connected IT systems that form an *organizational platform.* An organization seeks to *align* its high-level goals with its platform so that its strategy is properly supported by its IT infrastructure. Expressing and achieving alignment remains a key challenge for modern organizations.

Essentially organizations exist to accomplish specific goals. In the case of a *not-for-profit* organization, the goals are often expressed as a *mission statement.* In the case of a *for-profit* organization, the mission should also include achieving profit for investors. Frequently these goals are not consciously or explicitly written down or even clearly recognized. But, even so, they still exist. Whether goals are explicitly identified and documented or not, organizations wish to know whether the goals are achieved and, where possible, concrete metrics may be used to help answer these questions.

Whilst top-level organizational goals can often be captured as relatively simple mission statements, the internal processes, resources, IT systems and structures that are put in place to realise the goals are usually highly complex. Measuring aspects of a typical large multi-national corporation in order to determine its internal consistency, to measure quality metrics, to perform any number of change-based activities, and ultimately to establish consistency between the organizational goals and the human- and IT-centric processes by which it operates, is a difficult task.

In the field of software system development, it is accepted that various forms of modelling need to be brought to bear on the problem of measuring complex system properties, controlling its behaviour and to establish that an implementation satisfies its specification. The key feature at play is *abstraction* where unnecessary detail is elided leaving important aspects of a system to be represented in a language that is suitable for the stakeholders and exhibits appropriate properties for reasoning, transformation, simulation *etc.* Our claim is that the same use of abstraction through modelling can be applied to the problems of managing an organization, leading to the idea of a Model-Driven Organization:

> **Def:** A *Model-Driven Organization* (MDO) is an institution that uses models, both formally and informally, in conscious and intentional ways, throughout the organization to define who it is and what it does, to help better and more quickly train employees, to carry out, assess, and improve its functioning, and to more effectively develop and modify systems that support the organization.

The use of models in system development has established the fields of Model Driven Architecture (MDA) and Model Driven Engineering (MDE). Although these fields use models as abstractions of systems in order to establish properties, they predominantly use models to help with system construction during the design phase. In some cases part or all of the program code for a system is generated from models.

Although MDA/MDE may be viewed as providing a contribution, the MDO vision is much broader than system development. An MDO uses models at multiple levels: Organizational level models, not just system / software level models; conceptual, as well as technical models; strategic, tactical, and operational models. MDO models relate to much more than just the IT systems of an organization and need not be expressed in a formal language. The models must address people-centric aspects of an organization in addition to the systems-level aspects. We can see this by comparing the definition of the MDO as given above, with the following two common definitions for MDE and MDA:

> **Def:** *Model-Driven Engineering* (MDE) is a software development methodology which focuses on creating and exploiting domain models (that is, abstract representations of the knowledge and activities that govern a particular application domain), rather than on the computing (or algorithmic) concepts.

> **Def:** *Model-Driven Architecture* (MDA) is a software design approach for the development of software systems. It provides a set of guidelines for the structuring of specifications, which are expressed as models. Model-driven architecture is a kind of domain engineering, and supports model-driven engineering of software systems. It was launched by the Object Management Group (OMG) in 2001.

Our claim is that a model-based approach can be used in many ways within an organization. Models can be used to describe the organization itself, its structure, its processes, its business rules, its regulatory constraints, *etc.* This can help in a wide variety of organization-centric use-cases including training new employees and helping outsiders better understand and effectively interact with the organization. These models may be able to help potential investors compare organizations, or even make it easier to merge organizations. They may make it easier to compare the functioning of organizations, or systems within organizations, to allow decision-makers to better determine which parts of merged organizations to keep and which to eliminate after a merger takes place, for instance. Models may be used to help in the analysis of an organization, to determine and clearly identify its mission, and assess how well it is meeting its goals.

If the models are formalized and automated, it may be possible to ease and/or automate some types of changes within an organization. And, of course, models have had a long history of helping to automate parts of organizations.

Achieving the MDO vision will involve expertise from many different disciplines. Experts in modelling will be required to design suitable languages both domain-specific and general purpose using appropriate meta-technologies. Expertise in model processing including transformation, synchronization and model-checking will be required. Experts in the field of Organization Theory will be required to understand the structures and processes to be modelled. Information Systems experts will be required to analyse and represent the ontologies and complex data used within an organization. Management theorists will be required to understand the human-centric aspects of the problem and where models can be used to facilitate all stakeholders. Experts in tools will be required to determine how to offer the features of the MDO in an effective way.

The goals of the AMINO workshop are to work towards a better understanding of where models can be used to address all aspects of organizational development, operation and management use-cases. The workshop took the form of an invited speaker and presentations of 7 peer reviewed papers. The rest of this introduction provides an overview of the presentations.

In *Goals, Domains, and Enterprise Architecture in the Model-Driven Organization* Desmond D'Souza argues that goal models are fundamental to achieving the MDO. He presents a notation and method for incremental model construction in terms of goals and their constraints over domains. The paper concludes by indicating how goals can be exploited in other MDO areas including architectures and migration plans.

In *Multimodel-Driven Software Engineering for Evolving Enterprise Systems* Richard Paige, Radu Calinescu, Dimitrios S. Kolovos, Nicholas Matragkas and Dave Cliff address the issue of organizational evolution and how to analyse the Quality of Service (QoS) issues that arise. The paper proposes a multimodel-driven approach (MMSE) and concludes with some thoughts about a research agenda that will lead to solutions in this area.

Organisations consist of many different roles that link to business processes and provide access to information at various security levels. In *UML/OCL based Design and Analysis of Role-Based Access Control Policies* Oliver Hofrichter, Martin Gogolla, and Karsten Sohr discuss an approach to modelling the access control and use OCL to address a case study based on EasyChair.

As discussed above, a key feature of the MDO is goal-IT alignment. In *Metamodel Patterns for Expressing Relationships Between Organization Model Concepts and Software Implementation Concepts* Jens Gulden discusses this issue and provides patterns for mapping between the two levels.

The state-of-the-art in Enterprise Modelling and Enterprise Architecture uses enterprise frameworks to represent and reason about aspects of an organisation or even its entirety. Few of these frameworks use modelling technologies and techniques. In *MDE Support for Enterprise Architecture in an Industrial Context: the TEAP Framework Experience* Hugo Bruneliere, Jordi Cabot, Stphane Drapeau, Flavien Somda, William Piers, Juan David Villa Calle and Jean-Christophe Lafaurie describe a new framework called TEAP that applies Model Driven Engineering techniques to the construction of an enterprise framework.

In many ways organisations are more complex that standard software systems and the structures, information, resources and processes involved are less precise. Therefore it is important to understand the reasoning behind design decisions and the reasons for organisational change. In *Introducing Argumentative and Discursive Enterprise Leading and Management* Sebastian Bittmann, Balbir Barn, Tony Clark and Oliver Thomas develop this theme in terms of *argumentation theory* as a basis for recording the intentions behind organisation-related actions.

There are many different organizational use-cases involved in achieving an MDO. One is *mergers and acquisitions* whereby two organisations become one. In *(Multi-) Modeling Enterprises for Better Decisions* Sagar Sunkle, Vinay Kulkarni, and Hemant Rathod argue that in order to represent and reason about such complex use-cases it is necessary to use multi-models.

The MDO is likely to involve the use of models in many different ways. Some of these may simply be to help understand the organisation, but others may help to operationalise it. In *Enterprise Models as Drivers for IT Security Management at Runtime* Anat Goldstein and Sietse Overbeek discuss the technique of models@Runtime and its use to achieve IT security.