CoreEAF – a Model Driven Approach to Information Systems

Tomas Jonsson[1] and Håkan Enquist[2]

[1] Genicore AB, Göteborg, Sweden
tomas@genicore.se,
WWW home page: http://www.genicore.se
[2] IT University, Göteborg, Sweden
enquist@chalmers.se
WWW home page: http://www.ituniv.se/english

Abstract. Model driven IT systems development with code generation is today used in several, mostly technical, application areas. However, for large IT based Information Systems (ITbIS) it is still quite uncommon. An innovative case of low cost and high quality ITbIS is presented along with the model driven framework applied. This innovation is made possible by a coherent model driven approach with integrated method and tool support for the complete development and maintenance cycles of an ITbIS. For over 20 years and still, the FMV ERP system has been designed, extended and modified in pace with changes in the organization as well as disruptive changes in information technology.

Keywords: Business Information System, Business Knowledge Model, Domain Model, Model Driven Design, Innovation, Information Engine

1 IT based Information Systems (ITbIS) Development

Large IT based Information System (ITbIS) projects frequently miss targets or fail completely [1] [2]. In Europe, the cost of missed targets and failures was estimated to 142 Billion Euros per year[3]. Obviously, methods in current use are not sufficient to solve the task. There are only a few examples of large ITbIS being built with a model driven approach [4] although very promising results are shown by some model driven approaches [5] [6].

Core Enterprise Architecture Framework (CoreEAF^{TM}) – a model driven ITbIS framework – comprises method, tools and platform for building and executing ITbIS from models. CoreEAF represents an interdisciplinary approach to ITbIS design, combining theories from disciplines such as management [7], information theory [8] [9], cognition [10] [11] and computer science [12].

Focus in this paper and demo is on illustrating the unbroken chain of support for model driven design and execution of ITbIS, leaving in-depth descriptions of each enabling concept and its implementation to be presented in the future.

The framework Fig. 1, is designed to produce ITbIS with a Model View Architecture, as originally proposed by Reenskaug [13]. The Business Knowledge

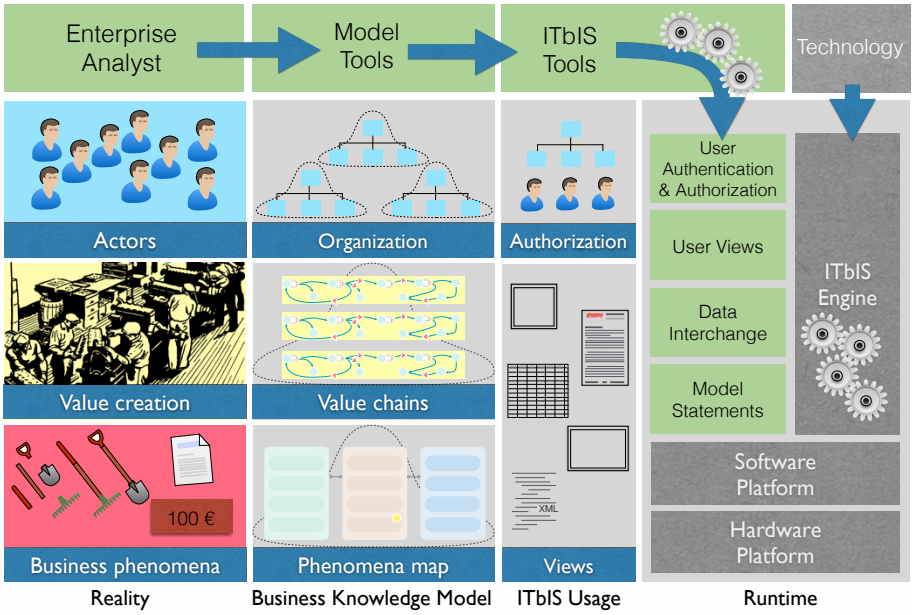

Fig. 1. Core Enterprise Architecture Framework

Model (BKM) is from a business perspective, formalized business knowledge. From a technical perspective, the BKM is an object oriented declarative model describing information structure, information processing and information rules, including information access rules. Views are defined declaratively and connected to model elements, which in runtime operates as windows for accessing information in the ITbIS. There are different types of views: graphical user interface; word processor interface; document generation; report generation and data interchange views. Each type of view is described either with a tool or a language.

Building an ITbIS with CoreEAF is done solely by declaring a BKM and it's views, excluding programming in the traditional sense.

2 ITbIS Case – ERP System for Defence Material Acquisition

The Swedish Defence Material Administration's (FMV's) Enterprise Resource Planning (ERP) system called FMV-Core, Fig. 2, is the most extensive system out of five, built and executed with the current version of CoreEAF. Applying the CoreEAF for FMV-Core is by now a proven innovation for providing ITbIS to an organization at low cost, low error rate, short lead time and high requirement fulfillment. This innovation is made possible by coherent design support for consistent model driven development and change actions throughout the ITbIS

life cycle, along with a coherent runtime environment for ITBIS execution with full consistency between models and target system behavior.

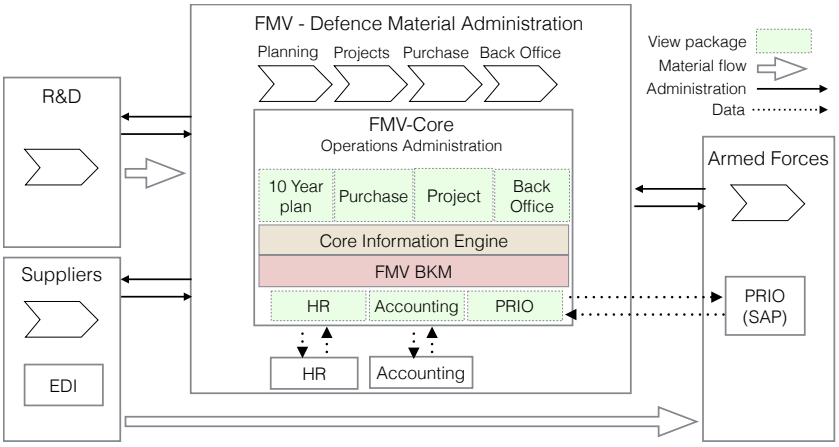

Fig. 2. Organizational structure and FMV-Core

FMV has ongoing material acquisition projects worth about 6 Billion Euros and a turnover of 2 Billion Euros per year. FMV-Core is an ITbIS with 1.200 users and handles information in FMV's core business. This includes project planning, project execution and accounting as well as complex purchasing processes from RFQ, tenders, complex contract drafting to delivery and payments. FMV-Core is integrated with several other IT systems, one of which is the Swedish Armed Forces SAP based ERP system (PRIO), using the integration facility CoreCom of the CoreEAF information engine.

Regarding design metrics, the FMV-Core system is defined with 33.000 declarative statements, whereof 13.000 statements define the BKM and 20.000 define views. FMV-Core BKM includes 230 business phenomena with 7 levels of generalization and 400 relations. The phenomena altogether carry 3000 value attributes all managed by 5600 rules and calculations.

This level of content complexity implemented in e.g. standard systems would typically include millions (>1.000.000) of lines of program code. For instance SAP is built with more than 300.000.000 lines of code [14].

3 Modeling the Enterprise – Method, Language and Tools

The key concepts applied to implement coherent support for consistent model driven design of ITbIS are described below.

3.1 Business Phenomena – Understanding the Reality of an Organization

The first and most profound aspect is the business phenomena aspect. The model of business phenomena shall directly and only directly correspond to actual business phenomena and be labelled with the business terminology.

Concrete as well as abstract business phenomena such as agreements, projects, services, etc. need to be understood and documented in the model. It is often challenging to understand and document abstract phenomena as they only exist in the minds of people and are constantly undergoing changes and redefinitions. Thus, business phenomena modelling requires fundamental understanding of human perception and of how to capture human knowledge into formal models using a modelling language.

3.2 Value Creation – What is Being Achieved

The second aspect is the value creation aspect, documented with value chains, as in Fig. 3.

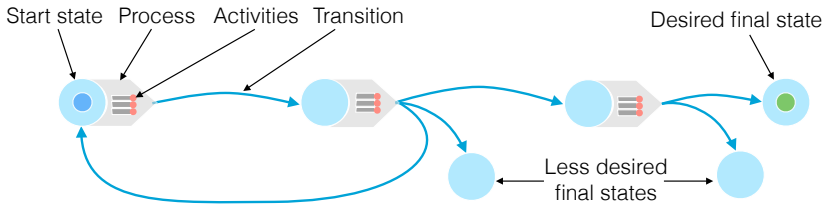

Fig. 3. Value Chain Diagram

Some of the business phenomena can be considered more or less static, i.e., they exist in the business and do not undergo any transformation of value. Other phenomena undergo value transformations, such as products being assembled in a factory. Also intangible phenomena undergo value transformations and are most likely the only kind of value transforming phenomena in organizations that produce something other than tangible products.

Examples of value transforming intangible phenomena are business agreements such as sales agreements or purchase agreements. In the public sector we can find value transforming phenomena such as claims and license applications.

For each value transforming phenomenon, a state diagram is defined representing the value chain of the phenomenon. The value chain has a start state and typically one or more final states, where one of the final states represents the desired final state of the value transformation. The intermediate states represent steps towards and sometimes away from the desired final state.

E.g. the value chain of a sales agreement can be composed of the states suggested, quoted, rejected, ordered, delivered, returned, paid and refunded, where *suggested* is the start state and *paid* is the desired final state. Note that a state transition such as *quoted* to *ordered* represents a positive value progression and *quoted* to *rejected* represent a negative value progression.

Connected to each state is a set of activities to be performed in order to reach one of the next states of the value chain. E.g. the transition *ordered* to *delivered* could be connected to several activities such as picking items in warehouse, package items, attach address label, give to post office, etc.

3.3 Organizational Structure – Roles, Responsibilities and Actors

The third aspect is the organizational structure. Business roles with responsibilities are added and connected to value chains of value transforming phenomena in the model.

Business roles can have two different kinds of responsibilities in relation to the value chains. Firstly, a role can have the overall responsibility for a certain phenomenon to progress to certain states. Secondly, for performing activities moving the phenomenon closer to a following state in the value chain. Roles can further be grouped into organizational structures representing e.g. management structure, team structure etc., depending on phenomenon, activity and organizational policies.

Finally, actors can be connected to the roles. Actors are either individuals e.g. staff members of the organization, customers, suppliers, etc. or machines performing some activity. Machines could be mechanical machines or IT systems.

3.4 Core Model Language (CML) – How to Describe BKM

Core BKMs are created and maintained in modelling tools allowing graphical editing and navigation of the model. The tools are based on a declarative language, Core Model Language (CML)

CML formalism follows the basic principles of a strongly typed class based object oriented (OO) language. However, CML is declarative and all data processing is described with parameterless functions without side effects (expressions). I.e. the value of an attribute can only be set by the attribute's value function, not by instructions in several different methods in the class. This means that each calculated value of the system is clearly defined in one place and one place only, usually as a single line of declarative statement. This gives a great advantage in terms of changeability, predictability and fault localisation. A change in one expression will only affect the values of one attribute or if a value of an attribute is incorrect, the problem will be located in its value expression.

In CML the OO *class* concept is called *phenomenon*.
Data container: A phenomenon has attributes which when instantiated can hold values.

Generalization: A phenomenon can be part of a generalization – specialization hierarchy.
Encapsulation: Stronger than in classic OO since attribute values can only be set by the attribute's value function, not by other components of the phenomenon (class).
Polymorphism: An attribute can have different definitions in a generalization – specialization hierarchy.

Additional structural components have been added to the traditional OO concept, such as relation, view and change permissions, value chain, activity, role and organizational entity.

Examples of attribute and value declarations (identifiers in ' ')
phenomenon 'person'
attribute 'full name' **value** 'given name' + " " + 'family name'
phenomenon 'order'
attribute 'order total' **value sum** 'items' [**not** 'complimentary'].'price'

The language, as shown, defines data flow graphs not execution order. It is up to the underlying information engine to resolve execution order depending on events in the data flow graphs.

3.5 Tools for Enterprise Modeling

There are two tools which can be used for modeling as described in sections 3.1-3.4. They fit in to the CoreEAF as shown in Fig. 4.
Browser: WEB based environment for creating and viewing models.
Builder: Windows based environment for creating and viewing models.

4 Modeling the ITbIS – Tools and Languages

Once the business model is created it is time to design an ITbIS that meet the information management needs of the organization and its people.

Designing the ITbIS essentially means modeling various views of the BKM. For this purpose various tools and languages are used, which assumes and is connected to the BKM. The design tools and languages architecture is shown in Fig. 4.

AppBuilder: Tool for building a component based graphical user interface, connected to a model, i.e. an application.
Reporter: User tool for creating reports with flexible search and result criteria.
IDoc-L: XML based language to define document oriented interactive user interface using MS-Word as platform.

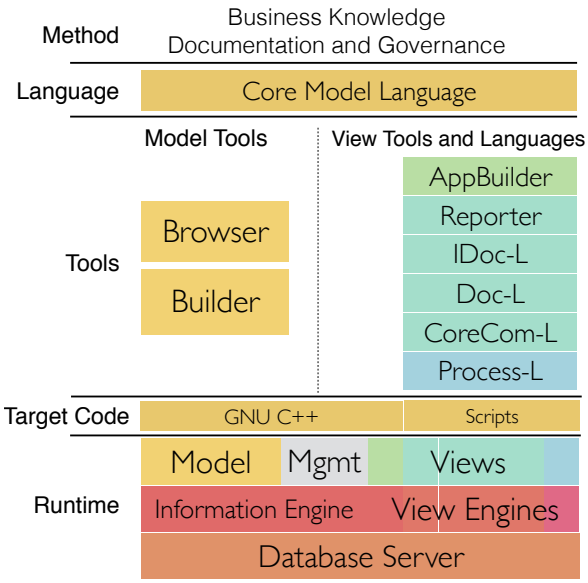

Fig. 4. Method and Tool architecture

Doc-L: XML based language to define paper documents from model data.
CoreCom-L: XML based language to define import/export of model data in XML or other text formats.
Process-L: XML based scripting language to monitor and modify data in the model based system.
Target Code The tool chain is based on C++ and executable on Linux and MS-Window. The script languages are CoreEAF specific.

5 ITbIS Runtime Components

Information Engine: The Information Engine handles persistence, information consistency, serving multiple users with real-time information while maintaining transactional integrity over a distributed network of resources, information security and a timeline for each phenomenon and its context. The information engine is programmed in C++ and uses an SQL database engine for data storage.
Applications: Model, information engine, view engines and views created by AppBuilder are compiled together into user applications. In an ITbIS there can be several different applications, where all applications have the same model but different views.
Dynamic data input and output: There are four dynamic view engines active

in runtime. One view engine which interprets report definitions and three view engines which run XML defined scripts one for each type of view tool: IDoc-L, Doc-L, CoreCom-L.
Management (Mgmt): Component for administrating users roles in the organization and for authorization in the ITbIS.

6 Current Status and Future Developments of CoreEAF

Ongoing research and publication on Model Driven ITbIS Design (MDID) include a case study and subsequent thematic studies on FMV Core.

A community for collaboration on knowledge development on MDID is being formed with academia, industry and public sector. The community will also provide knowledge resources and a web based platform for tools and experimentation.

Genicore is currently developing a new web based modeling tool with a lightweight information engine and auto generated UI for MDID to enable education and training.

References

1. L. Laird and C. Brennan : Software Measurement and Estimation A Practical Approach. IEEE Computer Society / John Wiley & Sons (2006)
2. The Standish Group: The CHAOS Manifesto. The Standish Group (2013)
3. McManus, J. & Wood-Harper T. : Understanding the sources of information systems project failure. Management Services, 51(Autumn), 3843 (2007)
4. Enquist, H. & Jonsson, T. : Sammanställning av information om Model Based System Development (MBSD) för informationssystem typ affärssystem. Research report, Genicore AB, Gothenburg (2013)
5. Pawson, R. : Naked Objects. Thesis, University of Dublin, Trinity College (2004)
6. Whittle J., Hutchinson J. Rouncefield M. : The State of Practice in Model-Driven Engineering. Software, IEEE Volume: 31, Issue: 3, Page 79 - 85 (2014)
7. Mintzberg H. : Structure in fives, designing effective organizations. Englewood Cliffs, N.J. Prentice-Hall (1983)
8. Langefors B. : Essays on Infology. Studentlitteratur: Lund (1995)
9. Enquist H. Makrygiannis N. : Understanding Misunderstandings. HICSS 82480083 (1998)
10. Hauser M.D., Chomsky N., Fitch W.T. : The faculty of language: what is it, who has it, and how did it evolve? Science vol 298 p.1569-79 (2002)
11. Kohonen T. : Associative and self organizing memory. Springer-Verlag (1988)
12. Lindskov Knudsen J., Lofgren M., Lehrmann Madsen O., Magnusson B. Et. al. : Object-Oriented Environments - The Mjølner Approach. Prentice Hall (1993)
13. Reenskaug T. : THING-MODEL-VIEW-EDITOR an Example from a planningsystem. Xerox PARC technical note. (1979)
    http://heim.ifi.uio.no/~trygver/1979/mvc-1/1979-05-MVC.pdf
14. Vishal Sikka, Keynote (2008)
    http://scn.sap.com/docs/DOC-14735