A professional Case Management System in production, modeled and implemented using DEMO

John Hintzen[1], Steven J. H. van Kervel[1], Tycho van Meeuwen[1],
Joost Vermolen[1], Bob Zijlstra[1],

[1] Formetis Consultants BV, Boxtel, The Netherlands,
info@formetis.nl

Abstract. This case study describes a real world project carried out by Formetis in 2012, for a professional client, a utility company, who maintain the infrastructure for electricity and gas services for their customers. The client requires an IT system for the production of complex contracts, while many quality requirements have to be met. The project has been carried out “by the book”, using the theory, methodologies and software tools provided by the discipline of enterprise engineering. This includes DEMO modeling, model validation and simulation, derivation of specifications for a supporting adaptive case management system and application of the Enterprise Operating System. The proposed approach is new and radically different from state of the art methodologies. This case is a guide to carry out similar projects for a wide range of applications in financial services, government services, legal, litigation etc.

Keywords: Enterprise engineering, DEMO, enterprise operating system, case management systems.

1 Introduction

This case study is a real world project, the “AVS case management system” for Endinet BV, The Netherlands, carried out by practitioners, Formetis BV in the Netherlands, applying the theories, methodologies and software tools provided by the discipline of enterprise engineering (EE) [5]. Formetis is also the developer of the DEMO engine and the enterprise operating system [section 3.6].

The purpose of this case study is 1) to provide other practitioners a new and radically different but appropriate and promising approach to carry out comparable projects in related domains [sections 4, 7.2]; 2) to assess empirically the quality of the approach and the result itself, compare functional qualities to state of the art methodologies [section 3]; 3) assessment of the alignment between the theoretical foundations of enterprise engineering and the here applied engineering approach [section 3.1]; 4) to identify areas for further scientific and engineering research [section 6].

The applied discipline of enterprise engineering includes the DEMO methodology, the software DEMO engine, the Enterprise Operating System, model simulation and validation, the derivation of high quality specifications for supporting IT systems.

This case study is not about the practice of DEMO modeling, nor the underlying theories since there is much empirical evidence for the appropriateness in many areas and the theory is well described [section 3.5]. The case study is about using the DEMO models as foundation for enterprise IT systems, how to do this, and the experiences so far.

This paper involves: 1) a description of the case [section 2] with an assessment of the specific problems and challenges that have to be met for business cases such as the one described here [section 2.1]; 3) an assessment of state of the art methodologies and best practices with their limitations [section 3]; 4) the task conducted [section 4] and the applied engineering approach [section 4.1]; 5) project results and experiences [section 5]; Future research [section 6] and 6) reflections [section 7].

This paper is partly based on scientific engineering, the discipline of enterprise engineering, but also based on many years of experience and domain knowledge of the industry. This paper represents our current personal stance and view on things, with client's interests on the foreground, and not always founded on published scientific research. The proposed approach is founded on a new paradigm and may be found controversial. It should be considered that enterprise engineering is still a young and immature engineering discipline and this is the first business case with this approach.

2 The Background of the Case

The Client of Formetis is a semi-public organization, Endinet BV ("client"), The Netherlands. They provide the infrastructure of electricity and gas services to their customers, which are private persons and various types of companies with a wide range of requirements. The delivery and periodical billing of these services is based on a contract signed by Client and the customers. This contract is a complex and custom made document – set of documents - that should meet various requirements, set out below.

As with most custom made services, the customer is an active co-producer of the contract. There are usually several subcontractors involved for the fulfillment of the contract. The contract covers issues such as type of services provided, costs, costs calculation methods, conditions for payments, instructions for the subcontractor, correspondence etc. Documents such as technical drawings, inspection reports, approvals originating from several sources are part of the contract or are related to this specific contract.

2.1 Special Requirements and Conditions

Special requirements and boundary conditions include:

1. Validation of the completeness and correctness of the draft contract, before sending it to the customer for his approval and signing.
2. The delivery of a contract proposition within a specified period of time after a customer's request, defined by law.
3. The contract should comply with external legal regulations and internal business policies, conditions, procedures and this compliance should be enforced, which is a GRC related topic [23], [section 3.4]. The quality of the contracts should be high; errors are unacceptable and may be expensive, incomplete transactions, deadlocks, deadline overflow, violation of business procedures etc, must be eliminated. The production of the contract is highly structured; there is a controlled sequence in the production of certain parts and there is (sometimes) an assessment and acceptance/approval step of production parts before the production may continue.
4. The possibility to modify the contract due to changing requirements from the customer during its life time, while maintaining the guaranteed coherence and correctness of the new version of the contract.
5. The possibility to modify the business process whenever needed, due to changing legal requirements or and improved way of working. This implies the need for evolving IT systems to support this agile enterprise.
6. The requirements that external parties, subcontractors participate in the execution of the contract and that the customer is also an active co-producer or subcontractor. The customer executes various roles in production, which should be modeled also.
7. The long life time of a contract; after signing it may exists for decades and may be subject to many modifications. This, and the fact that during operation the contract is the basis for production and billing, demand a high degree of continuity, maintainability and stability of the IT system over time.

2.2 The Required Solution

The required solution is a dedicated IT system (AVS case management system) to support the production of these contracts, meeting the requirements mentioned before.

3 Assessment of the State of the Art Technologies

The state of the art technologies and their strength - limitations are briefly discussed below, “as is”, as stated before, from our experience and professional point of view, an opinion, not always supported by published research results.

3.1 Case Management Systems

A case management system is defined by us being "*an IT system, supporting the production of tailor made services, produced by the human actors of an enterprise, in collaboration with the demanding customer, following certain rules and procedures*".

The services may encompass financial services, investments products, mortgage loans, litigation cases, medical services provided in hospitals, many (governmental) services for customers, such as this case. The professional application domain is very wide and may cover industries and domains currently captured by ERP[1] systems. The actual service is often some legally binding document combined with certain actions taken or "things" delivered. The service depends much on the production of documents. The services may be produced by chains of producers or sub-contractors. One of the sub-contractors is typically the customer.

Currently there is a lack of a precise and appropriate definitions. The AIIM (Association for Information and Image Management) [1] provides the following definition: "*A "case" is any project, transaction, service or response that is "opened" and "closed" over a period of time to achieve resolution of a problem, claim, request, proposal, development or other complex activity. It is likely to involve multiple persons inside and outside of the organization, with varying relationships to each other, as well as multiple documents and messages"*.
The AIIM and other parties define and describe the ways how the case management system is implemented, a "project", "transaction", "CRM systems", or "help desk", or "involve multiple persons", or "opened and closed over a period of time". While this is often true, this "how to" is considered not so relevant here and should not be part of a definition of what a case management system is.

There is an OMG standard for case management systems in development[2]. It is observed by us – our professional stance - that some of the limitations of BPMN we identified [section 3.3] (lack of formal foundations, lack of ontological qualities, concept overload, construct excess etc) also apply here. CMMN is therefor considered not (yet) good enough for our purpose.

The key notions of our case management system, presented above, definition are: 1) the production of a specific service; 2) the production is subjected to specific (business) rules from various origins [section 3.4]; 3) the service is tailor-made for a demanding specific customer; 4) the customer can be an active co-producer of the service. Our definition makes no statements on implementation choices. This definition is compliant with, and based on the definition of an "enterprise" provided by the discipline of enterprise engineering, Dietz J.L.G et al. [5]; an enterprise is a social system, composed of human actors, each actor is producing some part of the

[1] Enterprise Resource Planning systems are business software systems that store and manage data from many production stages, inventory, manufacturing, purchasing, sales, etc. ERP systems provide a set of perspectives on production and operation.

[2] The OMG Group, http://www.omg.org/spec/CMMN/ Case Management Model and Notation, CMMN. Version 1.0, 2014-05-05.

production, actors communicate about the production of some service for an external customer. This notion is covered in detail in the theory of Enterprise Ontology by Dietz [4].

From the above follows that a case management system is composed of two subsystems; one subsystem for implementation of business rules and procedures using some workflow-like[3] system, and one subsystem for management of all types of produced documents. Our approach for a case management system is composed of a structured dossier system [section 3.2] and the enterprise operating system [3.6].

3.2 Document and Dossier Information Systems

Most of today's many case management systems are essentially document-based, supporting the production of documents, and often supported by a so-called work flow system to control sequences of operations on a specific document. If the service to be provided is simple, not demanding much customer interaction, not handling many exceptions, then a document-based approach may be very suitable. Example: processing of credit card payments. If the service is complex, with interactions and contributions from several sources and sub-contractors, such as for example a mortgage loan, then a document-based approach may be found inadequate. The *context* of all related information carriers from many sources then becomes important to support the case worker and stakeholders well.

The concept of an electronic dossier is proposed here. It does: 1) provision of the case worker with all required information within the context of the case, in a well supporting way; 2) guarantee compliance to all applicable rules and procedures of the dossier using the first subsystem; 3) take autonomously appropriate actions when needed; 4) supports the production of documents.
An electronic dossier, is a highly contextual structured aggregation of documents, organized in such a way that all needed functional useful perspectives on the case are provides. The derivation of the structure of the documents is based on the “fine-grained” DEMO models, described in section 4.1, step 2. The electronic dossier itself is a model-driven dynamic structure and supports incremental linking and embedding of new documents at run time. The built-in knowledge of the electronic dossier about business procedures and rules to be followed is described in section 3.6.

3.3 Work flow Systems

A work flow procedure is typically defined being an orchestrated and repeatable pattern of business activities, based on the systematic organization of resources[4]. In practice it involves a sequence of operations, declared as work of a person or group. State of the art technology is BPMN[5], which provides a graphical representation of a business process.

[3] The term workflow-like idicates a similar purpose but based on different principles.

[4] Several definitions and 'types' of workflow exist.

While BPMN is the current de facto world standard[6], there are some serious limitations that may become problematic when dealing with these types of complex services. In Van Nuffel et al [17] an assessment has been made of BPMN 1.0[7] and recommendations have been made to enhance the formal foundations of BPMN 1.0. The most important findings were: 1) the lack of ontological completeness of BPMN, typically only the "happy flow" is/can be modeled, which means that any exceptions in the business process are not captured and at run time a manual intervention must be made; 2) The lack of representation of state and history in dynamic systems; 3) For non-trivial business processes the complete work flow procedure cannot be modeled by humans, simply because the complexity explodes exponentially; 4) The applied symbols and concepts are not aligned with a formal domain ontology and not based on an appropriate scientific theory. A large degree of concept overload, concept mismatch and construct excess has been observed[8]; 4) There is lack of appropriate abstractions in BPMN models, the models encompass "too much" and tend to become too large for human understanding; 5) A lack of formal rigor; BPMN is a notation, not a formal language. A formal language would enable the development of an software automaton that executes models expressed in that language in a proper way. 6) As a result, early simulation and validation of models, before any commitment of resources to implementation and coding, is difficult or even impossible. This limitation is considered very serious and a source of business-IT alignment problems[9] much later [10]. To address the business-IT challenges early simulation and validation, before any commitment to expensive coding implementation, is mandatory.

Our professional stance: For the production of complex services, BPMN is considered to be much better than other approaches in the industry, but not good enough for state of the art applications such as this case [section 2.1]. We calculated that an ontologically complete BPMN model of the 40 transactions [fig 4] of this case would require the "manual" modeling of 2080 state transitions [shown in section 3.3], which is considered 'impossible' to do and to verify for humans. Calculation of work flow should be done by a software engine using higher abstracted models.

3.4 Governance, Risk and Compliance (GRC)

The mentioned rules are partly in a domain encompassing three sources; GRC, the abbreviation of Governance, Risk and Compliance. The compliance of the enterprise in operation to externally imposed legal rules and regulation, is generally known as Compliance. The compliance of an enterprise to internally defined principles and rules is generally know as Governance. The identification and mitigation of risks,

[5] The OMG Group, Object Management Group, Business Process Model and Notation. http:www.bpmn.org

[6] An important foundation of the state of the art is described in 'workflow management', W. van der Aalst, K van Hee, 2002. However, we do not follow this approach or the BPMN approach for modeling, methods and systems.

[7] The current latest version of BPMN is 2.0. Some comments may not apply for version 2.0.

[8] These notions are described by the Bunge Wand Weber ontology.

[9] In [10] is shown that if any IT systems are implemented when the requirements and specs are not yet of high quality, then many expensive modifications on software can be expected.

from many sources and of different nature, in operation is generally known as Risk. The GRC domain is also immature and not well-defined as described by Racz et al [20]. Racz observes 1) a lack of appropriate definitions; 2) a lack of scientific foundations and scientific research; 3) a lack of shared understanding of concepts between professionals and their customers; and 4) the fact that most literature is "sales-driven", provided by software companies and professionals. Racz provides, based on a survey, a set of appropriate definitions and a framework. We follow the Racz framework since we consider this of great relevance, especially in financial services. Verwaest [24] provides an assessment and an approach for COBIT5, but this approach is also applicable to other domains, especially in finance. A systematic assessment of GRC issues using DEMO model simulation is discussed in section 6.2.

3.5 The Discipline of Enterprise Engineering and the DEMO Methodology

The engineering approach of this case study is founded on the (emerging) discipline of enterprise engineering ('EE'), defined in [5]. EE is well-founded on empirical sciences, formal methods and design science, as opposed to state of the art ICT techniques that are 'best practice' and generally lack scientific foundations, as described in [6] for TOGAF and in [17] for BPMN. EE is "just as solid" as other engineering disciplines, for example electronic chip design, automotive engineering, aviation etc.

The methodology to devise models of enterprises is DEMO, in detail described in [4]. The functional appropriateness of DEMO models for 'shared reasoning' between stakeholders is supported by much empirical evidence and many business cases, such as provided by Mulder [18], and [16] for requirements analysis, [24] for COBIT, and many more at [2] and [3]. The advantage of DEMO models, and the relevance for this case study, is that these models are formal propositions in a formal language, of a high C4-ness quality: comprehensive[10], coherent[11], concise[12] and consistent[13]. These ontological qualities enable the construction of a software engine, the DEMO engine, that executes DEMO models [10, 14] "as source code". The DEMO engine, executing a model, is in fact an IT system that controls the operation of an enterprise and enforces compliance to a model [7, 8, 9].

A very short overview of DEMO modeling and its symbols is provided here. DEMO is founded on four axioms. The operation axiom of DEMO states that there are actors, fulfilling actor roles, and transactions. Transactions are composed of communication between the actors about the production to be delivered, and a reference to the production it self.

The DEMO transaction axiom states that any communication between actors follows a specific well-defined pattern.

[10] Comprehensive: all symbols for concepts that must be in a model, are there.

[11] Coherent: semantic meaningfulness of symbols and their relations from every perspective.

[12] Concise: there are no symbols of concepts in a model that are not necessary.

[13] Consistent: the lack of any anomalies and contradictions.

The informal example on the right of fig. 1 shows the delivery of flowers, where 1) the customer (initiator) request flowers, 2) the producer (executor) promises the delivery of flowers, 3) the producer states that the flowers are delivered and 4) the customer accepts the flowers. This is the basic communication pattern. More sophisticated patterns exist where an executor may decline a request ("we have no flowers, I cannot deliver") and an initiator may refuse the flowers ("these flowers are not what I want, I do not accept").

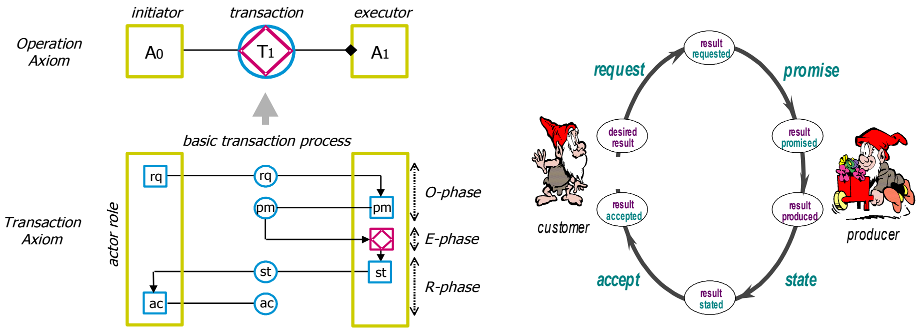

Fig. 1 DEMO actors, communication and production.

The left part of fig. 1 is a formal representation that shows the symbols of the operation axiom for this transaction; the actor that fulfills the initiator role, the actor that fulfills the executor role and the transaction T1 with the communication between the actors and the production about which the actors communicate.
These actor roles and transactions are the atomic building blocks of any complex organization, as shown in figures 3 and 4, and the detail shown in fig 5 as example. The model on the top left side (Initiator A0, Transaction T1, Executor A1) is drawn using a graphical tool and is directly executed by the DEMO engine.

3.6 The Enterprise Operating System (EOS)

The foundation of our approach is the Enterprise Operating System (EOS), [10,14] composed of the DEMO Engine that executes a specific DEMO model of an enterprise. The main capabilities of the DEMO engine shown in fig. 2 are:

1. Edit Model: the construction of guaranteed verified DEMO models.
2. Models can be stored in DMOL XML representation in a file repository.
3. Models can be rebuild from DMOL XML files, for further simulation or production.
4. Model simulation and model validation, supporting shared reasoning by stakeholders for optimal business-IT alignment, assessment of governance [7], risk and compliance issues [24], and the derivation of specifications.
5. Precise formal compliance with enterprise ontology, the PSI theory [5].

6. At run time (*Communication and Control of Actors 1..n*) complete *prescriptive* control of the enterprise, with enforced compliance to a DEMO model, which is a work flow capability, calculated from the DEMO model instance [8,9]. Actors cannot act 'outside' the state space and state transition space of a DEMO model.
7. *Descriptive* control of the enterprise, there is total knowledge of every atomic act and fact [8,9] from any actor in the enterprise.

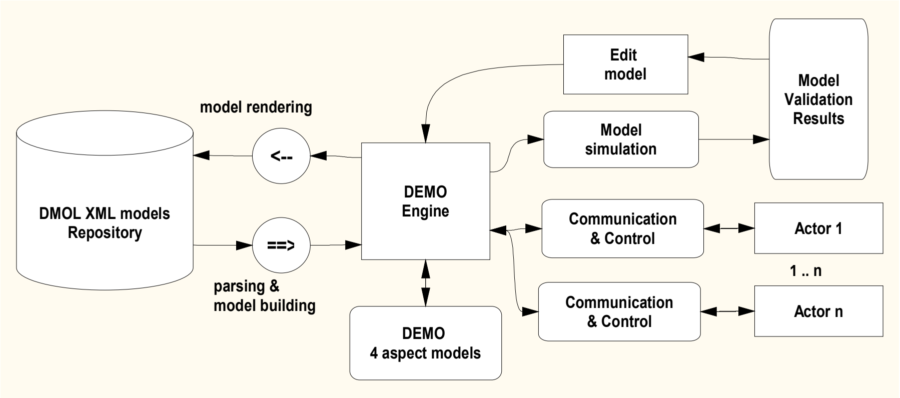

Fig. 2 Overview of the DEMO engine functions and capabilities.

The DEMO engine, with its descriptive and prescriptive control capabilities of an enterprise to a model in execution, is very similar to a computer operating system. It provides the interface and abstractions between the hardware electronics and any software applications, hence the term enterprise operating system (EOS).

The calculation at run time of workflow-like[14] capabilities from much simpler higher level DEMO conceptual models that overcome the before-mentioned BPMN limitations is a major asset. It reduces the effort in coding and BPMN type of modeling for the process aspect completely. These DEMO models, expressed in the DMOL[15] formal language, are executed by the software DEMO engine [10,14,15]. In addition, DEMO models executed on the DEMO engine – together they constitute the enterprise operating system - are a 'strong' prescriptive system that guarantees compliance of the enterprise to a specific model.

To summarize; the case management system is composed of two subsystems; 1) The subsystem enterprise operating system, composed of the DEMO engine executing DEMO models. 2) The subsystem for the management and production of documents.

[14] The workflow capabilities mentioned here are based on the Language Action Perspective, and communicative acts, provided by Habermas [11] and others. There exist other workflow approaches.

[15] DMOL (DEMO Modeling Language) is an XML-based representation of the four DEMO aspect models (Formetis).

4 The Task Conducted

The AVS system is implemented as described, meeting the requirements of section 2.2. The planned deadline for the first version was slightly exceeded due to the fact that we built the DEMO engine simultaneous with this project, a calculated risk that paid off well later. The project was accepted and carried out for the most part on a fixed price contract, except for those parts that could not be specified or calculated well in advance. Later additions were all carried within the deadlines and planned budgets. The current system is continuously being enhanced and extended, capturing more functionality in the organization, due to its ease and flexibility to modify and enhance it. The management of the Client recently concluded that all legal requirements were met – operational compliance with regulations, which is a GRC topic [section 3.4]. Simultaneous the DEMO engine has been developed, which is application-independent and can be used for any DEMO model.

4.1 The Applied Engineering Approach

The following steps were performed for the AVS case management system.

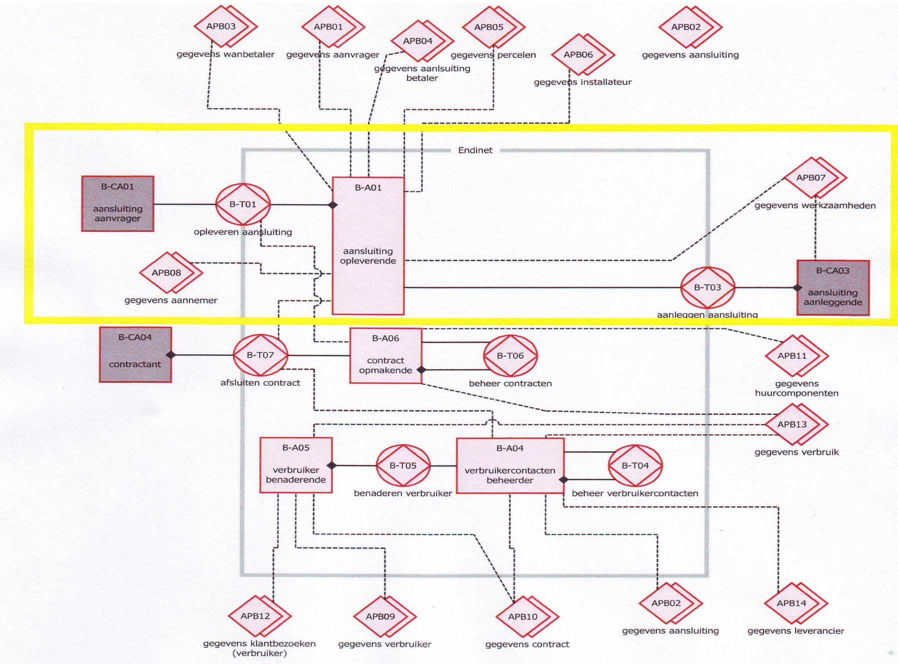

fig 3. The ontological DEMO model of a part of the organization (example).
(The yellow line is irrelevant here)

1. The project started with DEMO modeling with the knowledgeable staff, which delivered the so-called ontological model of the enterprise of fig. 3.

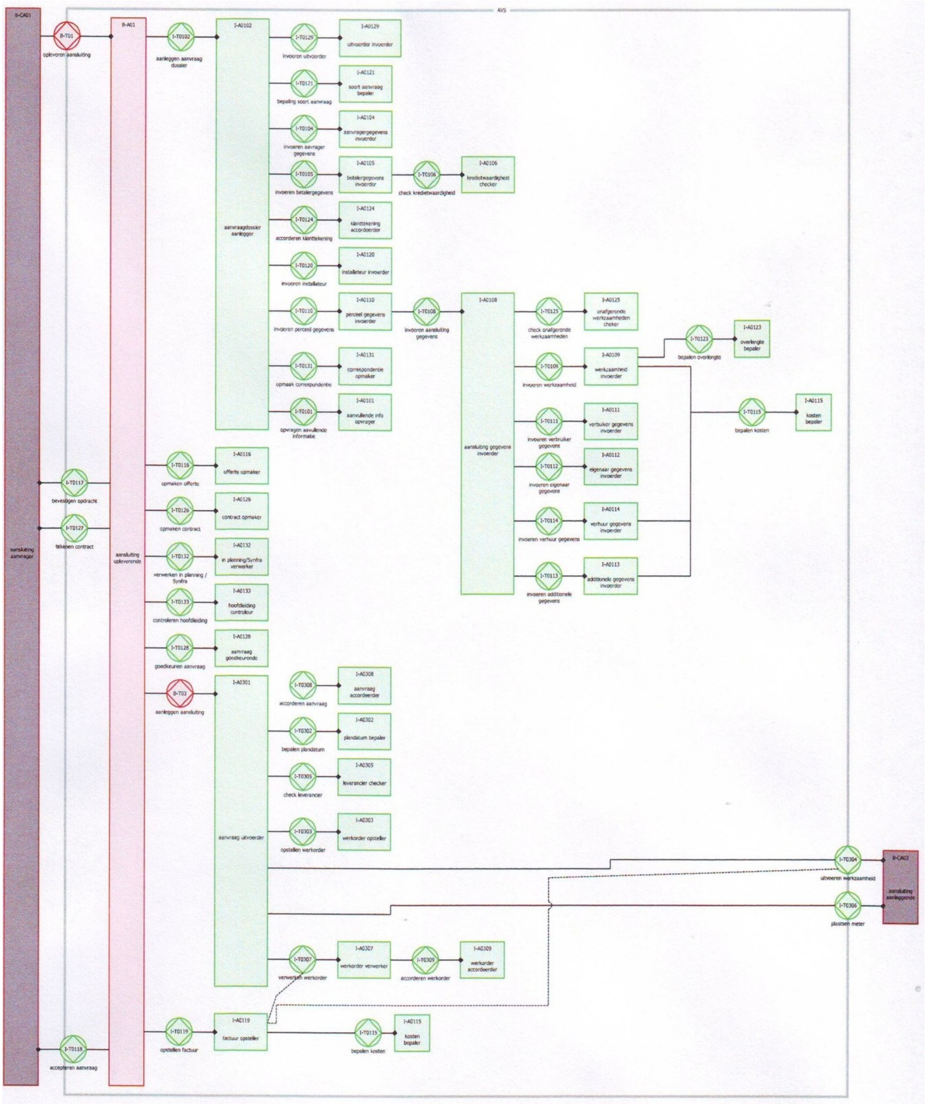

Fig 4. Detailed realization example – the NCD_AVS model is an implementation of a DEMO model, including infological and datalogical transactions.

Extensive shared reasoning with the staff has been done to validate that this DEMO model (fig. 4) represents properly the service of the enterprise delivered to the customer. This ontological model is strictly implementation-independent.

2. The second step involved a “realization” DEMO modeling step (designing a specific implementation, one of many), based on the implementation-independent ontological DEMO model. De Jong provided a detailed methodology to carry this out in a systematic way [13]. The ontological DEMO model (only “red” ontological transactions) is extended with so-called “green” infological and “blue” datalogical transactions. The result is a DEMO model with 40 ontological, 'infological' and 'datalogical' transactions[16] [fig 4]. This model supports also high quality specifications for the dossier production system; each production fact examined provides these specs. The model made also clear which further design specifications were needed, for example GUI specifications and specification of interfaces with existing external legacy type IT systems. At this stage the enterprise operating system, providing all work flow capabilities is ready for simulation and for full production, without any programming so far. A part of this model is shown in fig 5 as an example.

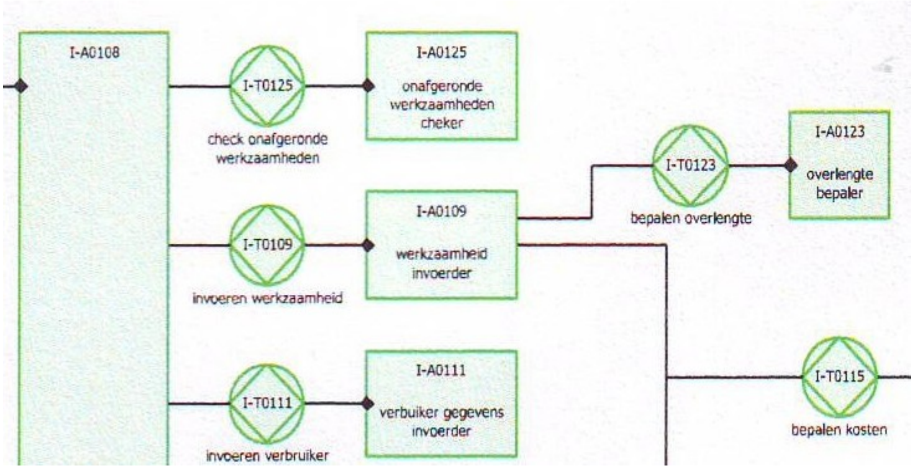

Fig 5. A part of the model of fig. 4, as example.

This part (Fig. 5) has the following meaning:
Actor I-A0108 is initiator of these transactions:
- Transaction I-T0125 (“check for unfinished work”), to be done – executor - by actor role “actor who checks any unfinished work”.
- Transaction I-T0109 (“fill out data of the type of work to be done”), to be done by role I-A0109, actor who fills out this data.
- Actor I-A0109 is also initiator for the work described by transaction I-T0123 (“determination of excess length”), executed by actor “who decides on excess length”.

3. The third step involved the implementation in executable software in a systematic way of the dossier system. The DEMO models provides specs for decomposition of

[16] DEMO methodology identifies three types of human activities, ontological *performa*, infological *informa* and datalogical [5].

this subsystem into a large number of elementary document production components. Each of these components can be implemented typically, depending on complexity, within a few hours or a few days at most using modern IT system tools.

4. The fourth step is acceptance testing with the stakeholders, followed by production after approval. Stakeholders found that the actual implementation was precisely compliant with the models they had made. Extensive acceptance testing led to some further improvements of the underlying model. There have never been any errors in the model execution of the DEMO engine. The number of software bugs for the dossier system was low and easy to fix.

5. The final step is the ongoing support and modifications of the system over its long production time. These modifications may come from various sources, new products, new regulations, better insight in the operation. Essentially a modification of the model is enough to enforce a new way of working. Any modification to the document production system is as simple as possible due to the well-defined isolated components that must be modified.

5 The Results of the Project

The practical experiences until now are:

1. The first two steps [section 4] exploit the knowledge of key personnel and the early model validation assures an optimal business-IT alignment. The application has in essence been designed by the key staff, which almost assures acceptance for full production. The staff was strongly motivated to do this.

2. The complex document production and management software has been decomposed into 40 simple production system software components [fig 4]. Each software component is highly structured, well specified and very simple to implement in software (also model-instance-driven). Its simplicity enables easy modifications.

3. The resources in time required for the implementation in software are already less than the resources in time for the modeling stages. A further reduction in resources for software implementation is expected, a clear move from programming to modeling.

4. After acceptance testing and some time running in production, stakeholders proposed a number of implementation changes of the system and the DEMO model. These modifications were simple to implement without much impact on other software modules. The staff recognized also that the software implementation we delivered complies precisely with the models they designed and approved themselves. Their general perception is that the system is very demanding and structured in the separation of various production steps and enforces a specific way of working with little freedom. They find also that the transaction steps deviate from 'standard' GUI practice and demands many “extra mouse clicks”, which is true and will be fixed. They appreciate that ongoing modifications due to better insight are quickly implemented.

5. The communication between the actors for each production has been simplified because in a good case management system one single person – the responsible case worker - does most of the contract production work. However, the precise sequence of the production steps is well guarded and there are no anomalies or incomplete transactions. Correctly executed nested transaction rollback is assured.

6. There is no separate 'work flow' modeling stage (BPMN etc). The communication patterns between actors and the sequence of process steps are completely calculated from the DEMO model instance under execution. A modification of the business process demands only a new DEMO model to be installed.

8. The relation between project costs and required resources versus project size, measured in number of transactions, seems to become closer to linear. This is opposed to the typical exponential ratio found so far in larger software engineering projects.

9. An important advantage is that the only remaining 'creative' phase is the modeling stage. The implementation in software leaves much less space for 'free and creative' programming choices. Software engineering for adaptive case management systems is becoming more a standard production process, as in other engineering domains and as it should be.

10. This is a project where we simultaneously build an IT system and the underlying executing software engine, a calculated risk. We missed the first deadline by a number of weeks. Some software shortcuts have been made to achieve operational functionality within the limited available time. Currently most of these have been replaced by proper software implementations, but much more work remains to be done.

The customer is satisfied, since it functions very well, adaptations are easy to implement and this is the only system in the Netherlands of this type that meets the legal compliance requirements.

6 Future Research

Two specific topics for future empirical research are mentioned below.

6.1 Systematic Derivation of High Quality Project Specifications

The systematic derivation of high quality project specifications while using the DEMO engine to simulate DEMO models must be well supported. Simulation of DEMO models at design time identifies all specific transaction states that may occur in real life operation. For each process state a set of requirements must be devised. This includes the availability of all required information for an actor to take a justified decision for a communication act, specified in the DEMO external data banks. This must be carried out in a much more structured and systematic way than in this project.

6.2 Assessment and Mitigation of Governance, Risk and Compliance (GRC)

The topic of GRC is included in this paper since there are some compliance issues, but also because this approach seems to support this important topic well [24]. Similarly to the topic before, detailed model simulation supports the identification of all possible transaction states for the whole enterprise, which must be assessed also from a risk (GRC) perspective. What are the specific risks[17] from outside or inside related to any specific state? Does reaching an "unusual state" imply a specific risk, an attack from outside or inside? Is the actor provided with all information needed to carry out a communication act, but nothing more than that (need to know information limitation)? Is any actor fulfilling more actor roles and does this represent a risk if the actor is malicious? Is there any combination of two actors, assuming that they might be malicious, that are able to commit fraud or cause damage? Which actions can be designed at design time to mitigate potential hazards at run time? Process mining type of analysis in operation is considered very valuable. This type of analysis supports the validation that a specific DEMO model based operation complies with Governance principles and Compliance requirements (or not). Detailed analysis of the interaction between the enterprise of external customers is of interest to Compliance analysis.
For financial services industry, the compliance to regulation coming from sources such as Basel, Sarbanes Oxley, is of such paramount importance, which merits and requires extensive research.

7 Reflections

We consider the decision taken to develop the DEMO engine simultaneous with the project execution as a well-calculated risk that paid off favorably, despite exceeding the first project phase deadline for acceptance testing. Without the DEMO engine it would be impossible to design and implement such a case management system that operates so strict and precisely model driven. We might have stayed within the deadline for acceptance testing, but would have had much more rework and fixes immediately after acceptance testing due to the incomplete and lower quality project specs. The implementation of this business process of fig. 4 in BPMN is considered to be 'impossible' for humans.

DEMO modeling is considered 'difficult' for many stakeholders to capture since it is a complete abstracted representation; there is not a trace of implementation or parts of an example. Stakeholders need time and training to grasp the meaning of a DEMO model before they are able to reason about a model. BPMN models are much easier to understand because "much more"is shown. However, the high level of abstraction with formal rigor is the strongest point in complexity reduction of DEMO.

[17] We consider here exclusively risk related to business procedures, human actions and communication, truthfulness, sincerity, responsibility, competence, assigned to human actors. We do not state anything about financial and similar risks, which is outside our scope.

Essentially, the followed approach is radically different from state of the art methods and best practices in four ways.

1. The first difference is the solid and systematic “engineering approach”, provided by the discipline of enterprise engineering with its scientific, formal foundations and engineering methodology.
2. The second deviation from the state of the art is the top-down approach, we start with modeling the ontological enterprise. This is based on the recognition that an enterprise is essentially a social system, composed of human actors who collaborate (production) and communicate together to produce some (valuable) service of production for an external 'customer'.
3. The Greek Philosopher Protagoras[18] stated “Man is the measure of all things”, which is considered to be applicable here too. Key notions are human qualities such as responsibility (a relation from an actor to all other actors), authority (a relation from all other actors to a specific actor) and competence (the assumption that a specific actor has all the required competences needed). In addition, actors are assumed to act and communicate in a truthful and sincere way with each other. We start with the human qualities. Human stakeholders and actors design their optimal “way of working”, and supporting IT systems, using DEMO. They regain freedom over their work. This is totally opposed to for example state of the art ERP systems that squeeze humans into their rigid application structure.
4. We abandon in this engineering approach the state of the art approach of starting modeling business processes, in a bottom-up way using Petrinet modeling. We also abandon the OMG approaches for BPMN and CMMN.

Three major benefits and advantages:
The first and far most important benefit is considered to be the human oriented approach where individual humans are provided with a clear description of their tasks, their responsibilities and competences. In addition the humans provide the essential domain knowledge to design their enterprise. Our human-oriented approach makes Weber-type[19] of machine bureaucracies and fat layers of managers obsolete.
The second benefit is the fact that state of the art problems in IT system engineering are addressed [14] with this approach. These include addressing 1) the business-IT alignment challenge; 2) the unmanageable and uncontrollable costs in resources and the high failure rate of IT projects; 3) the lack of support for the agile enterprise and evolving information systems.

[18] The Greek Philosopher Protagoras stated that man is the measure of all things, which implies that there exists exclusively a subjective truth for each human individual, therefor excluding any objective world. It also implies that human values are the only tru values that exist. While the human oriented approach is considered of great importance, we do also assume the existence of an objective representation of reality, the foundation of engineering, provided by the theory of enterprise ontology [Dietz, J.L.G., 2006].

[19] Max Weber, German sociologist, claimed that a bureaucracy is the most efficient and effective way to organize human activities. Humans are reduced to machines working in a strictly coordinated way designed by authorities. Weber recognized however the dehumanizing threat to individual freedom.

The third benefit is the strong overall reduction in complexity that can be achieved.

In its current implementation the DEMO engine is not yet fully mature, some programming shortcuts have been taken, not all four DEMO aspect models have been implemented completely, and the user interface must be improved.
It should be noted that this business case is the first professional IT system case of this kind, that this approach needs much refinement and that many more business case studies are needed. The theoretical foundations of enterprise engineering are very well defined but the applied methodologies, except for DEMO itself, are still young and immature.

7.2 Comparable IT Systems that may benefit from the proposed Approach

The application domain of DEMO enterprise operating system, combined with a case management system, may challenge to a substantial degree state of the art ERP systems and be the core production environment for financial services. We consider IT projects comparable and potentially suitable for this approach if most of the following conditions apply:

1. There is a demanding customer who demands a unique tailor-made service (or product). Within a generic type definition for the service, the provided service is a unique instance, which goes further than a selection of values for a set of parameters, such as the color and features of a car. An example is the legal requirement for the service producer to assess the personal situation of a customer, "does this service, as requested and selected by the customer, beneficial or involve potential risks and disadvantages for him?". Part of the service is the assessment of risks, value for the customer.
2. The customer is active co-producer of the service. This extends much further than the provision of customer data. There is a dialog between customer and producer to configure the service in the best way, investigate alternatives, and eventually decide to recommend the customer not to take the service.
3. There is a set of legal requirements that have to be met. In addition, convincing evidence that the legal requirements have been met in operation for each production instance must be supplied. The same applies to (similar) governance principles that apply to the operation of the enterprise.
4. The production is business process driven, where the processes are not trivial or execute exclusively a simple happy flow of a fixed sequence of steps. When sub-contractors or the customer are involved any process is already complex and unmanageable using BPMN -type technologies.
5. The production is event driven, where events originate from many sources, sometimes in random sequence. There are many different process execution paths possible.
6. The production is document-based, optionally accompanied by specific actions, such as for example a medical treatment in a hospital.

Comparable projects can be found in financial services, legal, litigation, government services, medical treatments etc.

8 Acknowledgments

Formetis expresses it gratitude to Eduard Babkin, National Research University, Higher School of Economics, Nizhny Novgorod, Russia; to Sergio Guerreiro, Lusofono University, Lisbon, Portugal; and Jan Verwaest, Verwaest.biz, Westerlo, Belgium, with whom we cooperated very well and gained better understanding of this approach.

9 References

1. AIIM (tm) (Association for Information and Image Management) - http://www.aiim.org/What-is-Case-Management.
2. Ciao! Consortium; Cooperation & Interoperability - Architecture & Ontology, online: www.ciaonetwork.org
3. DEMO Knowledge Centre, Design and Engineering Methodology for Organizations. online: http://www.demo.nl. (2012)
4. Dietz J.L.G.; Enterprise Ontology: Theory and Methodology. Springer Berlin Heidelberg New York (2006).
5. Dietz, J.L.G; Hoogervorst, J.A.P; The discipline of enterprise engineering. Int. J. Organisational Design and Engineering, Vol. 3, No. 1, 2013.
6. Dietz, J.L.G; Hoogervorst, J.A.P; A critical investigation of TOGAF, Lecture Notes on Business Information Processing, LNBIP, no 79, Springer Verlag, 2011.
7. Guerreiro, S.: Enterprise governance enforcement in the operation of the runtime transactions using DEMO and ACM. Enterprise Engineering Working Conference. CIAO! DC, Antwerp. (2011)
8. Guerreiro, S., Vasconcelos, A., Tribolet, J., Kervel, S. J.H. van: Executing Enterprise Dynamic Systems Control with the DEMO Processor: the Business Transactions Transition Space Validation. Proceedings of the 7-th Mediterranean Conference on Information Systems 2012. Springer LNBIP series Lecture Notes in Business Information Processing. (2012)
9. Guerreiro, S., Vasconcelos, A., Tribolet, J.: Adaptive Access Control Modes enforcement in Organizations. Proceedings CENTERIS 2010, Springer-Verlag Berlin Heidelberg, Part II, CCIS 110, pp. 283–294, (2010)
10. Guerreiro, S.L.G.; Kervel, Steven J. H. van; Babkin E.A; Towards Devising an Architectural Framework for Enterprise Operating Systems. Proceedings of ICsoft 2013 – 8th International Conference on Software Paradigm Trends. SciTePress. (2013)
11. Habermas, J.: Theorie des Kommunikativen Handelns, Erster Band, Suhrkamp Verlag, Frankfurt am Main, (1981)
12. Hevner, A.: A three cycle view of design science research. Information systems and Decision Sciences, Scandinavian Journal of Information Systems, 2007, 19(2): 87-92. University of South Floria, USA. (2007)
13. Jong, J. de; A Method for Enterprise Ontology based Design of Enterprise Information Systems. PhD thesis, Delft University of Technology; ISBN: 978-90-5335-758-3. 2013.
14. Kervel, S. J.H. van, Dietz, J.L.G., Hintzen, J., Meeuwen, T. van, Zijlstra, B.: Enterprise Ontology driven Software Engineering. Proceedings of ICsoft 2012 – 7th International Conference on Software Paradigm Trends. SciTePress. (2012)
15. Kervel, S. J. H. van: High quality technical documentation for large industrial plants using an enterprise engineering and conceptual modeling based solution. 30th International Conference on Conceptual Modeling ER 2011.

16. Krouwel, Marien and Op 'T Land, Martin, “Using enterprise ontology as a basis for requirements for Cross-organizationally Usable applications” (2012). MCIS 2012 Proceedings. Paper 23.
17. Nuffel, van D., Mulder, H., Kervel, S. van.: Enhancing the formal foundations of BPMN using Enterprise Ontology. In: CAiSE CIAO. (2009)
18. Mulder, J.B.F.: Rapid Enterprise Design. PhD. Thesis, Delft University of Technology. (2006)
19. Nicolas Racz, Edgar Weippl, Andreas Seufert; A Frame of Reference for Research of Integrated Governance, Risk & Compliance (GRC), 2010. (http://www.grc-resource.com/)
20. Nicolas Racz, Edgar Weippl, Andreas Seufert; Governance, Risk & Compliance (GRC) Software – An Exploratory Study of Software Vendor and Market Research Perspectives
21. Sauer, C., Cuthbertson, C.: The state of IT project management in the UK. Templeton College, Oxford University. (2003)
22. Standish Group,: CHAOS Summary 2009. Online: http://www1.standishgroup.com/newsroom/chaos_2009.php (2009)
23. Vicente, P.F.O, A Reference Architecture for Integrated Governance,Risk and Compliance, CAiSE 2011.
24. Verwaest, J. “Towards a comprehensive methodology for the implementation of Enterprise Governance & Assurance of IT, based on COBIT5, using the discipline of enterprise engineering”. 2013. MSc. thesis, Antwerp Management School.