Exploring Organizational Implementation Fundamentals

Martin Op 't Land[1,2,3] and Marien Krouwel[1,4]

[1] Capgemini Netherlands, P.O. Box 2575, 3500 GN Utrecht, The Netherlands
{Martin.OptLand,Marien.Krouwel}@capgemini.com
[2] Antwerp Management School, Sint-Jacobsmarkt 9-13, 2000 Antwerp, Belgium
[3] Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands
[4] University of Antwerp, Prinsstraat 13, 2000 Antwerp, Belgium

Abstract. To survive and even thrive on environmental and internal change, organizations have to be agile. Though change occurs in organizational essence, such as products and services delivered, most of the time change deals with different organizational implementations, such as sourcing, order of working and distribution of tasks. To informedly decide upon such organizational implementations, a systematic overview of organization implementation variables is required, which is currently not available. We drafted a list of organization implementation variables from literature, and tested it against two different organization implementation descriptions of OMG's EU-Rent case, using the DEMO model for this fictitious car rental company as its implementation independent essence. We found a list of 20 of such variables from literature, which was extended in the two tests by another 10 variables. Using these variables in Enterprise Engineering enables traceability in governing enterprise transformations; moreover, we expect many of them to have the potential to be generically supported by IT, thus enabling agile IT.

Keywords: DEMO, Agile Enterprise Engineering, Enterprise Ontology.

1 Introduction

As strategic and operating conditions become increasingly turbulent due to factors such as hyper-competition, increasing demands from customers, regulatory changes, and technological advancements, the ability to change becomes an important determinant of firm success [1]. This ability is generally referred to as *agility*, e.g., as summarized by Oosterhout [2]: "Business agility is the ability of an organization to swiftly change businesses and business processes beyond the normal level of flexibility to effectively manage highly uncertain and unexpected but potentially consequential internal and external events, based on the capabilities to sense, respond and learn."

Though change – as a consequence of external and internal events – occurs in organizational essence, such as products and services delivered, most of the time change deals with different implementations [3]. Typical organizational implementation choices include sourcing, order of working and distribution of tasks.

H.A. Proper, D. Aveiro, and K. Gaaloul (Eds.): EEWC 2013, LNBIP 146, pp. 28–42, 2013.

To informedly decide upon such organizational implementations, a systematic overview of organization implementation variables is required, which is currently not available. Research in the agility domain appears to focus on black-box variables – such as the type of events triggering change [4] or the measurement of agility [5] – or on the transformation processes needed to bring about the change [6]. Research in the agility domain about white-box variables is until now restricted to the IT domain; e.g., Normalized Systems theory [7] proposes a "Set of Anticipated Changes" for IT systems in terms of the (detailed) function of an IT system, such as an additional data field or an additional trigger element.

We drafted a list of organization implementation variables from literature, and tested it against two different organization implementation descriptions of OMG's EU-Rent case [8]. For this fictitious car rental company, we used its DEMO model [9] as its implementation independent essence, since it remains the same as long as the products and services of an enterprise stay the same.

We found a list of 20 of such variables from literature, which was extended in the two tests by another 10 variables. Examples of these variables include (a) the choice to combine or split actor roles in the work of an employee, (b) to apply delegation and separation of functions, and (c) to apply a fixed order of working in a process or allow steering of that order by the individual employee.

Explicitly using these variables in Enterprise Engineering enables traceability in governing enterprise transformations. Also we expect many of them to have the potential to be generically supported by IT, thus enabling agile IT. For instance, when the choice how to combine or split actor roles in the work of an employee (ad *a*) can be registered explicitly and in one place, all connected software applications can use this information to change their – e.g., GUI and security – behavior accordingly, potentially without the need for this software to be reprogrammed when this choice is changing. Since changing (also organizational) implementation variables tend to have combinatorial effects [7], the future potential for wider validation and application of this list is significant.

The remainder of this paper is structured as follows. Section 2 elaborates the problem statement: what do we understand by implementation and agility, and why is it relevant to find organization implementation variables. Section 3 describes the draft list of organization implementation variables from literature, which is then tested in Section 4, using the two different organizational implementations of the EU-Rent case. Finally, Section 5 provides the conclusions as well as directions for further research.

2 Problem Statement

In this section we will first introduce some definitions by explaining the Generic System Development Process (GSDP) as defined by Dietz (Fig. 1). After that, we will present our findings on existing literature about agility, events and implementation.

2.1 Enterprises and Generic System Development Process

We define *enterprise* as a goal-oriented cooperative. The organization of an enterprise is a heterogeneous system, constituted as the layered integration of three aspect systems, namely the Business (B) system, the Informational (I) system and the Documental (D) system [10, p115]. The production of these systems concern (B) original acts (material and immaterial), such as deciding, judging and creating, (I) informational acts, such as remembering, recalling and computing and (D) documental acts, such as storing, retrieving, transmitting and copying.

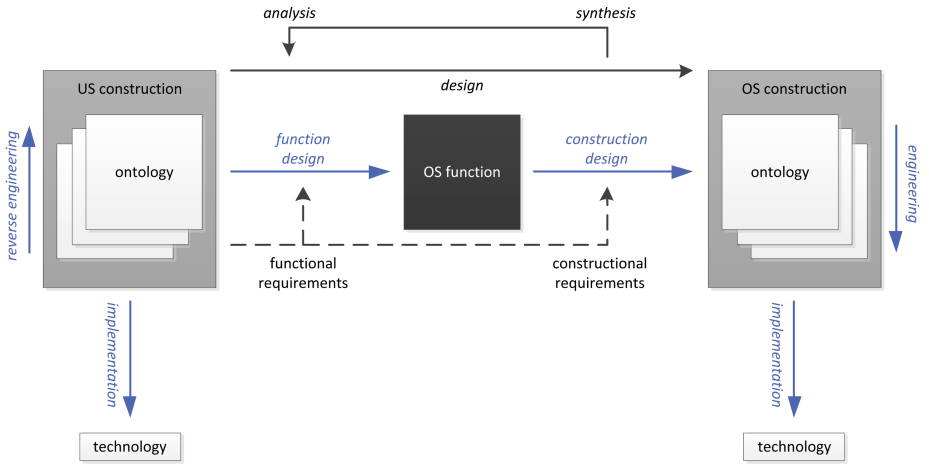


Fig. 1. Generic System Development Process [11]

Any development process concerns two systems involved, the *Object System (OS)*, and the *Using System (US)*. The OS is the system to be developed; the US is the system that will use the services (the functionality) offered by the OS once it is operational. The development of the OS consists of a *design*, an *engineering*, and an *implementation* phase. The design phase comprises a function design and construction design. *Function design*, the first phase in the design of the OS, starts from the construction of the US and ends with the function of the OS. Function design delivers the requirements of the OS, or a black-box model of the OS. This black-box model clarifies the behavior of the OS in terms of (functional) relationships between input and output of the OS. This function model of the OS does not contain any information about the construction of the OS. *Construction design*, the second phase in the design of the OS, starts with the specified function of the OS and ends with the construction of the OS. Construction design bridges the mental gap between function and construction, which means establishing a correspondence between systems of different categories: the category of the US (where the function of the OS is defined), and the category of the OS. Construction design delivers an ontology, the highest level white-box model of the OS. This white-box model clarifies the internal construction and operation of the system in terms of collaboration between its elements

to deliver products to its environment. By an *ontology* or ontological model of a system we understand a model of its construction that is completely independent of the way in which it is realized and implemented. The *engineering*[1] of a system is the process in which a number of white-box models are produced, such that every model is fully derivable from the previous one and the available specifications. Engineering starts from the ontological model, produces a set of subsequently more detailed white-box models and ends with the implementation model. By *implementation* is understood the assignment of technological means to the elements in the implementation model, so that the system can be put into operation. By *technology* we understand the technological means by which a system is implemented. A wide range of technological means is available, including human beings and organizational entities, ICT artifacts (e.g., phone, email, computer programs), and mechanical means. By *implementation variables* we mean the dimensions in which organizational implementation choices are made.

As an enterprise consists of three integrated layers, it can be developed by applying the GSDP three times [12]:

1. first the US is the (many times: commercial) environment in which the enterprise is going to be operational, and the OS is the B-organization of which the functional model contains the services that the enterprise will deliver to its customers [10, p77];
2. then the US is the B-organization, and the OS is the I-organization of which the functional model contains the information services (e.g. reason, compute, remember, reproduce) that the I-organization will deliver to the B-organization [10, p114];
3. finally the US is the I-organization, and the OS is the D-organization of which the functional model contains the documental services (e.g. store, retrieve, copy, destroy, transmit) that the D-organization will deliver to the I-organization [10, p114].

By applying GSDP for the enterprise as a whole (so three times), it is now possible to systematically categorize impact of change. Change in environment can be responded to by choices in function, which in turn will influence construction on both ontological and implementation level. Similarly, changes in the B-organization generally will influence the I- and D-organization, and the other way around. We illustrate this by some examples of changes in a law:

- a law stating that one organization cannot provide both banking and insurance services, affects the B-functional model;
- a law stating rules for granting a subsidy affects the business rules, i.e., the B-ontological model;
- a law stating reporting obligations affects the way of providing information and/or saving of data, i.e., the I-functional and D-functional model;
- a law stating the channels offered affects at least the implementation.

[1] Engineering is meant here in the narrow sense of the term, contrary to its general use in civic engineering, electrical engineering, mechanical engineering, etc.

Although one is obliged to adhere to legislation, law often leaves freedom of choice. For example, if law states one must at least provide a non-digital channel, one is still free to choose between telephone and physical service desk (or both). So, the change in law is an event in the environment, possibly but not necessarily responded to by an organization with a change in the function and/or construction of the organization.

2.2 Agility, Events and Implementation

To thrive in an environment of continuous and often unanticipated change, an enterprise needs to be agile [5]. Oosterhout [2] summarizes several definitions of agility as "the ability of an organization to swiftly change businesses and business processes beyond the normal level of flexibility to effectively manage highly uncertain and unexpected but potentially consequential internal and external events, based on the capabilities to sense, respond and learn." The question then arises what these kinds of events are.

Using the perspectives of the Enterprise Engineering Framework (EEF) [13], we categorized several event classifications found in literature (Table 1). As literature does not explicitly mention whether Technology and Resources deal with an *available* or a *chosen* implementation, we split EEF's original Context perspective in Context (environment & demand) and Context (supply), and positioned Technology/Resources (available) in the Context (supply) perspective. Likewise, we made a distinction between Customer needs, which is in the Context (demand perspective, and the choice of an enterprise to answer these needs with certain Products and services (supply), residing in the Function perspective. So in the event classes from literature

- 8 concern changes in the context of the organization, and can be reason for change in any aspect of the organization;
- 2 concern changes in the function of the organization, representing the choices in response to the context;
- 3 concern changes in the ontology of the organization, and
- 5 concern the implementation of the organization – 3 for parties and people and 2 for ICT.

Remarkably, no events specific for the informational or documental organization are discerned.

Common definitions of agility emphasize the contextual and functional perspective. Sarkis [5] focuses on *metrics* for agility – such as acquisition time, demand change cost and amount of capable workers on certain equipment – just as Tsourveloudis et al. [15], which propose a set of quantitative agility parameters for calculating the overall agility of an enterprise. Van Oosterhout [2, p216] asks for more research to analyze different types of business agility needs, also because he expects that building IT platforms which support all these types of business agility needs will be very expensive. So, all definitions of agility found (including [16], [4], [6]) are mostly black-box or functional, i.e., they agree that

Table 1. Categorization of event classifications in EEF's perspectives (adapted)

Context (environment & demand)			Catastrophic [2] Social/legal [2, 4, 5] Business network [2] Competition [2, 4] Customer needs [2, 4, 5]
Function			Products and services (supply) [5] Quality of Service (QoS) [14]
Construction	Ontology		Processes (business rules) [5] Technology (methods) [4] Internal change [2, 4]
	Implementation	Parties and People	Resources [5] Processes (responsibility) [5] Internal change [2, 4]
		ICT and other means	Technology [2, 4, 5] Internal change [2, 4]
Context (supply)		Parties and People	(available) Resources [5] Social (workforce expectations) [4]
		ICT and other means	(available) Technology [2, 4, 5]

one should be able to change; they do not tell, white-box or constructional, in what respect an organization should be able to change.

Directed searches for organization implementation (variables) did not yield anything useful. On top of earlier mentioned literature, Google Scholar searches (in English and Dutch) were performed with the terms organization(al) implementation, organization(al) change, organization(al) aspects, organization(al) dimensions, organization(al) design, and (organization(al)) implementation aspects.

The only frameworks that seem to deal with organization implementation, using other terminology however, are the COPAFILTH[2] framework [17, 19] and Hoogervorsts Framework for Enterprise Engineering [18], summarized in Table 2. Half of the aspects mentioned still concern the environment or functional perspective of the organization, the other half concerns the level of construction, of which 3 categories deal with organizational implementation. In each of these categories some examples are mentioned, however no exhaustive list is provided.

So, our broad research question is to have a complete list of organization implementation variables. In this paper we will create a first set of such variables.

2.3 Approach

We will first categorize the variables found in literature in the system types of EEF [13]. Secondly, we will test (deductive step) the completeness of these variables against two different organization implementation descriptions of OMG's

[2] Translation of Dutch COPAFIJTH.

Table 2. Categorization of COPAFILTH [17] and Framework for Enterprise Engineering [18] in EEF (adapted)

			Business	Informational	Documental
Context (environment & demand)			Commerce/Business (demand) [17, 18]		
			Legal [17]		
Function			Commerce/Business (products and services) [17, 18]		
			Organization (flexibility) [17, 18]		
			Administrative (management) [17]		
			Finance [17]		
			Technology (quality and flexibility [17], security [18])		
				Information (supply, quality) [17, 18]	
Construction	Ontology		Administrative (structure) [17]		
			Information (need) [17, 18]	Information (structure) [18]	
	Implementation	Parties and People	Organization (structure and culture [17], culture and processes and employees [18])		
			Personnel [17]		
			Administrative (order) [17]		
		ICT and other means	Technology [17, 18]		
			Organization (technology, means) [18]		
			Housing [17]		
					Information (gathering, storage, distribution) [18]
Context (supply)		Parties and People	Business (suppliers, partners) [18]		
		ICT and other means			

EU-Rent case [8], and possibly extend the list (inductive step). In this paper we will not elaborate the motives to give these variables a certain value – e.g., choose for a specific organizational split [20] – or the coherence of these variables – e.g., more ICT could influence the amount of personnel; we stop at the level of identifying the variables.

3 Variables from Literature

Table 3 summarizes the organization implementation variables as found in our literature search (see subsection 2.2). Recent research in Adaptive Case Management (ACM) [21, 23] elicits the following nuances in these variables.

- some business rules are optional, others are mandatory; it should be possible to document the (lack of) complying to these rules in an *execution trace*;

Table 3. Organization implementation variables in EEF (adapted)

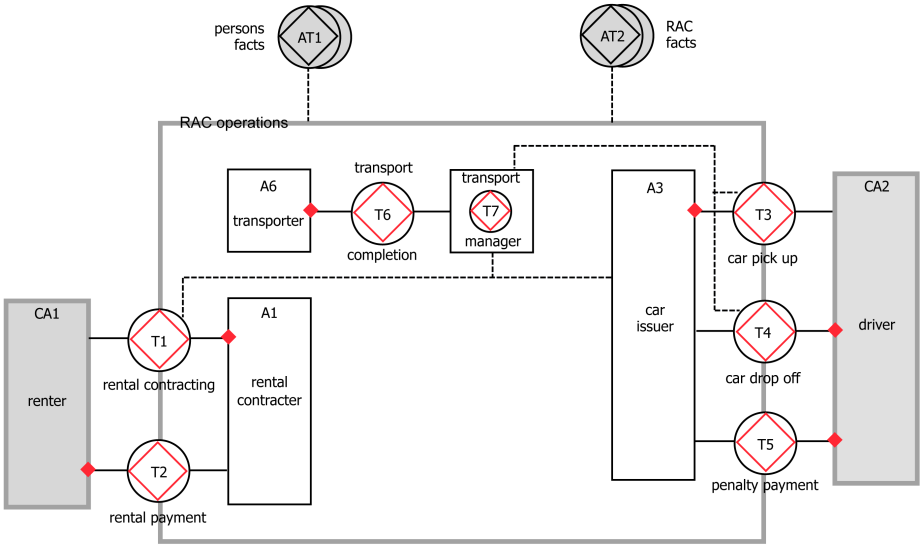
			Business	Informational	Documental
Construction		Ontology	Actor roles, transaction kinds, information links [10, 17, 18][a]		
			Business rules [5, 10], methods [4]		
	Implementation	Parties and People	Organization structure [17]		
			Departments		
			Functionary types[b] [17]		
			Delegation [5]		
			Separation of function[c] [17]		
			Order of working [17, 18]		
			Assignment of tasks[d] [5, 17]		
			#Full-time Equivalent (FTE) [5]		
			Skills and competences [5, 17]		
			Sourcing [2, 5]		
				Language support	
				Data structure[e] [18]	
		ICT and other means	Locations of offices [17]		
			Equipment and infrastructure[f] [2, 4, 5, 17, 18]		
			WFMS and execution trace [21]		
			Degree of automation [17]		
			IT Integration level [5, 17]		
				External data sources [22]	
					D(B)MS, CMS
					Channels[g] [17]

[a] In the Informational world these concern information needed for B-actors, management information, term monitoring, reporting, etc.
[b] including mapping of actor roles (responsibility) to functionary types
[c] e.g. splitting of different steps over different functionary types/persons, or 4-eye principle
[d] top-down, self-regulatory, priorities, teams or individuals
[e] non-structured – e.g., tape (audio/video), (Word, Excel) documents – or structured – e.g., database, XML
[f] Including Man-Machine Interface and GUI
[g] web/email, phone/sms, paper, ...

- some organization implementation variables get their value at the very last moment, even when the process is running; e.g.,
  - which process steps should be performed next in the dealing with this specific case – the so-called *dynamic working plan* as opposed to a fixed work flow,
  - who (which person, team, department or even external organization) is going to perform a certain process step, and
  - what source of data is sufficient to perform a certain task – e.g., a salary-statement or a bank statement to establish credibility.

4 Validation

In this section we will present the organization implementation variable analysis of two different organization implementation descriptions of OMG's EU-Rent case [8] which will then be compared to the variables found earlier (Table 3). The ontology of the B-organization of this case is presented in Fig. 2. A reading guide for this model can be found in [9].



transaction kind	product kind
T1 rental contracting	P1 Rental is contracted
T2 rental payment	P2 the rent of Rental is paid
T3 car pick up	P3 the car of Rental is picked up
T4 car drop off	P4 the car of Rental is dropped off
T5 penalty payment	P5 the penalty of Rental is paid
T6 transport completion	P6 Transport is completed
T7 transport management	P7 transport management for Day is done

Fig. 2. OCD and TPT of Rent-A-Car (adapted from [24])

The first analysis was performed on the Rent-A-Car description [24] (descr. 1). Below, we will show per sentence the organization implementation variable(s) found. For the length of this paper, we cannot present the complete description but summarized the findings in Table 4. The results of the second analysis, performed on the Mini EU-Rent Business Model [25] (descr. 2), can be found in the same table.

> *Rent-A-Car (or RAC for short) is a company that rents cars to persons, both private ones and representatives of legal bodies, like companies.*

This line states that "Rent-A-Car is a company" but it could also have been a network of companies, carrying the same name and embodied in different legal entities. We conclude as implementation variable the embodiment of the ontological model by organizational and legal entities (V1).

> *It was founded by the twin brothers Janno and Ties back in the eighties.*

Contextual sentence, no implementation variables.

> *They started to hire out their own (two) cars, and they were among the first companies that allowed cars to be dropped-off in a different location than they were picked-up.*

From this we find the need for a transportation function between locations which will influence the ontological model of the B-organization. However, no implementation variables are found in this sentence.

> *To this end, Janno and Ties had made agreements with students in several cities.*

We read students are hired to perform some task. So we have an implementation variable regarding the employees in the organization (V2).

> *For a small amount of money, a student would await the arrival of a rented car, e.g. at an airport, and drive it back to the office of RAC, after which the student would go home by public transport.*

From this line we read several things. First, the drop-off location could be anywhere (airport departure hall 3, town center, ...) and not necessarily a RAC office. This implies that the state and accept of the drop-off can happen at any location. For that, the locations of performing certain acts must be defined. Secondly, students are authorized to accept the drop-off, so there is an assignment between employees and act types (during some time frame), and, as the student is not the requester of the drop-off, there is some form of delegation. This implies the students need the relevant information to be available on location, need the right competences to perform this type of task, and possibly need facilities to record the data created. In summary, we found the following implementation variables.

V3: Denotation (syntax) and accuracy of entity types;
V4: Workplace;
V5: Cross-reference which act type can be performed on which location;
V6: Cross-reference which employee is allowed to perform which type of act;
V7: Delegation of act types from functionary type to other functionary type;
V8: Competences/certification.

From the variables regarding employees, location and assignment of tasks (including delegation), the need for information and data recording per employee and per location can be determined. When employees perform different type of acts (possibly involving different transaction kinds), one possibly wants to combine the I- and D-functions to support these acts in one information product.

For example, a student is allowed to perform both drop-offs (accept), pick-ups (state) and transports. If he gets his tasks for a day presented in three lists, he will have to sort out in which order he has to perform his tasks. If he gets this information presented in some Google Maps overview with time tables, in which he can also confirm the end of performing a task, he will better be able to plan the order of performing his tasks.

> *Currently, RAC operates from over fifty geographically dispersed branches in Europe.*

Obviously, again there is the implementation variable about workplace and which type of act is performed at which location – e.g., pick-up can only be done at branches near airports, while drop-off can be done at any branch. However, several questions then arise:

- Who is the addressee of a coordination act (C-act, e.g., request, promise) that is directed to RAC, e.g. the request for rental start? Is it the legal entity RAC, is it a specific branch, or is it an employee at some branch?
- Can a customer request a contract at branch A while the pick-up is done at branch B? Can branch A promise a car rental while the pick-up is done at branch B?
- And if it is necessary for branches to share data, will data be stored locally, centralized or in the cloud?
- Will offices be supported by IT locally (with possibly different systems), shared (using the same systems but locally), or centralized?

From this we summarized the following implementation variables:

V9: Specificity of C-act addressee;
V10: The extent to which the execution of acts within one event, is restricted to the location at which the event is triggered;
V11: Location of data storage (local, centralized, cloud);
V12: Applications, including at which locations.

> *Many cities have a branch, some even several, and there are branches located near all airports. One of the branches is the original office where Janno and Ties started and where both are still around. Being mechanical engineer by education, they have kept loving to drive and maintain cars, even since they are the managing directors of a million euro company.*

Context, no implementation variables.

> *The head of the front office of the home branch is Chiara.*

First, we recognize the notion of departments (front office) and organizational hierarchy. Secondly, we see the notion of functionary type (head of front office), the fulfillment of functionary types by employees (Chiara is head of front office), and the location an employee works at. A question that remains is which type of acts this employee or functionary type performs. We believe the functionary type is the level between employees and actor roles, meaning that V6 must be discarded.

V13: Departments (clustered by e.g. responsibility, competence, market, ...);
V14: Organizational hierarchy;
V15: Functionary types;
V16: Cross-reference employee/functionary type (replacing V6);
V17: Cross-reference employee/workplace;
V18: Cross-reference functionary type/act type.

There are two more desk officers working in this department.

First, we see an amount of FTE for the functionary type desk officer. Second, the question arises how these persons work together. The variables are:

V19: Amount of FTE (per department, functionary type, location, ...);
V20: Per act, way of fulfilling actor role (sequentially, concurrently, or collectively [10, p.125]);
V21: Separation of function.

Customer orders are placed through several channels: walk-in, telephone, fax, and e-mail. Walk-in customers are usually people who want to rent a car immediately. Through the other channels one makes in general advance reservations.

V22: Channels, including degree of integration and availability per C-act and workplace.

These can be made up to 200 days in advance.

This is a business rule and is thus present in the ontological model.

In all cases, an electronic rental form is filled out by one of the desk employees, as input to RACES (RAC Information System).

V23: Medium of entering data (writing, typing, voice)

Note that it is the desk employee who registers the request, delegated by the customer, and the promise. Other variables found are:

V24: Medium of gathering data (ask, search on the internet, get from central registrations (external data source), ...);
V25: Medium of saving data (digitally, paper, human brain, ...);
V26: Medium of receiving information (sound, image, text, ...);
V27: Rules for assigning people to tasks;
V28: Order of working;
V29: Language support.

Comparing Table 4 and Table 3, we conclude that our analysis did not reveal many new elementary implementation variables, but mostly implementation variables regarding the coherence between the (elementary) variables from Table 3. For example, the notions "Functionary type" and "Assignment of tasks" were made specific and complemented in cross-references such as employee X functionary type, functionary type X act type, and functionary type X location. Also, all variables and categories from the earlier tables, except for Sourcing, were found in these case descriptions, confirming existing literature.

Table 4. Full list of implementation variables found in EU-Rent case

	Business	Informational	Documental	descr.
Parties and people	Organization structure: organizational/legal entity			1,2
	Employees and Sourcing			1
	Delegation			1,2
	Competences/certification			1
	Addressee specificity^{NEW}			1
	Departments			1,2
	Organizational structure			1,2
	Functionary types			1,2
	X-ref employee/functionary type^{NEW}			1
	X-ref functionary type/act type^{NEW}			1
	#FTE			1
	Way of fulfilling actor role^{NEW}			1
	Separation of function			1
	Order of working			1,2
	Assignment of tasks			1
		Language support		2
ICT and other means	Workplaces (including locations of offices)			1,2
	Equipment (including infrastructure)			1
	X-ref workplace/act type^{NEW}			1
	Event location restrictions^{NEW}			1,2
	Applications (including WFMS, D(BMS), ...)			1
	X-ref employee/workplace^{NEW}			1
		Media (entering, gathering, saving, receiving)^{NEW}		1
			Channels	1
			Denotation^{NEW}	1

5 Conclusions and Future Research

Our ideal was to formulate a list of anticipated changes for which agility is ensured, just like the theory for Normalized Systems [7] did for automated IT systems. This would contribute to uniformity and standardization in the competence of Enterprise Engineers, thus enabling traceability in governing enterprise transformations. Also, where an ontological model gives already a solid starting point for cross-organizationally usable IT applications [22], using the explicit knowledge of organizational implementation variables in an IT platform could turn this into an actual cross-organizationally running IT application. How far have we come with such a list?

First of all, we noticed that our analysis (Section 4) did not reveal new categories, compared to the literature (Section 3), but did reveal implementation variables regarding the coherence between existing variables. For example, the notion Housing (or location) was made specific and complemented in a cross-reference employee X location, act type X location and event location

restrictions. Also, the variables and categories from literature were found in and confirmed by the case descriptions.

To have these variables explicit and operationalized offers opportunities for building IT flexibility in a platform. In the example of the student, who is now only allowed to perform drop-offs (accept) and pick-ups (state), it would be possible to present all his tasks related to that in some Google Maps overview with time tables, in which he can also confirm the end of performing a task. Suppose one day RAC decides to allow students to do transports as well, and the IT knows the notion of functionary type X act type etc., then all connected software applications can use this information to change their – e.g., GUI and security – behavior accordingly, potentially without the need for this software to be reprogrammed. Since changing (also organizational) implementation variables tend to have combinatorial effects [7], the future potential for wider validation and application of this list is significant.

We realize this is a modest start on the way to a complete list of organization implementation variables. Therefore we propose the following future research:

- repeat the procedure from Section 4 for real-life observations or procedure descriptions from large organizations;
- add rigor to each variable found: what exactly is its meaning, and why is this variable positioned in a certain EEF-cell;
- validate with existing IT systems to what extent these variables are explicit, and for the implicit variables, where they hinder organizational flexibility;
- elaborate a model for coordination and work flows, including the assignment of subjects to actor roles or functionary types, the assignment of tasks to subjects, and the prioritizing and scheduling of tasks;
- explore functional / constructional gaps, e.g. Quality of Human Services (QoHS) and Quality of Automated Services (QoAS) as functional with respect to Resourcing and IT support respectively.

References

1. Overby, E., Bharadwaj, A., Sambamurthy, V.: Enterprise agility and the enabling role of information technology. Eur. J. Inf. Syst. 15, 120–131 (2006)
2. van Oosterhout, M.P.A.: Business Agility and Information Technology in Service Organizations. PhD thesis, Erasmus University Rotterdam (June 2010)
3. Dietz, J.L.G., Hoogervorst, J.A.P.: Enterprise Ontology and Enterprise Architecture – how to let them evolve into effective complementary notions. GEAO Journal of Enterprise Architecture 1 (2007)
4. Conboy, K., Fitzgerald, B.: Toward a Conceptual Framework of Agile Methods: A Study of Agility in Different Disciplines. In: Proceedings of the 2004 ACM Workshop on Interdisciplinary Software Engineering Research, WISER 2004, pp. 37–44. ACM, New York (2004)
5. Sarkis, J.: Benchmarking for agility. Benchmarking: An International Journal 8(2), 88–107 (2001)
6. Seo, D., La Paz, A.I.: Exploring the Dark Side of IS in Achieving Organizational Agility. Commun. ACM 51(11), 136–139 (2008)

7. Mannaert, H., Verelst, J.: Normalized Systems: Re-creating Information Technology Based on Laws for Software Evolvability, Koppa, Kermt, Belgium (2009)
8. Object Management Group: Business Motivation Model (BMM) Specification, V1.1. OMG Available Specification OMG Document Number: formal/2010-05-01 (May 2010), http://www.omg.org/spec/BMM/1.1/PDF/
9. Op 't Land, M., Dietz, J.L.G.: Benefits of Enterprise Ontology in Governing Complex Enterprise Transformations. In: Albani, A., Aveiro, D., Barjis, J. (eds.) EEWC 2012. LNBIP, vol. 110, pp. 77–92. Springer, Heidelberg (2012)
10. Dietz, J.L.G.: Enterprise Ontology – Theory and methodology. Springer (2006)
11. Dietz, J.L.G.: Architecture: Building strategy into design. Sdu Uitgevers bv, The Hague, The Netherlands (2008)
12. van Dipten, E., Mulder, J.B.F.: Basic Enterprise Engineering Map. Informatie 10, 54–61 (2011)
13. Op 't Land, M., Proper, H.A.: Impact of Principles on Enterprise Engineering. In: Österle, H., Schelp, J., Winter, R. (eds.) Proceedings of the 15th European Conference on Information Systems, pp. 1965–1976 (2007)
14. Op 't Land, M., Pombinho, J.: Strengthening the Foundations Underlying the Enterprise Engineering Manifesto. In: Albani, A., Aveiro, D., Barjis, J. (eds.) EEWC 2012. LNBIP, vol. 110, pp. 1–14. Springer, Heidelberg (2012)
15. Tsourveloudis, N.C., Valavanis, K.P.: On the Measurement of Enterprise Agility. Journal of Intelligent and Robotic Systems (33), 329–342 (2002)
16. Sherehiy, B., Karwowski, W., Layer, J.K.: A review of enterprise agility: Concepts, frameworks, and attributes. International Journal of Industrial Ergonomics 37(5), 445–460 (2007)
17. BIZZdesign: Handboek Business Process Engineering. Academic version 7.1 (in Dutch) edn. BIZZdesign B.V. Academy Publishers (2009)
18. Hoogervorst, J.A.P.: A framework for enterprise engineering. International Journal of Internet and Enterprise Management 7(1), 5–40 (2011)
19. de Bruin, B., Verschut, A., Wierstra, E.: Systematic Analysis of Business Processes. Knowledge & Process Management 7(2), 87–96 (2000)
20. Op 't Land, M.: Applying Architecture and Ontology to the Splitting and Allying of Enterprises. PhD thesis, Delft University of Technology (2008)
21. Rychkova, I.: Towards Automated Support for Case Management Processes with Declarative Configurable Specifications. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 65–76. Springer, Heidelberg (2013)
22. Krouwel, M., Op 't Land, M.: Using Enterprise Ontology as a basis for Requirements for Cross-Organizationally Usable Applications. In: Figueiredo, A.D., Ramos, I., Trauth, E. (eds.) Proceedings of the 7th Mediterranean Conference on Information Systems 2012 (MCIS 2012). MCIS Proceedings, University of Minho, Portugal, AIS Electronic Library (AISeL), Paper 23 (2012)
23. Scheithauer, G., Hellmann, S.: Analysis and Documentation of Knowledge-Intensive Processes. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 3–11. Springer, Heidelberg (2013)
24. Dietz, J.L.G.: The Essence of Organization - an introduction to Enterprise Engineering. Sapio (2013), to be published @ http://www.demo.nl
25. Schacher, M.: Mini EU-Rent: Business Model. Technical Report v23.06.2008, KnowGravity (2008), http://www.knowgravity.com/pdf-e/Mini%20EU-Rent%20BU.pdf