DEVELOPING INFORMATION SYSTEMS WITH NOMIS
A Model-Driven Systems Development Approach Proposal

José Cordeiro
E.S.T. Setúbal, I.P.S, Campus do IPS, Setúbal, Portugal
jose.cordeiro@estsetubal.ips.pt

Keywords: Information Systems, Information Systems Modelling, Human-Centred Information Systems, NOMIS, NOMIS Vision, NOMIS Models, NOMIS Modelling Notation, NOMIS Metamodel, Model Driven System Engineering, Metamodelling, Model Transformations, Domain Specific Language.

Abstract: NOMIS is a human centred information systems modelling approach that is based on human *observable* actions. It models a business domain through a set of views focusing on human actions, human interactions, context for actions and information. These views are represented by a set of tables and diagrams using the NOMIS graphical notation. The modelling elements depicted in these views are defined in the NOMIS metamodel. NOMIS metamodel and graphical notation are a first step for formalizing and automate the design and implementation of computer applications with NOMIS. In this paper, we propose to develop NOMIS applications using a Model Driven System Engineering approach. The suggested approach will define formally NOMIS models and notation and, using model transformations, will derive a code structure to be used by the computerized information system, and a schema for a relational database supporting business data. Additionally, other components of a specified application middleware will be created.

1 INTRODUCTION

NOMIS (Cordeiro, 2011) is a human centred information systems modelling approach based on human *observable* actions that intends to overcome the difficulty to define an information system with the necessary objectivity and precision. The key to achieve these goals is on the philosophical stance in which NOMIS is based – Human Relativism (Cordeiro et al, 2009). According to Human Relativism, reality is subjective, dependent on the observer, but there is an *observable part* that can be used objectively. The *observability* concept, used in NOMIS, is a starting point to model precisely an information system.

NOMIS models a business domain through a set of views focusing on human actions, human interactions, context for actions and information applying some ideas and perspectives from, respectively, the Theory of Organized Activity (Holt, 1997), Enterprise Ontology (Dietz, 2006) and, Organisational Semiotics (Liu, 2000). The information view is added by NOMIS. These Views are represented by a set of diagrams where NOMIS modelling elements defined in the NOMIS metamodel are shown following a graphical notation proposed also in NOMIS.

Despite NOMIS modelling formalisms, there is no formal use of NOMIS models to create an information system. For example, in Cordeiro, 2017 an e-learning prototype is modelled and created according to NOMIS, however, a model based strategy is used instead of a model driven one. Therefore, in this paper we propose to develop NOMIS applications that uses a Model Driven System Engineering approach. We suggest the creation of a domain specific language to represent NOMIS models based on NOMIS metamodel for the abstract syntax and NOMIS notation for the concrete syntax. It is also suggested some transformations to produce: 1) a code structure to be used by the computerized information system, 2) a schema for a relational database supporting business data and 3) a rule based system to store business rules.

This paper is organized as follows: section 2 gives a brief overview of NOMIS. Section 3 presents a brief overview of MSDE, section 4 introduces and describes the detail of a NOMIS MSDE based solution, section 4 refers related work and, section 5 concludes and points some future research directions.

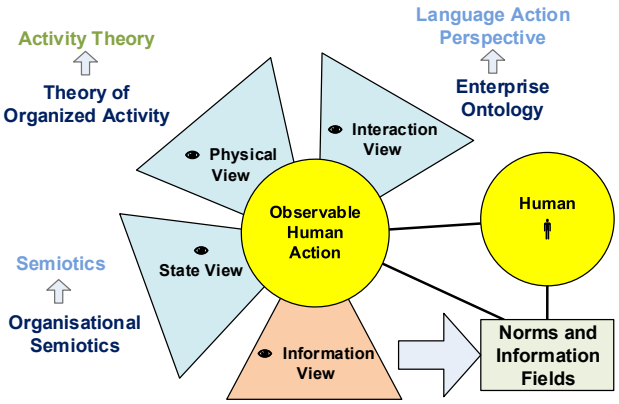

Figure 1: NOMIS Vision – its views and foundational theories

2 NOMIS OVERVIEW

2.1 Introduction

NOMIS – **NO**rmative **M**odelling of **I**nformation **S**ystems is a human centred modelling approach to information systems development (Cordeiro, 2011).

Recognizing the difficulty to define precisely the requirements of an Information System, NOMIS proposes a solution that is based on the *observability* concept: "*what we observe is more consensual, precise and, therefore more appropriate to be used by scientific methods*". In NOMIS, an Information System is a human activity (social) system which may or may not involve the use of computer systems. In these systems what is *observable* are the human physical actions and the physical things and, what is *not observable* are 1) the human mental actions such as decisions, intentions, judgements, goals, etc. that are not externalized and, 2) the *conceptual or informational features* of physical things such as a price or a qualitative aspect of a specific good that are not shown.

Observability is a key concept in Human Relativism (Cordeiro et al, 2009), the philosophical stance in which NOMIS is based.

Seeking to use *observable* elements, NOMIS models information systems with a focus on *human observable actions* including both material and language actions. Straightly connected to human actions is *information*. In fact, information is used as an input, an auxiliary element, an output, even as a target element for human actions. As human actions information also has an observable part and a not observable one: *data* is an observable material support for information and *information* itself, as a meaning extracted by humans from data, is an immaterial part. NOMIS understands *information* as the result of an *interpretation* process coming after *perceiving* the *observed reality*. Information is only available from data after being interpreted by a human. There is no information without a human interpreter.

2.2 NOMIS Vision

Based on *human observable actions*, NOMIS proposes a *vision* of information systems composed by a set of four views – **Interaction View**, **Physical View**, **State View** and **Information View** – addressing, respectively, human interactions, action processes and, context for actions besides an additional view on information.

NOMIS views form a coherent and consistent information system vision from a human *observable* action perspective.

Considering the unpredictable nature of human actions, NOMIS adds *Norms* as human behaviour regulators. Norms addresses and regulates sequences of human actions. Expected (human) behaviour is derived from *systems of norms* or *information fields* as they were defined within Organisational Semiotics (Stamper, 1996). Within an information field people tend to behave in a certain, expected and controlled way. Examples of information fields could be an organisation, a department, or even a family or a team. Information Fields and Norms are a *glue* connecting human actions and information.

NOMIS Overall Vision is depicted in Figure 1.

2.3 NOMIS Models

Modelling, in NOMIS, is done through a set of tables and diagrams that represent NOMIS Views according to NOMIS Vision. The complete set of these diagrams and tables is presented and described in Table 1. The *elements* represented in these diagrams correspond to the key concepts in NOMIS, namely:

- Human Actions
- Actors – human performers
- Bodies – physical things
- Information Items – without physical support
- Language Actions (or Coordination-acts)
- Environmental States

To represent NOMIS elements a modelling notation is also provided (see Cordeiro, 2017).

3 MODEL DRIVEN SYSTEM ENGINEERING

Model Driven System Engineering (MDSE) is a software engineering approach that aims at derive software systems from models. Its core concepts are models and transformations. For specifying these models MSDE uses modelling languages. In a higher abstraction level, metamodels are used to model models themselves and, meta-metamodels to model metamodels. Usually meta-metamodels are the highest abstraction level. Metamodels are useful, specially to define new modelling languages. In this case, also modelling languages are specified by a model (its metamodel). The second core concept – transformations – is used to transform a model into another model according to a set of transformation rules. Ultimately, a transformation can be used to transform a model into code. Although there are other applications, code generation is perhaps the most important application of MDSE.

A good introduction to MDSE can be found in Brambilla et al, 2017.

3.1 Modelling Languages

As stated before, modelling languages are used to specify models and, metamodels can be used to specify modelling languages. Modelling languages can be classified as general-purpose modelling languages or domain-specific modelling languages. A well-known example of a general-purpose modelling language is UML that can be applied in different domains. A domain-specific modelling language, on the other hand, is used to model a particular domain. Examples could be HTML markup for creating Web Pages or SQL for database access. These are text based modelling languages.

A modelling language is defined by 1) an *abstract syntax* that describes the structure of the language, its primitives and the way they are combined, 2) a *concrete syntax* describing its visual representation and appearance and, its *semantics*, the meaning of its elements.

Table 1: NOMIS modelling artefacts

Diagram		Content	Used in…	Observations
HID	Human Interaction Diagram	Actors and their interactions	Interaction View	A kind of Construction Model as used in Enterprise Ontology
ASD	Action Sequence Diagram	Sequences of actions	Interaction View and Physical View	A kind of UML activity diagram
BSD	Body State Diagram	The different states of a body	State View	A kind of UML state diagram
EDD	Existential Dependencies Diagram	Environmental States and their existential dependencies	State View	A kind of an Ontology Chart as used in Organisational Semiotics
ESD	Environmental State Diagram	Details each environmental state showing its elements	State View	A kind of an Ontology Chart as used in Organisational Semiotics
AVD	Action View Diagram	Show all the elements related to a single action	Physical View	Shows Information Items, Bodies, Actors related to an action
ABD	Action Body Diagram	Sequences of action related to body state changes	Physical View	A kind of Diplan as used in the Theory of Organized Activity
ICD	Information Connection Diagram	Information Items related to bodies and actors	Information View	Shows information interpreters and information supporting bodies
Table		**Content**	**Used in…**	**Observations**
HAT	Human Action Table	Human actions, actors, bodies and information items	Interaction View and Physical View	Collects human actions, actors, bodies and information items
HADT	Human Actions Dependency Table	Human actions, their dependencies on Bodies, Information and Context	State View	Collects dependencies for human actions
IIT	Information Items Table	Information Items	Information View	Collects details of information items

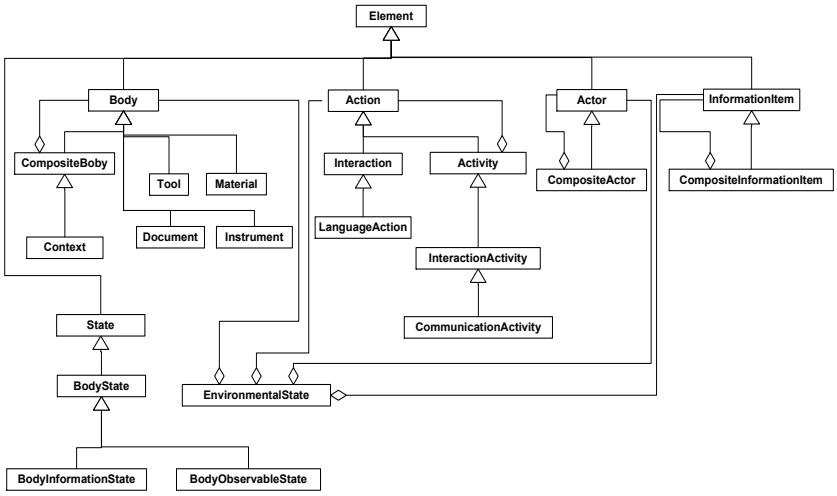

Figure 2: Metamodel of NOMIS Elements

3.2 Model Transformations

Model transformations are the engine of MSDE. They can be used to transform models into code, to refine or refactor models, to translate models, etc. Usually model transformation can be applied between two graphical models, known as Model-to-Model transformations (M2M) or between a graphical model and a text model, these are Model-to-Text transformations (M2T). M2T are usually used to produce source code from graphical models.

4 A MDSE APPROACH FOR NOMIS

NOMIS allow us to model a business domain and give us guidelines for a computerized support system implementation. Effectively, in Cordeiro, 2017 an eLearning system was modelled using NOMIS models and implemented according to NOMIS vision. Also, a system infrastructure for NOMIS applications was proposed. In this system there was no clear connection between NOMIS models and system implementation. Therefore, in this section we propose a Model Driven System Engineering approach to derive part of the system implementation from NOMIS models.

4.1 From NOMIS Models to a MDSE Solution

As mentioned before, with NOMIS is possible to model a business domain using a set of tables and diagrams showing the key elements of NOMIS together with their relationships. What is shown in these diagrams is a representation of NOMIS Vision and is the way NOMIS sees the Information System reality. This is NOMIS language that is formalized with the metamodel reproduced in Figure 2 and a notation (see Cordeiro, 2017). These are the first steps for creating a domain specific language (DSL) for NOMIS. A DSL abstract syntax will be provided by the NOMIS metamodel, whereas a concrete syntax is obtained from NOMIS notation. A DSL is the best and required solution for implementing a MDSE approach for NOMIS.

4.2 Deriving the System Boundary

The central element of NOMIS are human *observable* actions. In a computerized information system these actions correspond to the functionalities to be provided by the system to a user. For example, if a user wants to "store a document" the system will provide this functionality. In this case, the system acts as a tool enabling and supporting human actions.

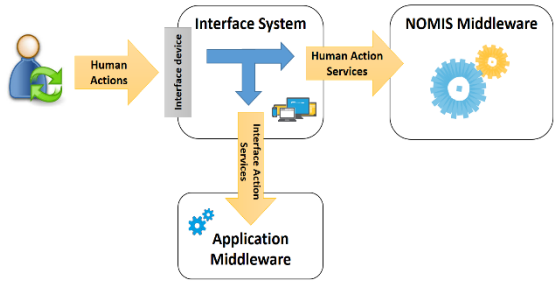

Figure 3: The Interface software component

In general, NOMIS human actions could be implemented as services by a system that would use a typical service oriented architecture (SOA). This is to be achieved by a model transformation where actions in models are transformed to service software interfaces that are used in the implementation of the corresponding functionalities. Also, interactions between two actors are modelled as two services, a request service from a first actor is delivered as a receive service by the second actor.

4.3 An Interface System

Human actors do not interact directly with a computer system. This is done through a mouse, a keyboard, a computer monitor or other type of input/output device. It will be the responsibility of a human interface software component to make the connection between user and services provided by the system. For example, the action "store a document" can be done through pressing an interface button in a monitor screen. The interface software component will be responsible to trigger the appropriate system service in response to this action. This solution permits to design and program the interface separately from the rest of the system. Designers can design the interfaces using different screen widgets and programmers can make the necessary connections to system services.

Some human actions does not need to be stored in the system or have system services for it. This is the case of simple interface actions. For example, in an application forum component, a user usually wants to check received messages. A service could retrieve these messages but the interface component would be responsible for implementing the actions of sorting or paging and, showing the messages freeing the main system from this *usability* task. A diagram of the interface software component is shown in Figure 3.

The separation of interface and system services provides a desired flexibility to adapt or use difference designs and interfaces with the same services in line with user preferences.

Besides the separation between interface functionalities and system services, an interface system can also make use of activity elements from NOMIS models. An *activity* in NOMIS corresponds to a composite of human actions. So, some activities can be mapped/transformed to an application window (or a browser window) in a windows based environment. In this case application windows will make available actions for the different human actions present in the activity. These actions will trigger the associated system services.

4.4 Persisting Business Data

Information is another central element of NOMIS that is represented by the *information item* element in NOMIS models. Each *information item* is associated with a business domain concept and is detailed within an Information Item Table. In Cordeiro, 2017, content of *information items* is stored as records in relational database tables. This is part of a NOMIS Middleware responsible for storing all business related information. In that work is proposed a flexible database structure that allows business terms and concepts to evolve and change without being attached to the underlying structure. Application data, on the other hand, is kept separated from business data outside NOMIS Middleware. Another model transformation should be applied to *information items* that will be used to persist NOMIS application business domain data. This transformation will generate or reuse the described database structure.

4.5 Considering Environmental States

States and Environmental States in NOMIS are used as context or condition for actions. The physical elements, or *bodies* can exhibit different states. A "special" type of body is the human actor. A human actor state, in this case, is a role or roles performed by an actor within a specific context. Different roles give access to different types of actions. Other *bodies* can have states as well. For example, a form can be "filled" or "not filled", a book in a library can be "for lend" or "not for lend", if it is "for lend" can also be "lent" or "not lent". A lent book cannot be lent again before being returned. Therefore, different body states lead to different possible actions. Environmental States (ES) represent a composite of NOMIS elements, namely: *actions*, *bodies*, *actors*

and *information items* where *actors* and bodies may exhibit a particular state. An ES is an utmost important element in NOMIS, it constitutes the information system *anchors*. For example, in a library information system a "membership" ES is a required condition to lend a book (see Cordeiro, 2015). This ES is composed by an actor, in this case a library member, his/her membership data and a "paid membership fee" condition. Only with all this ES elements fulfilled, is possible for a library member to lend a book. "To lend a book" is a key action for a library information system. To acquire a "membership" state there is a registration activity that collects the registration data and the membership fee. This activity is a typical business process. In reality, it is possible to create different registration business processes but the states (ES) stay essentially the same. For a most complete MSDE solution also NOMIS ES and states need to be translated to system features and code using model transformations. In this case, all information necessary for ES is already stored in NOMIS Middleware database but, for example, a state machine can be derived or, at least, complemented with the necessary code to identify and process NOMIS ES elements.

4.6 Normative Nature of Actions

In NOMIS, action sequences are regulated by (*Behavioural*) *Norms*. A *Norm* is defined by a semi-formal analytical representation as defined in Organisational Semiotics (See Liu, 2000):

IF condition **THEN** agent
ADOPTS attitude **TOWARD** something

The condition part refer to an Environmental State that induces a human actor (the agent) to perform a particular action (attitude). So, sequences of actions are not mandatory as they depend on human behaviour. Anyway, an implementation of a business process as a sequence of system services or a service orchestration may have an automatic implementation as long as it is possible to be easily changed. A possible solution, mentioned in Liu, 2000 is a rule based system to store norms known similar to a Normbase (Stamper et al, 1991). Norms are not shown directly in NOMIS Models, instead they can be attached as notes to some NOMIS elements. Norms are written in text and therefore, require a M2T transformation to derive the correspondent rule to be stored in the described rule system.

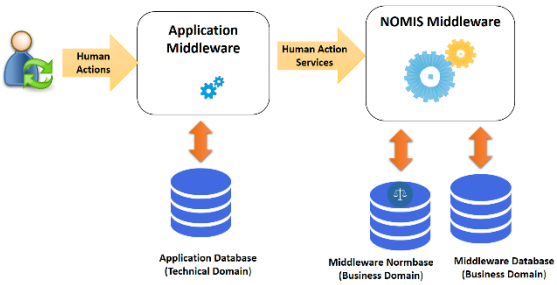

Figure 4: NOMIS Application System

4.7 NOMIS Information Systems

Following our proposed approach, resulting information systems will have a clear separation between the technical and the business domain. NOMIS is used to model the business domain focusing on human actions and information. States, in NOMIS, are used to bring a necessary system stability but they depend mainly on information which is already addressed. On the other hand, the technical domain deals with technical aspects and can be modelled separately. The connection between both domains is derived from NOMIS models through the definition of (human action) services, the schema of a business information database (the middleware database) and a special database for NOMIS norms (the normbase). A diagram of these systems is shown in Figure 4.

4.8 Summary of the NOMIS MSDE Approach

NOMIS models provide a comprehensive, coherent and consistent view of an information system from a business domain perspective. By using an MSDE approach it will be possible to derive the business software elements that will be used by the technical system. This will be achieved by MSDE model transformations of NOMIS Models. For this goal, a first step it to create a Domain Specific Language (DSL) to be used with NOMIS. This DSL abstract syntax will be based on NOMIS Metamodel, whereas its concrete syntax will follow NOMIS notation. Next, model transformations will create part of the computerized information system. These transformations are applied to NOMIS modelling artefacts. As described, each NOMIS key element have a clear correspondence to a technical part: NOMIS actions and interactions are transformed in system services software interfaces, information items are transformed to database tables and bodies

are simply registered as information although *electronic* bodies may also be stored in the technical database. Finally, norms will be stored in a specific database – the normbase.

By using this approach most of the technical part needs to be developed separately but, the boundaries are clearly defined simplifying the technical development work.

5 RELATED WORK

NOMIS was not formalized before as proposed in this work. Hence, there is no known previous work. However, some elements and diagrams in NOMIS have similar concepts in the underlying theories in which NOMIS is based. Effectively, except for the Information View, NOMIS Views are inspired by the theories of Enterprise Ontology (EO) (Dietz, 2006), Organisational Semiotics (OS) (Liu, 2000) and, the Theory of Organized Activity (TOA) (Holt, 1997). Some diagrams in NOMIS are adaptations, improvements or extensions of the diagrams used in these theories. Therefore, related work can be found in written research on these theories, this will be exposed in the next sections.

5.1 Diplans in TOA

The Theory of Organized Activity (TOA) that is related to the Physical View in NOMIS, uses a diagrammatic language – Diplans (Holt, 1988) – to show human actions, bodies, states and their relationships. This is a language similar to Petri Nets but applied to a business environment. In NOMIS is possible to have a similar representation with ABD diagrams. In Cordeiro, 2007 there is a proposal to formalize Diplans with UML profiles. Unfortunately, UML was found not suitable for this task due to some adaption difficulties such as extended UML metaclasses with underlying features that did not match extension classes, limited UML relationship types or limited UML element combinations. To the best of author's knowledge there is no other related research work in Diplans.

5.2 Ontology Charts in OS

Organizational Semiotics (OS) is behind NOMIS State View, where some diagrams inspired by OS Ontology Charts are used to show states and their existential dependencies. This is the case of EDD and ESD diagrams used in the State View. Also, in Cordeiro, 2007, Ontology Charts are modelled with UML profiles. As with Diplans, similar adaptation problems were found with Ontology Charts.

Bonacin et al, 2004 proposes some heuristic rules for class diagram derivation from OCs. This work just gives some hints on how to obtain (and translate) the OC elements into UML elements. Used UML elements are limited to classes and associations, compositions and generalizations relationships between them.

Tsaramirsis and Poernomo, 2008 proposes the generation of a prototype system from Ontology Charts. The solution uses a database structure to store information from the elements in the OC. From NOMIS point of view, this solution is not consistent with the proposed theoretical framework. Tsaramirsis and Yamin, 2014 suggested later the generation of UML 2 use cases from Ontology Charts. They map agents to actors and communication acts to use cases. This transformation is not suitable as well for NOMIS as it does not cover the required detailing.

Santos et al, 2016 made an extensive review on OS literature from 2011 till 2015 covering conferences, journals, and book chapters with 91 publications found. We could not find any related research in those publications.

5.3 Aspect Models in EO

Enterprise Ontology (EO) uses aspect models to model a business system. These aspects models uses a set of diagrams, textual rules and tables for modelling purposes. From EO diagrams, NOMIS only has equivalents for Actor Transaction Diagrams (ATD) and Process Structure Diagrams (PSD) with, respectively, HID and ASD diagrams. They are used in the Interaction View. In Cordeiro, 2008, there is also a proposition for a UML profile for ATD, PSD and Actor Bank Diagrams. Again, there were issues in extending UML with specific profiles for these diagrams that excludes UML profiles as a suitable solution.

EO is the most studied and researched approach from the foundational theories of NOMIS. Therefore, we just mention some relevant research related to this work that can inspire a MSDE approach.

First, Wang et al, 2011 suggests the transformation of EO metamodels to a XML schema. This could be an interesting transformation to be used in a MSDE approach for NOMIS.

den Haan, 2009 provides a complete MSDE based approach for EO. It uses a SOA architecture with a process engine to execute EO models. The overall view is provided.

van Kervel et al, 2012 describes a EO processor that fully automates EO development. Also uses a MSDE approach.

6 CONCLUSIONS AND FUTURE WORK

This paper presented a proposal of a MSDE approach for NOMIS computerized information system implementations. The proposed solution uses a DSL for representing NOMIS models and establishes the guidelines for the necessary model transformations. As a result of these transformations part of the implementation code together with a persistency system for business information and business norms will be created. Using this approach, the technical and the business part will be modelled separately, but the connection points between these parts will be established and derived from NOMIS models.

As future work, it is our intention to create a DSL for this proposal using the Eclipse Modelling Framework. This DSL will include the creation of a concrete syntax using the Graphical Modelling Framework, model validation and model persistence. A prototype of a simple application will be used to validate this approach.

REFERENCES

Bonacin, R., Baranauskas, M. and Liu, K. (2004). From Ontology Charts to Class Diagrams - Semantic Analysis Aiding Systems Design. *In Proceedings of the 6th International Conference on Enterprise Information Systems,* Porto, Portugal. v. 1. p. 389-395.

Brambilla, B., Cabot, J., and Wimmer, M. (2017). Model-Driven Software Engineering in Practice, 2nd edition. ISBN: 978-1627057080. Morgan & Claypool.

Cordeiro, José A. M. (2017). Applying NOMIS - Modelling Information Systems Using a Human Centred Approach. In *Business Modeling and Software Design*, ed. Shishkov, Boris, 27 - 45. ISBN: 978-3-319-57222-2. Berlin, Heidelberg: Springer International Publishing

Cordeiro, José A. M. (2015). A New Way of Modelling Information Systems and Business Processes - the NOMIS Approach. In *Business Modeling and Software Design*, ed. Shishkov, Boris, 102 - 118. ISBN: 978-3-319-20051-4. Berlin, Heidelberg: Springer International Publishing.

Cordeiro, J. (2011). *Normative Approach to Information systems Modelling*. PhD Thesis. The University of Reading, UK.

Cordeiro, J. and Liu, K., (2007). UML 2 Profiles for Ontology Charts and Diplans - Issues on Meta-modelling. In *Proceedings of the 2nd International Workshop on Enterprise Modelling and Information Systems Architectures*. St. Goar, Germany.

Cordeiro, J. and Liu, K., (2008). A UML Profile for Enterprise Ontology. In Proceedings of the 2nd International Workshop on Enterprise Systems and Technology. Enschede, the Netherlands.

Cordeiro, J., Filipe, J. and Liu, K., (2009). Towards a Human Oriented Approach to Information Systems Development. In *Proceedings of the 3rd International Workshop on Enterprise Systems and Technology*. Sofia, Bulgaria.

Dietz, J., (2006). *Enterprise Ontology, Theory and Methodology*. Springer-Verlag, Berlin Heidelberg, Germany.

den Haan, J., (2009). An Enterprise Ontology based approach to Model-Driven Engineering, Master's thesis, Delft University of Technology.

Holt, A., (1997). *Organized Activity and Its Support by Computer*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Holt, A., (1988). 'Diplans: a new language for the study and implementation of coordination'. In ACM Transactions on Information Systems (TOIS) Volume 6, Issue 2, pages: 109 – 125.

van Kervel, S., Dietz, J., Hintzen, J., van Meeuwen, T. and Zijlstra, B., (2012). Enterprise Ontology Driven Software Engineering. In S. Hammoudi, M. van Sinderen & J. Cordeiro (eds.), ICSOFT (p./pp. 205-210), : SciTePress. ISBN: 978-989-8565-19-8

Liu, K., (2000). *Semiotics in Information Systems Engineering*. Cambridge University Press, Cambridge, UK.

de Souza Santos, M., Bertãozini. B., Neris, A. (2016). Studies in Organisational Semiotics: A Systematic Literature Review. In: Baranauskas, M., Liu, K., Sun, L., Neris, V., Bonacin, R., Nakata, K. (eds) *Socially Aware Organisations and Technologies. Impact and Challenges*. ICISO 2016. IFIP Advances in Information and Communication Technology, vol 477. Springer.

Stamper, R., Liu, K., Klarenberg, P., Van Slooten, F., Ades, Y. And VanSlooten, C. (1991). From Database to Normbase. In *International Journal of Information Management*. Vol. 11, pages 67-84.

Tsaramirsis, G. and Poernomo, I., (2008). Prototype Generation from Ontology Charts. In *Fifth International Conference on Information Technology*. New Generations, Las Vegas, NV, pp. 1177-1178.

Tsaramirsis, G. and Yamin, M., (2014). Generation of UML2 Use Cases from MEASUR's Ontology Charts: A MDA Approach. In *Model-Driven Business Process Engineering*, eds. Lano, K., Zandu, R. and Maroukian, K. page 67 - 76. ISBN: 978-1-60805-893-8. Shariqah, United Arab Emirates, Bentham Science Publishers Ltd.

Wang, Y., Albani, A., Barjis, J. (2011). Transformation of DEMO Metamodel into XML Schema. In: Albani A., Dietz J.L.G., Verelst J. (eds) *Advances in Enterprise Engineering V*. EEWC 2011. Lecture Notes in Business Information Processing, vol 79. Springer, Berlin, Heidelberg.