Enterprise Architecture Coherence and the Model Driven Enterprise: Is Simulation the answer or are we flying kites?

Balbir S. Barn
School of Engineering and Information Sciences
Middlesex University
London, UK
b.barn@mdx.ac.uk

Tony Clark
School of Engineering and Information Sciences
Middlesex University
London, UK
t.n.clark@mdx.ac.uk

Martin Loomes
School of Engineering and Information Sciences
Middlesex University
London, UK
m.loomes@mdx.ac.uk

ABSTRACT

Aligning information communications technology (ICT) to business goals is a common issue cited by senior executives and recent research in measuring alignment provides evidence that those organizations that have aligned successfully their business and IT strategy will out perform those that have not. Enterprise Architecture (EA) aims to capture the essentials of a business, its IT and its evolution, and to support analysis of this information and is thus seen as an important tool for this alignment requirement. However, existing methods, techniques, languages and supporting technology for EA may not be sufficient for helping deliver this agenda and increasingly, simulation is perceived as one such solution. Simulation however presents other challenges, notably, the problem of validity versus credibility. This paper charts a philosophical route through a discussion on models and what they represent, the communication structures implicitly required for models to work and proposes that model based simulation of a model based enterprise can be more effective if there is a theoretical basis to a simulation model. This hypothesis is evaluated by a re-interpretation of Toulmin's Argumentation model as a candidate for the underlying theory for constructing simulations of enterprise architecture coherence. The result of which is used to define an integration strategy with our existing work on lightweight enterprise architecture modelling processes.

Categories and Subject Descriptors

D.2.1 [**Requirements/Specifications**]: Tools

General Terms

Modelling, Enterprise Architecture, Simulation

Keywords

Simulation, Argumentation, Model Driven Engineering

ISEC '13, February 21-23, 2013 New Delhi, India

1. INTRODUCTION

Business and IT alignment has remained an ongoing concern for organisations [12] and researchers have addressed the importance of alignment and in particular the need for congruence between business strategy and and IT strategy [2]. Orthogonal but related to this is the development of practices around Enterprise Architecture (EA) specifically to support types of business analysis such as alignment between business functions and IT systems; business change describing the current state of a business (as-is) and a desired state of a business (to-be) means that EA has the potential to provide the basic machinery that can be used to address business and IT alignment [16].

EA aims to capture the essentials of a business, its IT and its evolution, and to support analysis of this information: '[it is] a coherent whole of principles, methods, and models that are used in the design and realisation of an enterprise's organisational structure, business processes, information systems and infrastructure.' [11] A key objective of EA is being able to provide a holistic understanding of all aspects of a business, connecting the business drivers and the surrounding business environment, through the business processes, organisational units, roles and responsibilities, to the underlying IT systems that the business relies on. Hence alignment between business goals and IT strategy seems to be at the heart of EA.

Despite the opportunities presented for addressing business and IT alignment needs using EA, the current state of the art of EA presents issues such as: large unwieldy methods such as those derived from TOGAF [23]; a lack of precision in the methods because of ambiguity between concepts, overlap of concepts and perhaps most importantly, no clear refinement relationships between business goals and enterprise architecture component changes. Some evidence of this latter issue can be noted in the Archimate language [10] which is now beginning to include concepts such as goals as part of its meta model. In essence, the languages, methods, frameworks used to described EA are limited in how they can adequately address the key emergent issue of EA coherence.

Increasingly there is an interest in applying simulation to the EA context for example, in the style of serious games to better understand and bridge the business and IT alignment gap and the role of EA in addressing that gap. Our response to the challenges of limited technology for EA has resulted in the development of a range of technologies to support both the modelling of EA and a simulation based environment for evaluating the "why" of EA [5, 4]. While such an approach is viable there is a need to develop and refine theories to support how simulation models, execute and present their results. This can be summarised as seeking to provide *credibility* of simulation models. To address this need, we revisit and re-interpret theories from other disciplines as a way of informing this debate. Thus far, we are conscious that the ideas and discussions presented here are conceptual in nature but such conceptualisations require wider informed debate but still have an established route to implementation and evaluation.

In this paper, the primary contribution is the proposal that Toulmin's Argumentation Theory [24] is one such candidate for evaluating the credibility as opposed to the validity of simulation models. In doing so, the paper outlines how such an evaluation opens up new avenues of research for the EA and Software Engineering community. Examples of new avenues of opportunity include a re-interpreting units of reuse and software lifecycles and their focus on the artefact rather than the process.

The remainder of this paper is structured as follows: section 2 traces a short philosophical inquiry around simulation, models and theory building in general; section 3 presents Toulmin's argumentation model in the context of asserting the credibility of models; section 4 proposes how the argumentation model can be formalised and integrated into simulation environments. Finally section 5 considers future work and some of the implications arising from this inquiry.

2. MODELS, SIMULATION, AND THEORY BUILDING

A computer simulation is the use of a computer-based dynamic model of some referent system. The referent system may or may not exist in full or in part. Such a model, as explained in Pidd [18] can be defined as follows:

> "an external and explicit representation of part of a referent system as seen by the people who wish to use that model to understand, to change, to manage and to control that part of the referent system." [17].

Simulations are used because they can be cheaper, faster and safer than actual experimentation. As a simulation is intrinsically a model, questions around the goodness of a model are dominant. The quality of the underlying model of a simulation has the potential to the make a significant impact on the interpretation of information arising from the simulation. Model quality can be described along a set of (partial) dimensions such as: correctness (is the model constructed using the appropriate and agreed syntax?); completeness (is all necessary information available?); consistency (are there any contradictions?) and comprehensibility (is it understood by its intended users?) [13]

Pidd provides a more condensed argument and proposes the question is one of *validity* versus *credibility*. Model validity is different from verification which concerns formal properties of a model, such as syntactical correctness, consistency, and completeness and can often be automated. Validity, however, is inherently an informal process requiring subjective human judgement. The assessment of the validity can be compromised by the size and complexity of the underlying model and the relative skills and experience residing with the human intervention,

Models Models, irrespective of the level of their formality, cannot possibly be fully validated so a more reasonable requirement is to ensure that the model is fit for purpose [18] and is credible. This credibility is determined by passing a number of tests such as those proposed by Sergeant [22] and Balci [1].

In a previously rehearsed argument for a critique on Naur's seminal paper: Programming as Theory Building [14], we have argued that models are the construction and vehicles for explanations of theories. Note, that for the purposes of this discussion the knowledge, insight or theory that the programmer has come into possession of is a theory in the sense of Ryle [21]. That is, a person who has a theory knows how to do certain things and can support the actual doing with explanations, justifications and responses to queries. That insight or theory is primarily one of building up a certain kind of knowledge that is intrinsic to the programmer whilst any auxiliary documentation remains a secondary product. However, as in a software programme where the lines of code cannot be simply interpreted to derive the theory, nor can the language model elements be easily interpreted for a theory. We suggest that validating or asserting the credibility of the model is therefore akin to theory building and theory sharing. Of course executions or explanations are needed and one such route is the use of simulation[7].

Returning to the original problems located in enterprise architecture, our response to the challenges of limited technology for EA has resulted in the development of a range of technologies to support both the modelling of EA and a simulation based environment for evaluating the "why" of EA [5, 4]. Although our emergent language and technologies (LEAP) for simulation has an executable semantics, the problem noted by Pidd, that of asserting the credibility of the resultant simulations remains. Given that we anticipate such credibility is subjective, our inquiry discussed in this paper is that of seeking how to integrate the human element in asserting credibility (or as we prefer to view it: as theory building) and testing via the simulation environment. In so doing, we note that Dalianis has similarly argued for this approach in [7]. Related to this, is research that has attempted to provide discursive descriptions for negotiating models [20], recognition that modelling is an interpreted activity [19]where an encompassing domain model is interpreted in a particular way to construct new models; and also where there is a specific focus on the fundamentals of process of modelling [8]. The latter is particularly interesting as the authors consider that whilst validation is important, it "... usually depends on subjective judgments passed and viewpoints held by humans." . This interest in the nature of the modelling process as well as its final product is reminiscent of Naur and his criticism for methods. The tendency of methods research in the IS discipline is to propose algorithmic steps to analysing and designing solutions to problems. As Naur notes: "A method implies a claim that program development can and should proceed as a sequence of actions leading to a particular kind of documented result". In contrast, a theory building view holds that a theory "held by a person has no inherent division into parts and no inherent ordering"[14]. At large, IS/SE research is embarked on a journey based on epistemological foundations and as a consequence has mostly neglected *techne* (the technical know how of getting things done) and *phronosis* (wisdom derived from socialised practices) [25]. Like Naur, we suggest there would seem to be a case for methods research that suggests more attunement with the effects that methods may have in the education of those involved rather than with the creation of artefacts. That is, the creation and embedding of tacit knowledge rather than the production of artefacts representing explicit knowledge through an algorithmic process.

The act of producing a simulation that describes some aspect of an enterprise architecture is about communication of an underlying theory. For example, when we want our simulation to show how an architecture description supports the ability to provide organisational reports on budgetary requirements we are requiring that architecture description to provide us with a theory of budgetary models and if it meets our understanding of such models then the simulation acquires credibility. We want also, to engage with the simulation models through a questioning process with a corresponding explanation or response. Whilst Hoppenbrouwers et al [8],provides some specific types of modelling dialogue such as those that might be present when a domain expert converses with a systems analyst, a structure that has the potential for integration with the meta model for the simulation model is missing. It is this element that we propose in this paper.

3. A PROPOSAL FOR VALIDATING THE CREDIBILITY OF A SIMULATION

As we have noted earlier, a simulation model may present outputs that meet the needs of some intended change to an organisation's enterprise architecture. The credibility of the outputs as distinct from the validation of the outputs remains an open question thus the user should ideally explore that simulation model and in so doing, engage in some dialogue with that model. How should that dialogue be constructed such that credibility can be asserted? We propose that a missing component in a simulation model is a theory that can usefully be used to support that dialogue. One such candidate theory is Stephen Toulmin's argumentation theory [24] which describes the dialogue structure of arguments. When this is combined with a simulation language technology such as the LEAP language then the Toulmin Argumentation Model (TAM) is a route to offering credible explanations of simulations.

In this section we introduce an initial language design (the abstract syntax) for integrating the Toulmin model with our LEAP simulation language. The LEAP language proposes that EA is fundamentally about representing and analysing data-rich and highly-structured executable systems at different levels of abstraction. It has been designed as a synthetic language where distinctions between many business concepts are deemed fundamentally irrelevant (domain specific presentation issues being viewed as perfectly respectable sugar). The key concepts in LEAP are structuring elements located around a component model, where the component model supports: data encapsulation, functions attached to components; messages between components; rules for defining component behaviour and conditions to express business goals and directives as invariants over the state of a component. LEAP has a precise semantics in the form of an operational implementation and an associated tool for graphical display and simulation. More details of the language and simulation technology is available [3, 5, 4].

3.1 The Toulmin Argumentation Theory

Toulmin describes an argumentation approach that is derived from jurisprudence rather than mathematics. Using his approach, an argument structure is initiated when a *claim* is made. If the claim is challenged then it must be justified or made good and it is the structure of making it good that forms the essence of the argumentation model. The claim asserts some proposition based on some grounds to support the claim. If the listener is not convinced by the *grounds* for the claim, then further grounds may be provided. Finally, a *warrant* may be required to show that grounds are indeed a basis for the claim. Warrants correspond to "practical standards or canons of arguments. They are likely to incorporate elements of domain specific knowledge. Circumstances might mean that the grounds and warrants are insufficient. In such cases, a backing form part of further arguments. Backings are evidence that the warrant is reliable. Grounds, warrants and backings may be subject to qualifiers which are used to assert the relationship between grounds and claims. There will be situations where the claim may not follow the grounds despite the production of warrants and backings, these situations are termed *rebuttals*. Of note, is the importance of arguments that are field-invariant (holds for any domain) versus field-dependent (hold for a specific domain).

Claim A proposition representing a claim being made in an argument;

Grounds One or more propositions acting as evidence justifying the Claim;

Warrant One or more rules of inference describing how the Grounds contribute to the Claim;

Backing The knowledge establishing the Grounds for believing the Warrant;

Qualifier A phrase qualifying the degree of certainty in the argument for the Claim;

Rebuttal One or more propositions challenging the validity of the Claim.

This argumentation approach can be a guide for gathering information from across a conceptual model for explaining a particular aspect of a claim and is particularly useful at the instance level, i.e. claims about particular objects and their relationships, thus it's potential role in providing a level of credibility for a simulation model by way of providing a satisfactory explanation of a sequence of outputs provides a case for its usage.

Let us consider a simple example. The hypothesis of this paper is a claim that the Toulmin Argumentation Model (TAM) has a form of structure that lends itself to use as as explanation tool in simulation. The grounds for this is an analysis of the "Uses of Argument text book [24]. A warrant is provided for by reference to the paper by Dalianis et al [7]. The backing further emphasising the credibility of the claim as the warrant is supported by a backing such as the credibility of the warrant being based on the quality of the conference (double blind peer review by three independent experts). Some qualification is necessary based on a rebuttal which states that such a claim can only be realistically achieved if there is integration between both simulation and argumentation models. The form and flow of the argumentation thus lends itself to an explanation form.

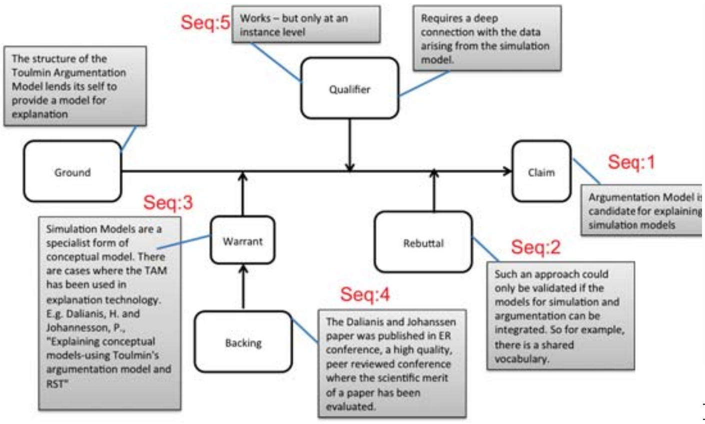


Figure 1: Argumentation as an approach for credibility of simulation models

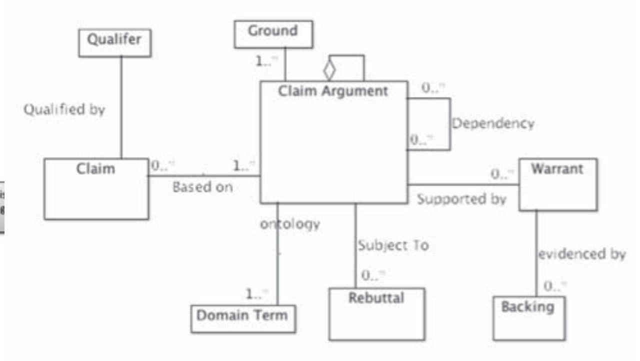


Figure 2: Toulmin Argumentation Model Abstract Syntax

4. TOULMIN ARGUMENTATION MODELS AS A LANGUAGE APPROACH

In order to investigate the integration needs between TAM and simulation models it is first necessary to define the concepts underpinning TAM. One way to arrive at a computationally relevant understanding is to treat TAM as a language design activity. This is an appropriate approach as Toulmin's original work proposes concepts and informally describes a visual layout for representing argument structures such as that seen in Figure 1. Furthermore, by utilising a language design approach, we are in essence, providing warrants and backings for our claim illustrated also in Figure 1. One intended consequence of such an approach is that it lends itself to potential easier integration route with the language utilised in the simulation technology forming part of the LEAP infrastructure. Once the language design is complete, the second step is to implement the language using a meta modelling tool or by hand-coding.

Model driven language engineering approaches typically include components focussed on: a defining an abstract syntax to represent the features of a language; concrete syntax representing the human friendly representations of the language; a semantic domain describing the definitions of the meaning of things represented in the language and various mappings between these components.

Simple UML-style class models and associated constraints can be used as a suitable meta-language for representing the language components listed above.

We propose the following abstract syntax based on Toulmin's concepts discussed earlier. We introduce one central construct - the Claim Argument which acts as a coherent focal point for managing the set of arguments that are relevant to particular simulation. This concept also supports what we consider to be the fractal nature of argumentation models, that is, claim arguments can be assemblies of claims, and can utilise dependencies between claims. Furthermore, concepts themselves can be treated as claims themselves for example, a Warrant can be claim itself and be treated as an argument. The set of domain terms represent a potentially restricted vocabulary for use in instances of all the other constructs.

The abstract model syntax can be further embellished by introducing well formedness rules. Some example rules might include: No Claim can exist without corresponding Grounds; and If a rebuttal is presented, then a Warrant or Backing must be supplied. Consistency rules should ensure that terms referred to in concepts do not contradict each other.

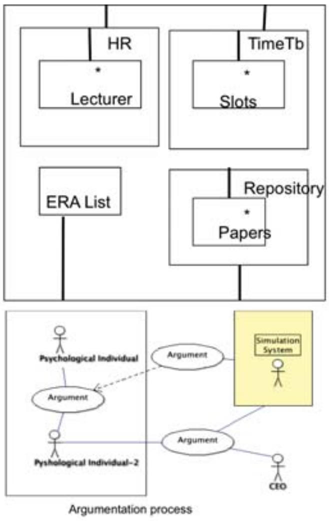
4.1 Tooling

The next step is to provide an implementation of the language, that is, a tool that provides a binding of the various syntactic and semantic models and thus allows users to construct argumentation models. To develop a proof of concept of this language, we plan to utilise the meta modelling toolset MetaEdit+ [9]. Meta Edit+ is a software toolset that supports the design and implementation of domain specific language and uses its own meta modelling language (GOPRR). The abstract syntax for TAM can be encoded in the GOPRR modelling language within the MetaEdit+ toolset in order to define the concrete syntax and the production of a tool for argumentation modelling.

4.2 Using Argumentation

Argumentation presents multiple use cases for the purposes of explanation of a simulation model (See lower left side of Figure 3). An integrated explanation system might work in the following way. Two psychological individuals [15] engage in argument-1, this argument is then translated to an argument-2 that is used as a basis for an interaction with the simulation system. The two individuals then utilise the system to engage in an argument with the end user, say, the CEO using the system simulation in an interactive fashion to persuade the CEO of the credibility of simulation. Thus two key uses emerge: Firstly, the simulation system provides pre-packaged explanations of the simulations - these can be derived from the goals of the simulation and their purpose is to provide cohesive explanations and a demonstration of how a goal is being met by the simulation. Secondly, the simulation architect uses argumentation/explanation models to walk through simulations with the end-user. As rebuttals occur, the architect can draw on instance data from simulation models. This is a more interactive use of the argumentation model and the simulation.

Both use cases require some level of language and tool level integration between simulation and argumentation models, the models proposed in this paper is a first effort at providing that basis for integration. In this section we provide an



Claim	Fact	State
	Lect-1	D1 Clear
	Lect-2	D2 Clear
Claim-1	TT	Rechedule
	Lect-1	D1, D2 Clear
Claim-2	Lect-1	Research Active
	Lect-2	D3, D4 Clear
	TT	Rechedule
	Lect-1	4 WKs Clear
	Lect-1	Paper-1
	Lect-2	4 WKs Clear
	Lect-2	Paper-2
Claim-3	Paper,1,2	Rank A

Figure 3: Components in the Simulation

example scenario on how argumentation can be used for a supportive explanation of a simulation.

4.3 Case Study Scenario

The corporate plan for the University of Scrabbleshire (UoS) has decided to include a business / strategic goal that aims to the research reputation of the institution. The measure for this goal will be based on the relative higher position in the periodic research rankings that the UK Government produces. The question arises: how will the EA of UoS need to be changed to align with this goal? The simulation language LEAP and its accompanying environment can potentially answer this question but a further question arises on how credible such a simulation is.

The systems impacted are identified and configured (we have described elsewhere how this can be achieved [6]). The resulting schematic (left hand side) represents the executing environment, whilst the right side shows some partial facts and their state. Thus there are four key systems at play: The HR system includes data about Lecturers; the TimeTB system describes a series of slots where classes are scheduled; the Repository is a store for holding all papers that have been published by academic staff and the ERA List is an external database that ranks outlets such as conferences and journals according to their quality.

As part of the specification in LEAP, a number of business rules supporting assumptions are defined. These rules could include:

Rule-1: If the Timetable schedules two clear days (D) for a Lecturer Then the Lecturer is Research Active

Rule-2: If the Timetable schedules four clear weeks (WK) for a Lecturer Then the Lecturer will produce a Paper

Rule-3: If the Paper is ranked A/B in the ERA List Then the Paper is of high quality

As the system executes the state of the system changes as the various facts associated with each simulated system component change. A claim whether the original goal can be met can be explained and justified by the simulation outputs. For example claim-1 (shown on the rhs of figure 6.) cannot be justified as the current grounds for an increase in quality are not met. This is so, for both Lect-1 and Lect-2 (two lecturers in the system), for illustrative purposes, do not have two consecutive days in which to become research active. Claim-2, however, can be justified as the grounds (two clear consecutive days) are available to both lecturers in the timetabling system so our two lecturers are research active. At Claim-3, two of the lecturers in our story now have 4 clear weeks in the timetable, and warrants are available to back up the claim as they have produced two Papers (1 and 2) which are now available in the repository. Further grounds to support the warrants can be generated as both papers are ranked “A” against the widely available ERA List that provides relative rankings of computer science conferences. As we noted earlier, use cases of an integrated argumentation and a simulation system can be such that the integrated system can be used both to define an interaction for canning as part of the simulation or the system can be used to engage with a third party to provide a credible explanation of the simulation and therefore how the goal is to be met.

A sub-text of such an argumentation is the type of modelling dialogue that may be occurring. These may include: the dialogue that occurs when an analyst learns about the intended system from a domain expert; dialogues when two opposing but well-developed views combine to identify similarities with the ultimate aim to arrive a full reconciliation; or finally one of colliding views where protagonists have differing views that need to be at least partially maintained leading to conflicts reflecting world view conflicts. As Hoppenbrouwers et al note:

> Modelling of this type will have to involve negotiation or argumentation about a common model. [8]

While this characterisation of dialog may be prevalent, utilisation of integrated models of argumentation with simulation remains open research territory.

5. CONCLUSION

This paper has presented a conceptual proposal that simulation models are a useful way for understanding EA strategic alignment. However the validity or credibility of a simulation model is non-trivial issue and requires alternative perspectives for addressing this area. We have proposed that Toulmin's argumentation model is a candidate route for supporting and developing credibility of simulation models. To enable this to happen a number of technologies are required to move from a conceptual position to one where an appropriate technology is viable. The steps to producing these technologies include: formalising Toulmin's AM as an exercise in language design; provisioning of tools that support such a language and integrating with simulation technologies such as LEAP. Given this our next steps are to first continue the formalising of the Toulmin AM as a language by developing tooling by using a software such as MetaEdit+; then integrating the conceptual models underpinning the TAM and our LEAP simulation technologies and so providing an overarching connected architecture.

Once a platform that can perform simulations of EA and provide credible explanations is in place, there is a potential to revisit and re-think our approach to methodologies for EA. We noted earlier, that like Naur we suggest there would seem to be a case for EA methods research that suggests more attunement with the *effects* that methods may have in the education of those involved. That is, the creation and embedding of tacit knowledge rather than the production of artefacts representing explicit knowledge through an algorithmic process could be more powerful as it is closer to the act of theory building. However, the need for re-use of theory (the understanding of goals and how the EA delivers those goals in this paper) is still paramount. Argumentation models as canned elements of simulation could be a candidate for re-usable chunks of theory knowledge.

6. REFERENCES

[1] O. Balci. Validation, verification, and testing techniques throughout the life cycle of a simulation study. *Annals of operations research*, 53(1):121–173, 1994.

[2] Y.E. Chan and B.H. Reich. IT alignment: what have we learned? *Journal of Information Technology*, 22(4):297–315, 2007.

[3] T. Clark and B. Barn. Event driven architecture modelling and simulation. In *Service Oriented System Engineering (SOSE), 2011 IEEE 6th International Symposium on*, pages 43–54. IEEE, 2011.

[4] T Clark, B Barn, and S. Oussena. A method for enterprise architecture alignment. In *PRET Industrial Track*, CAISE 2012. LNBIP, 2012.

[5] T. Clark, B.S. Barn, and S. Oussena. Leap: a precise lightweight framework for enterprise architecture. In *Proceedings of the 4th India Software Engineering Conference*, pages 85–94. ACM, 2011.

[6] T. Clark, B.S. Barn, and S. Oussena. A method for enterprise architecture alignment. *Practice-Driven Research on Enterprise Transformation*, pages 48–76, 2012.

[7] H. Dalianis and P. Johannesson. Explaining conceptual models-using toulmin's argumentation model and rst. In *In Proceedings of The Third International workshop on the Language Action Perspective on Communication Modelling (LAP98*. Citeseer, 1998.

[8] S. Hoppenbrouwers, H. Proper, and T. van der Weide. A fundamental view on the process of conceptual modeling. *Conceptual Modeling–ER 2005*, pages 128–143, 2005.

[9] MetaCase Inc. Metaedit+ workbench - build your own domain-specific modeling language, 2009. `http://www.metacase.com/mwb/`.

[10] M. M Lankhorst, H.A H.A. Proper, and J Jonkers. The Anatomy of the ArchiMate Language.

[11] Marc Lankhorst. Introduction to enterprise architecture. In *Enterprise Architecture at Work*, The Enterprise Engineering Series. Springer Berlin Heidelberg, 2009.

[12] J. Luftman. Assessing business-IT alignment maturity. *Strategies for information technology governance*, page 99, 2004.

[13] P. Mohagheghi, V. Dehlen, and T. Neple. Definitions and approaches to model quality in model-based software development-a review of literature. *Information and Software Technology*, 51(12):1646–1669, 2009.

[14] Peter Naur. Programming as theory building. *Microprocessing and Microprogramming*, 15(5):253 – 261, 1985.

[15] G. Pask. *Conversation, cognition and learning*. New York: Elsevier, 1975.

[16] C.M. Pereira and P. Sousa. Enterprise architecture: business and IT alignment. In *Proceedings of the 2005 ACM symposium on Applied computing*, page 1345. ACM, 2005.

[17] M. Pidd. Reusing simulation components: simulation software and model reuse: a polemic. In *Proceedings of the 34th conference on Winter simulation: exploring new frontiers*, pages 772–775. Winter Simulation Conference, 2002.

[18] M. Pidd. *Tools for thinking: modelling in management science*, volume 1987. Wiley, 2003.

[19] HA Proper and T.P. Weide. Modelling as selection of interpretation. *GI Proceedings*, 2005.

[20] P. Rittgen. Negotiating models. In *Advanced Information Systems Engineering*, pages 561–573. Springer, 2007.

[21] G. Ryle. The concept of mind (1949). *London: Hutchinson*, 1984.

[22] R.G. Sargent. Verification and validation of simulation models. In *Proceedings of the 37th conference on Winter simulation*, pages 130–143. Winter Simulation Conference, 2005.

[23] J. Spencer et al. *TOGAF Enterprise Edition Version 8.1.* 2004.

[24] S.E. Toulmin. *The uses of argument*. Cambridge Univ Pr, 2003.

[25] Boris Wyssusek. A philosophical re-appraisal of peter naur's notion of "programming as theory building". In *European Conference on Information Systems (ECIS)*, 2007.