Towards an Investigation of the Conceptual Landscape of Enterprise Architecture*

D.J.T. van der Linden[1,2], S.J.B.A. Hoppenbrouwers[2], A. Lartseva[3], and H.A. (Erik) Proper[1,2]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
{dirk.vanderlinden,erik.proper}@tudor.lu
[2] Radboud University Nijmegen, The Netherlands
stijnh@cs.ru.nl
[3] Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands
a.lartseva@fcdonders.ru.nl

Abstract. In this paper we discuss our preliminary work on clarifying the conceptual landscape of Enterprise Architecture. We do so to aid in the integration of conceptual models originating from different communities (of language users, concerns, viewpoints etc.). We propose that discovering the basic ontological structure used by these communities is necessary for the effective integration of models, and that different communities have a distinguishable different central understanding of some categories in their ontology. Our initial results include the description and categorization analysis of several languages and methods used in EA (as used by their creators), which suggest a prototype structure reflecting a community's focus.

Keywords: enterprise architecture, ontology, model, integration.

1 Introduction

Enterprise Architecture (EA) involves (the modeling of) many aspects, e.g. (business) processes, value exchanges, specifications, implementations, deployments and so on. These aspects are often handled by experts, who use specific languages or methods, rife with their own specialized terminology. As a result, these models are often created and maintained by separate communities and cannot trivially be related to each other.

* This paper results from the FNR ASINE project of CRP Henri Tudor and the Agile Service Development (ASD) project (http://www.novay.nl/okb/projects/agile-service-development/7628), a collaborative research initiative focused on methods, techniques and tools for the agile development of business services. The ASD project consortium consists of BeInformed, BiZZdesign, CRP Henri Tudor, HU University of Applied Sciences Utrecht, IBM, Novay, O&i, PGGM, RuleManagement Group, Radboud University Nijmegen, TNO, Twente University, Utrecht University, and Voogd & Voogd. The project is part of the program Service Innovation & ICT of the Dutch Ministry of Economic Affairs, Agriculture and Innovation.

T. Halpin et al. (Eds.): BPMDS 2011 and EMMSAD 2011, LNBIP 81, pp. 526–535, 2011.

A holistic perspective on the enterprise requires the integration of all specific views and related representations, which is a difficult endeavor. That integration presents a major issue is demonstrated by the amount of research and proposed methods for it [3,12,17]. What can be distilled from this research as a general trend is that integration requires that we respect the existence of the individual communities, and that we do not attempt to fix the issue by changing them. Instead we need to look towards how we (architects) deal with their models and products.

Efforts to integrate modeling are not new. Information system and model integration is covered by a staple of computing science efforts, yet most of them have focused on syntactical, and to a lesser extent, semantic integration. What is still missing, or has had less attention paid to it is local differences in semantics, comparable to 'local dialects' wielded by aspect communities.

To follow this line, we need a clearer understanding of what specific communities mean with their models, i.e. we need to reason about the meaning instead of the words and phrases they use to represent them. For a given aspect this means reasoning about a community's shared concepts – their conceptual landscape. We can use Discourse Communities for this (cf. [9]), which can function as a delimiter of the shared conceptual landscape. Perelman and Olbrechts-Tyceta [20] reason how meaning can only be understood in context of those shared factors:

> "All language is the language of community, be this a community bound by biological ties, or by the practice of a common discipline or technique. The terms used, their meaning, their definition, can only be understood in the context of the habits, ways of thought, methods, external circumstances, and tradition known to the users of those terms."

We propose that the users of some given modeling language or method in some typical context of use constitute such a community. Their shared habits (i.e. way and style of modeling) and ways of thinking (i.e. focus and construct availability), whether they existed prior to, or arose by influence of the language or method, are the differentiating factors from other (modeling) discourse communities, and as such can be used to find their shared common meaning.

In order to study these groups we have to represent their conceptual landscapes. This can be done best with ontology, as it is exactly that: a "formal explicit specification of shared conceptualization" [8]. However, it is necessary to clarify the structure and nature of ontology as it is often misused in computing. This specification must be understood as a representation or discovery of that conceptualization, and not a constructed or engineered optimal solution. Indeed, ontology in its original meaning concerns the (socially-constructed) categories of existence used by some group of people, and thus functions as a primer of what exists for these people in the 'real' world.

Where much existing work goes wrong is in not fully acknowledging the significance of categorization claims and commitments inherent in ontology (cf. Wyssusek's work on clarifying the mismatched basis of the Bunge-Wand-Weber ontology for information systems [26]). Furthermore, it is not uncommon for ontologies to be simply based on terminological categorization (cf. Uschold *et al.*'s “the Enterprise Ontology” [25]) with no explicit regard to conceptual categorization. Most worrying is that (parts of) standard ontologies (e.g. Cyc, SUMO, Bunge-Wand-Weber) are used for a specific aspect with little regard to adapting them to their conceptual needs. This causes them to be stipulative in nature (i.e. we say or assume that people will see the world as dictated by some ontology) instead of descriptive (i.e. we find that people see the world in terms of their ontology) in describing the specific aspect of the enterprise.

When we adopt a descriptive approach and set out to discover ‘the’ ontology of some aspect, we need to keep in mind the structure of the categories and concepts therein. While most methods forgo this and implicitly assume a classical, atomistic structure (i.e. concepts are this or that), we find it more fitting to apply a graded approach, as exemplified by Prototype Theory. It holds that categorization does not occur on a strict comparison of necessary or sufficient properties, but that new potential members are compared to the most central (i.e. best exemplar) member of that category (the prototype). As such there is no perfect equality in a category but a graded structure where, for a given community, some members are better examples of some category than others.

This theory was initially proposed by Rosch [21], adopted by many others and used in a variety of investigations. It has been demonstrated in the categorization of abstract objects [1], and research has shown that events are categorized distinctly as well [14]. Because of this we can apply Rosch’s theory to our concerns and the related discourse communities. What a prototype view adds is the ability to capture discourse community specific differences in interpretation, i.e. to find out whether users of some specific language or method share a different general conceptualization of a given category than other communities.

A descriptive ontology can then be used to resolve integration issues on a local semantic level by explicitly mapping each used word or phrase to the actual category of being. As such, it serves as a connecting layer between each community’s models and the shared conceptualization discourse across communities. If we know what a word, phrase or given meaningful element in a model originating from some community actually means, we can know whether those elements can be integrated or not.

However, to get to this stage several key steps are necessary. The discourse communities have to be found and defined in such a way that they and their models can be studied. The contents of their ontology (i.e. the way they see the world) have to be found and defined and several verification cases have to be passed to study the correctness (and exhaustiveness) of any integrated model.

Hence, the contribution of this paper is twofold: discovering the initial ontological structure of the conceptual landscape that is shared amongst the different communities, and finding a starting point for prototype investigation by offering a set of discriminatory properties which help find the central members of a community’s ontological categories. We believe this line of work will eventually lead to empirical data than can be used as input for other frameworks concerned with similar issues, such as the SEQUEL framework [11] and its ‘social quality’ aspect, i.e. the agreement (or Semantic Reassurance [9]) between modelers.

The remainder of this paper is structured as follows. In Section 2 we discuss our approach to studying the terms and meanings used by discourse communities and how we selected the languages and methods to study. In Section 3 we show the results of this categorization process and argue that they demonstrate a preference of different communities for different best examples of some categories: prototypes. Finally, we summarize our contribution in Section 4 and discuss the future work we will undertake to further validate our hypothesis.

2 Approach

In this section we discuss our approach to finding the discourse communities to study and the procedures used to analyze the data originating from this.

2.1 Selecting the Discourse Communities

We selected a variety of languages and methods originating both from academia and industry to ensure a wide pragmatic range in the respective language communities (albeit focused on the Western world). Where possible we based ourselves on official specifications (i.e. documents approved by standards management groups such as the OMG). If no such standardization or widely agreed upon specification existed, we based ourselves on the most complete and widely used information. An overview of the languages and methods selected as input material is given in Table 1. As we aimed to keep the analysis manageable in the initial phases there were many, more comprehensive, languages (e.g. MEMO, SysML, USDL etc.) and methods (e.g. TOGAF, IAF, MDA, etc.) that we did not include. Instead, we aimed to keep the input focused on languages and methods dealing with specific aspects, and not on comprehensive methods that attempt to solve the integration issue by stipulatively defining a shared vocabulary and way of working.

2.2 Categorization

The lexical representations of an aspect's units of meaning (i.e. a language's constructs or a method's given terminology) were gathered and classified according to their source of origin. Care was taken to prevent unwanted loss of information (i.e. collapsing multiple homonyms into one before the categorization process). During an iterative process we denoted (and subsequently refined) the (aspect-specific) categories that emerged from the meanings given for each term. These initially led to many specialized (almost atomic) categories, such as ACTOR[1], ACTORS, ARTIFICIAL ACTOR and so on. These were collapsed into the most common denominator (e.g. ACTOR) so that it would be possible to

[1] In order to avoid confusion between category and the words and phrases that populate it, a category will always be referred to in SMALL CAPS, while its population is shown in *italic*.

study whether prototype effects occurred within these categories for different communities. Finally each lexical representation was categorized into the final set of categories based on the meaning defined in the discourse community's specifications (e.g. *end event* being a typical RESULT before being considered a kind of EVENT in BPMN). In doing so we found a categorization (see Table 2) wide enough to fit the explicit differences found in the source material, while still remaining minimal in the amount of categories. Furthermore, the categorization is limited to the lexical representations of those constructs describing entities, as we found it undesirable to (superficially) cloud the analysis for the dubious virtue of including the many existing relationships within EVENT category.

Finally, we attempted to find possible clashing interpretations that could arise during integration and assigned them to a specific discriminant. We did so by means of introspective Semantic Differential analysis [18], i.e. looking for descriptions that would have both matches for themselves and their antonyms. For instance, the word *actor* could, depending on the discourse community, both be used to represent a human or non-human concept (for example, a person or a computer). From that clash it is deducible that 'human-ness' is a discriminating factor. The results of this analysis and the set of discriminants we found so far is shown in Table 3.

3 Discussion

The results of our analysis hint both at a common, shared categorization of concepts between different aspects of EA and at a difference in preferred, or central terminology. Thus, there seems to be a common conceptual ground that exhibits prototypical structure correlated to specific communities.

Some examples of this are members of the ACTOR category. In e3Value an *actor* is necessarily a being with some very specific properties (e.g. economic independence), while in a language often used with it, i*, it can also be other things, like a computer. Thus, while both communities share the common conceptual ground of an actor being some active entity, someone in the e3Value community would quickly associate it with what they see as economically independent beings (i.e. humans), while someone in the i* community can be expected to have a more generic abstract association.

Another example concerns members of the RESOURCE category. A language with a focus on deployment of software has many words for describing *hardware* that is needed, and as such a speaker of ADeL, ITML or other languages with comparable focus would readily associate RESOURCE with material objects that are needed to do something. Someone with other concerns, say architecture as expressed in ArchiMate, or more abstract process design like BPMN, would more quickly associate RESOURCE with some *information*, *knowledge*, or *environmental data* (i.e. immaterial objects) needed for something to be produced.

Yet more examples can be found in the RESTRICTION category, as many members thereof differ in how they deal with 'necessity' of compliance to some restriction. On the one hand, languages tasked with implementation or technical

Table 1. The discourse communities we used as the initial input for our analysis

Source	Aspect	Reason for inclusion
ArchiMate [23]	Architecture	A standard for EA modeling, it is one of the more expansive sources of EA-specific terminology.
ADeL [19]	Software deployment	The Application Deployment Language aims at describing (and validating) deployment of IT, which is an aspect that is missed in many other languages.
BPMN [16]	Processes	The Business Process Modeling Notation is a standard for process modeling and useful as a source as its language community is heavily involved with, and promotes abstract process thinking.
e3Value [5]	Value exchanges	While similar to other process thinking languages, e3Value has a very specific economic viewpoint, offering terminology specialized for value exchanges.
GRL [13] & i* [6] & KAOS [2]	Specifications	Goal-oriented Requirements Language, i* and Knowledge Acquisition in automated specification are widely used languages/methods used in goal thinking.
ITML [24]	Implementation and deployment	The IT Modeling Language offers terminology specific to those implementing and deploying software, which, as said earlier, many other languages lack.
ARIS [22]	Architecture	The Architecture of Integrated Information Systems is a framework whose community's terminology is on the intersection of process modeling and EA.
Balanced Scorecard [10]	Performance	An analysis framework intended for more than just (technical) modelers, it is a source of commonly used terminology by an important language community within EA; the not-necessarily technically inclined.
Game theory [15]	Behavior	Von Neumann's original concept and its terminology is often used in lieu of more 'standard' modeling languages when dealing with economic aspects.
RBAC [4]	Security	The terminology defined by Role-based access control is a near defacto-standard way of describing security issues, even outside of technical descriptions thereof.
VPEC-T [7]	Architecture (analysis)	The "Lost in Translation" approach offers an analysis framework that can be used by more than just those with technical expertise. For this reason it is worthwhile to take the terminology and its respective language community of non-technical users into account.

factors often deal with logical, computational laws, and therefore use words like *rule* in the alethic sense: restrictions that logically cannot be broken. On the other hand, those speakers dealing with human aspects often find themselves prescribing deontic rules or restrictions which can be violated, even though this is improper. It is obvious how, depending on the view of a speaker, his interpretation of the RESTRICTION category would be biased towards rules that are either alethic or deontic in nature.

Table 2. The categories and their members resulting from our analysis of the languages and methods. Numbers denote the originating community per Table 1's order.

Category	Class Members
ACTOR	unit[1], requirement unit[2], actor[1,4,5], role[1,5,10], collaboration[1], player[9], infrastructure/application component[1], device[1], application software[7], organizational unit[7], position[7,5], perspective[8], market segment[4], hardware role[6], software role[6], hardware[6], software[6], organizational role[6], environment/software agent[5]
EVENT	event[1,5,7,11], behavior[1], function[1,7], interaction[1], activity[3], task[3], business rule/service task[3], transaction[3], start event[3], intermediate event[3], value activity[4], value interface[4], value offering[4], connection[4], move[9], contribution[5], correlation[5], dependency[5], means-ends[5], control[5], goal refinement[5], monitor[5], operation[10], operationalisation[5], performance[5], operation[5], initiatives[8]
GOAL	goal[5,6,7], hard-goal[5], soft-goal[5], business goal[6], achieve goal[5], assignment[5], avoid goal[5], cease goal[5], expectation[5], maintain goal[5], requirement[5], consumer needs[4], belief[5], value[11], target[8]
PROCESS	organizational/infrastructure service[1], information/other service[7], service[1], IT service[6], process[3], sub/business/process flow[3], business process[6], dependency path[4], game[9], task[5]
RESOURCE	artifact[1,2], location[2,6], hardware[2,6] , cpu[2], hd[2], memory[2], software[2], value[1], data/business object[1], object[5], node[1], network[1,6], network device[6], representation[1], meaning[1], device[1], computer hardware[7], machine resource[7], environmental data[7], data input[3], input[5], value object[4], information[9], resource[5], content[11], value port[4]
RESTRICTION	interaction[1], contract[1], interface[1], message[7], catching[3], throwing[3], boundary[3,5], rule[9], decomposition[5], belief[5], priority[6], license[6], boundary condition[5], conflict[5], domain property[5], permission[10], value[11], policy[11], trust[11], (non)interrupting[3], strategy[9], measure[8], strategic objective[8]
RESULT	product[1], human/material/service output[7], data output[3], outcome[8], end event[3], payoff[9]
STATE	dependency element[4], dependency boundary[4]

Some counterintuitive categorizations can also be found in a number of communities. There is, for instance, a tendency for descriptions or types of things to be seen as specific kinds of those things. Some members of the ACTOR category like ARIS' *position*, ArchiMate's *organizational role* or RBAC's *role* are used by discourse communities not in the sense of *a role*, but as *some [actor] entity with the role of*. Similarly, depending on the focus of a community, words that are commonly seen as properties (for instance, *location*), can also be regarded as concrete concepts, like *location* being a (limited yet necessary) RESOURCE for those communities dealing with the (physical) deployment of machines.

While it is true that some of these differentiations of meaning between language communities might not be intended, factors like historical compromises in the design or implementation of a language (cf. ArchiMate's awkward handling of information as a physical entity [23], ch. 6.4), still allow for such differences to occur. In time those differences can become (unintentionally) entrenched in the

Table 3. Discriminants we found which can be used to differentiate category members

Discriminant	Explanation
Natural	Whether something was intentionally created (be it with a purpose or not) or naturally occurred. For example, an artifact versus a location or an actor versus some hardware.
Human	The human condition appears both in the sense of actors, for example a (human) actor or some software. It also seems to appear in composed structures, for example an organizational unit or market segment being necessarily subject to the human condition, a collaboration not so.
Composed	If something is composed of multiple parts (be it an aggregation or complex structure). For example, a single actor versus a organizational unit, or a cpu being a single resource, while a piece of hardware is seen as more than one.
Necessary	The difference between logical (im)possibility and probability. For example, a belief being a restriction that can be broken, while a (logical, or natural) rule is logically impossible to be broken.
Material	Something exists either as a physical, material object or is a (metaphysical) conception. For example, a materially existing resource can be some hardware, while a piece of information can be just as much of a resource without existing as a material object.
Intentional	Something is done, or provoked (be it with or without a reason) instead of spontaneously occurring. For example, an event can be a spontaneous occurrence while a transaction is always intentionally provoked
Vague	Something is either determined explicitly or is vague and left open for interpretation. For example, a hard goal or a soft goal.

conceptual landscape of a language and its speakers, especially if they are not quickly corrected. As a result, they will drift away from a shared understanding with once similar communities, and risk becoming a distinct community.

Knowing all this, it is obvious how important it is to discover these differences in meaning to know exactly what is meant by a model. In order for such information to be usable, let alone have a lasting usefulness, we need to explicitly map the results onto an ontology. As a starting point we have identified the basic categories (i.e. lower ontology) of the communities (of language creators) we analyzed. To figure out at a superficial level whether two communities mean the same thing by a word is thus to see if it at least belongs to the same category. This is a worthwhile endeavor as it can efficiently clear up confusion when the same word is used for different categories (i.e. homonymy). For instance, *hardware* is seen as a RESOURCE by some language communities and as an ACTOR by others.

More detailed mapping requires some more abstract, generic category of being, i.e. an upper ontology. The discovery of the upper ontology for this purpose is not a trivial matter, nor should an existing upper ontology be arbitrarily adopted. Discriminants (Table 3) can be used for this if we can find different combinations of discriminants that require different categories to exist and help to rule out existing upper ontologies that would not fit well.

For example, an e3Value *actor* and an ArchiMate *organizational unit* both belong to the ACTOR category. However, *actor* in this sense is naturally occurring, human and not composed, while *organizational unit* is not naturally occurring, human, and composed. Thus, while both communities speak of superficially the same thing, the mapping shows they are not interchangeable. Different words mapping to the same concept can be found in the same way. Take for example the RESULT category. An *outcome* in ARIS is often seen as a non-naturally occurring, non-materially intended 'thing', which would be the same as an *end event* is seen in BPMN. Thus, in this context the mapping shows the words are interchangeable.

To summarize, we have observed that different discourse communities seem to exhibit preference for the specific use of terminology, and thereby have different 'central' point(s) of a category. Thus, this work is a first critical step towards discovering the prototype structure of EA's conceptual landscape.

4 Conclusion and Future Work

In this paper we have argued for the use of descriptive, not stipulative, ontology as a means of promoting integration and a better shared understanding between (models originating from) different discourse communities in EA. We have created a starting point for such an EA-specific ontology by analyzing the language constructs and their meanings as they were defined by their creators. Furthermore, we have demonstrated several discriminating factors that set apart terminology within a given category, suggesting that these categories are based on prototypes reflecting their originating community's focus.

Future work will involve further investigation of the exact category structure for different discourse communities (e.g. process modelers, enterprise architects, software designers, etc.). We will do so by using the categories and discriminants from Tables 2 & 3 as a starting point for a Semantic Differential [18] experiment adapted to these groups. These experiments will result in datasets that can be used to characterize how different types of communities (and thus modelers) implicitly use the language constructs (different from their specified semantics), which can aid in integration by pointing out where semantic gaps are likely to occur. Further validation, depending on the outcomes of the first phase, may incorporate in-depth interviews and physiological conditioning experiments to also discriminate between conscious and subconscious semantic connections.

References

1. Adelson, B.: Comparing natural and abstract categories: A case study from computer science. Cognitive Science 9(4), 417 (1985)
2. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program. 20, 3–50 (1993)
3. Dijkman, R.M., Quartel, D.A.C., van Sinderen, M.J.: Consistency in multi-viewpoint design of enterprise information systems. Inf. Softw. Technol. 50(7-8), 737–752 (2008)

4. Ferrariolo, D., Cugini, J., Kuhn, R.: Role-based access control (rbac): Features and motivations. In: Proc. of the 11th Annual Computer Security Applications Conference (1995)
5. Gordijn, J., Yu, E., van der Raadt, B.: e-service design using i* and e3value modeling. IEEE Software 23, 26–33 (2006)
6. Grau, G., Horkoff, J., Yu, E., Abdulhadi, S.: i* guide 3.0. Internet (August 2007), http://istar.rwth-aachen.de/tiki-index.php?page_ref_id=67
7. Green, N., Bate, C.: Lost in Translation. Evolved Technologist (2007)
8. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5, 199–220 (1993)
9. Hoppenbrouwers, S.J.B.A.: Freezing language: conceptualisation processes across ICT-supported organisations. Ph.D. thesis, Radboud University Nijmegen (2003)
10. Kaplan, R.S., Norton, D.P.: The balanced scorecard - measures that drive performance. Harvard Business Review, 71–79 (January-February 1992)
11. Krogstie, J., Jørgensen, H.D.: Quality of interactive models. In: Olivé, À., Yoshikawa, M., Yu, E.S.K. (eds.) ER 2003. LNCS, vol. 2784, pp. 351–363. Springer, Heidelberg (2003)
12. Lankhorst, M.M.: Enterprise architecture modelling–the issue of integration. Advanced Engineering Informatics 18(4), 205–216 (2004)
13. Liu, L., Yu, E.: From requirements to architectural design - using goals and scenarios. In: Software Requirements to Architectures Workshop, STRAW 2001 (2001)
14. Majid, A., van Staden, M., Boster, J., Bowerman, M.: Event categorization: A cross-linguistic perspective. In: Proc. of the 26th Annual Meeting of the Cognitive Science Society (2004)
15. von Neumann, J.: Theory Of Games And Economic Behavior. Princeton University Press, Princeton (1944)
16. Object Management Group: Business process model and notation (bpmn) ftf beta 1 for version 2.0. Internet (2010), http://www.omg.org/spec/UML/2.0/
17. Opdahl, A.L., Berio, G.: Interoperable language and model management using the ueml approach. In: Proc. of the 2006 International Workshop on Global Integrated Model Management, pp. 35–42. ACM, New York (2006)
18. Osgood, C.E., Suci, G.J., Tannenbaum, P.: The Measurement of Meaning. University of Illinois Press, Urbana (1957)
19. Patig, S.: Modeling deployment of enterprise applications. In: Soffer, P., Proper, E. (eds.) CAiSE Forum 2010. Lecture Notes in Business Information Processing, vol. 72, pp. 253–266. Springer, Heidelberg (2011)
20. Perelman, C., Olbrechts-Tyteca, L.: The New Rhetoric: A Treatise on Argumentation. University of Notre Dame Press (June 1969)
21. Rosch, E.: Natural categories. Cognitive Psychology 4(3), 328–350 (1973)
22. Scheer, A.-W., Nüttgens, M.: ARIS architecture and reference models for business process management. In: van der Aalst, W.M.P., Desel, J., Oberweis, A. (eds.) Business Process Management. LNCS, vol. 1806, p. 376. Springer, Heidelberg (2000)
23. The Open Group: ArchiMate 1.0 Specification. Van Haren Publishing (2009)
24. Ulrich, F., Heise, D., Kattenstroth, H., Ferguson, D.F., Hadar, E., Waschke, M.G.: Itml: A domain-specific modeling language for supporting business driven it management. In: Proc. of the 9th OOPSLA Workshop on DSM (2009)
25. Uschold, M., King, M., Moralee, S., Zorgios, Y.: The enterprise ontology. The Knowledge Engineering Review 13(01), 31–89 (1998)
26. Wyssusek, B.: On ontological foundations of conceptual modelling. Scandinavian Journal of Information Systems 18, 63–80 (2006)