On the Identification of Modeler Communities

Dirk van der Linden\textsuperscript{1,2,4}, Stijn J.B.A. Hoppenbrouwers\textsuperscript{3,2,4}, and Hendrik A. Proper\textsuperscript{1,2,4}

\textsuperscript{1} Public Research Centre Henri Tudor, Luxembourg, Luxembourg
dirk.vanderlinden,erik.proper@tudor.lu
\textsuperscript{2} Radboud University Nijmegen, Nijmegen, the Netherlands
\textsuperscript{3} HAN University of Applied Sciences, Arnhem, the Netherlands
stijn.hoppenbrouwers@han.nl
\textsuperscript{4} EE-Team, Luxembourg, Luxembourg\textsuperscript{*}

Abstract. We discuss the use and challenges of identifying communities with shared semantics in Enterprise Modeling (EM). People tend to understand modeling meta-concepts (i.e., a modeling language’s constructs or types) in a certain way and can be grouped by this understanding. Having an insight into the typical communities and their composition (e.g., what kind of people constitute a semantic community) can make it easier to predict how a conceptual modeler with a certain background will generally understand the meta-concepts s/he uses, which is useful for e.g., validating model semantics and improving the efficiency of the modeling process itself. We have observed that in practice decisions to group people based on concept use are often made, but are rarely backed up by empirical data demonstrating their supposed efficacy. We demonstrate the use of psychometric data from two studies involving experienced (enterprise) modeling practitioners and computing science students to find such communities. We also discuss the challenge that arises in finding common real-world factors shared between their members to identify them by and conclude that there is no empirical support for commonly used (and often implicit) grouping properties such as similar background, focus and modeling language.

Key words: enterprise modeling, conceptual understanding, personal semantics, community identification

1 Introduction

The modeling of an enterprise typically comprises the modeling of many aspects (e.g., processes, resources, rules), which themselves are typically represented in a specialized modeling language or method (e.g., BPMN [Object Management Group, 2010], e3Value [Gordijn et al., 2006], RBAC [Ferrariolo et al., 1995]). Most of these
languages share similar meta-concepts (e.g., PROCESSES, RESOURCES, RESTRICTIONS\(^5\)). However, from language to language (and modeler to modeler) the way in which meta-concepts are typically used (i.e., their intended semantics) can differ. For example, one modeler might typically intend RESTRICTIONS to be deontic in nature (i.e., restrictions that ought to be the case, but can be violated), while a different modeler might typically consider them as alethic conditions (i.e., rules that are strict logical necessities and cannot be violated). The modelers could also differ in whether they typically interpret RESULTS as being material or immaterial ‘things’. Even for scenarios as simple as the delivery of a pizza these differences become apparent, as a pizza delivery can include alethic restrictions in order to observe temporal dependencies (“A pizza cannot be delivered before it is made.”), deontic restrictions (“A pizza should be delivered within 30 minutes of its order.”), and the result of the delivery can be a material thing (a certain amount of notes and coins of the local currency) or an immaterial one (a confirmation of payment on a debit card machine). If one is to integrate or link models (i.e., the integrative modeling step in enterprise modeling [Lankhorst, 2004, Kuehn et al., 2003, Vernadat, 2002, Opdahl and Berio, 2006, Delen et al., 2005]) and ensure the consistency and completeness of the involved semantics, it is necessary to be aware of the exact way in which such a meta-concept was used by the modeler. If this is not explicitly taken into account, problems could arise from, e.g., treating superficially similar concepts as being the same or eroding the nuanced view from specific models when they are combined and made (internally) consistent.

This challenge follows from the collaborative nature of enterprise modeling [Ssebuggwawo et al., 2009, Rospocher et al., 2008, Frederiks and Van der Weide, 2006, Hoppenbrouwers et al., 2006, Hoppenbrouwers et al., 2005], as it involves different people specialized in different aspects of the enterprise. These aspects have to be elaborated on to deal with the complexity of (re)designing modern day enterprises [Barjis et al., 2009]. Collaborative modelling in general [Rouwette et al., 2008, Hoppenbrouwers et al., 2009, Rittgen, 2009] deals with challenges like these that arise because of the different people involved, such as optimizing the actual modeling process [Bidarra et al., 2001], ensuring its effectiveness [Dean et al., 2000] and dealing with conflicts and problems that arise when integrating models made by different people with different viewpoints [Renger et al., 2008].

The particular challenge we are concerned with in the enterprise modeling process is mismatched understandings between different modelers and stakeholders [Kaidalova et al., 2012]. Note that mismatched understanding does not only refer to misunderstandings that the involved parties might be aware of. It explicitly also refers to the (more damaging) misunderstanding that the parties involved might not be aware of.

People might disagree on what words to use, what they should mean, or use the same words without realizing they talk about different things. When these apparent or hidden disagreements extend to the words used by a modeling language (i.e., the meta-concepts), the produced models themselves might

\(^5\) To distinguish concepts from words used for them we print concepts in SMALL CAPS.
On the Identification of Modeler Communities

no longer reflect correctly or fully the conceptualization of the individuals involved. As models should be there to support the building of knowledge and its exchange [Stahl, 2000], any threat to the validity and integrity of the models is a threat to the knowledge exchange itself. An often used strategy to deal with this is a priori agreeing on or working towards a set of standardized terminology and semantics. However, it is neither safe nor effective to simply assume that such expressed agreements, or even the models themselves, express correctly and completely the way a modeler conceptualizes them [Guarino et al., 1994].

To deal more effectively with the issue of semantics it is necessary to have an insight into the ‘mental models’ of the people involved [Uschold, 2011, Almeida, 2009]. It is important to gain such insight on a personal level because “semantic memory for concepts is based on a subject’s memories of past experiences with instances of those concepts” [Geeraerts, 2010] and because people generally do not think in the semantics of a given modeling language, but in the semantics of their own natural language [Sowa, 2010]. Furthermore, some modeling languages do not have an official, agreed-upon specification of their semantics (e.g., i* [Ayala et al., 2005]) and if they do, there is no guarantee that their semantics are complete or consistent (cf., [Breu et al., 1997, Nuffel et al., 2009, Wilke and Demuth, 2011]). In addition, language users might, deliberately or not, ignore the official semantics and invent their own [Henderson-Sellers, 2005]. Understanding the intended semantics of a given model thus cannot come solely from knowledge of the language and its semantics, but also requires us to invest in understanding the people who created the model.

However, one cannot realistically be expected to look into each individual modeler’s semantic idiosyncrasies. Instead, a generalized view on how people with a certain background typically understand the common meta-concepts could be used to infer, to some degree, the outline of their conceptual understanding. Such (stereo)types of modelers could be found by identifying communities of modelers that share similar semantic tendencies for given concepts and analyzing whether they have any shared properties that allow us to treat them similarly. A community in this context is nothing more than a group of people who can be seen to share certain things, in this case their understanding of a modeling language and its (meta-)concepts. As language, or any means of communication, is inherently bound to a community using it [Perelman and Olbrechts-Tyteca, 1969] (regardless of whether that community is bound by geography, biology, shared practices and techniques [Wenger and Snyder, 2000, Meyerhoff, 2008], like-minded people [Alani and Shadbolt, 2002], used and shared information [Bishr et al., 1999], cognitive strengths and weaknesses [Wilmont et al., 2012] or simply speech and natural language [Gumperz, 2001, Hoppenbrouwers, 2003]), it seems safe to assume that there are communities which share a typical way of understanding modeling language concepts. This is not to say that such communities would be completely homogeneous in their semantics, but merely that they show enough overlap to be able to be treated as belonging together during a process which integrates models originating from their members without expecting strong inconsistencies in the final product.
Finding such communities based on, for example, empirical data is not a difficult matter in itself. However, the difficulty lies in going from simply finding communities to understanding them and generalizing them, i.e., being able to predict, on the basis of empirical data or prior experience, that communities of people sharing certain properties will typically use certain semantics. To do so it is necessary to find markers – properties that are shared between the members of a community. These markers (e.g., dominant modeling language, focus on specific aspects) are needed to be able to postulate that a given modeler, with a given degree of certainty, belongs to some community and thus likely shares this community’s typical understanding of a concept.

Between 2010 and 2012 several collaborative modeling workshops were organized in the context of the Agile Service Development (ASD) project\(^6\), resulting in [Lankhorst, 2012]. With the partners involved in these workshops, who themselves are involved in different kinds of (collaborative) domain modeling (e.g., enterprise modeling, knowledge engineering, systems analysis), we have found that there are a number of common markers modelers are typically (and often implicitly) grouped by. That is, on the basis of these properties they are often assigned to collaborate on some joint domain modeling task. These properties are, for example, a similar background, education, focus on what aspects to model (e.g., processes, goals), in what sector they do so (e.g., government, health care, telecommunications), and modeling languages used. Thus it seems that in practice, it is assumed that when people have similar backgrounds, use similar modeling languages and methods, etc., they will share a similar enough conceptualization of the involved modeling meta-concepts, and will thus be able to effectively collaborate.

In this article we will test whether the premise (that people with similar backgrounds, using similar languages and methods and so on, will share a similar conceptualization of modeling meta-concepts) of this assumption holds, as it is so rarely tested or backed up by empirical data. To do so we hypothesize that commonly used properties (e.g., modeling language used, modeling focus, operating sector) should be reflected in communities that share a similar semantic understanding of common modeling meta-concepts. To test this we will investigate the personal semantics for practitioners and students alike (whereas other work on finding such communities and their conceptualizations often focuses on analysis of their produced texts or models [Flake et al., 2002, Recker and Dreiling, 2007] instead of the modelers themselves). We will then group them by shared semantics and investigate whether they share the expected, or indeed, any amount of properties. If this is found to be so, then the ‘naive’ grouping procedure commonly used already in practice might have some merit. Furthermore, it could

---

\(^6\) The ASD project (www novay nl/okb projects agile service development 7628) was a collaborative research initiative focused on methods, techniques and tools for the agile development of business services. The ASD project consortium consisted of Be Informed, BiZZdesign, Everest, IBM, O&i, PGGM, RuleManagement Group, Voogd & Voogd, CRP Henri Tudor, Radboud University Nijmegen, University Twente, Utrecht University & Utrecht University of Applied Science, TNO and Novay.
lead to predictive theories that, to a certain degree, predict what (the range of) understanding is that a modeler has for a given concept.

The specific focus of this study is thus to investigate and test whether this common assumption made in modeling practice can be backed up by empirical investigations. In terms of Gregor’s [Gregor, 2006] types of theories in information systems research, we strive to analyze and describe in detail the modelers’ conceptual understandings, and whether that analysis challenges any held assumptions. It is thus out of the scope of this study to propose an approach stipulating how to more effectively ‘do’ the act of enterprise modeling, nor is it our intention to describe in elaborate detail how existing methods (e.g., TOGAF, ADM) might be adapted to fit with our findings. Instead, we will discuss the more fundamental implications our findings have (be they bad or good), and what steps could be taken both by practice and research in order to deal with them.

The rest of this article is structured as follows. In Section 2 we discuss the data used and how we acquired it. In Section 3 we demonstrate how this kind of data can be analyzed to find communities, discuss the difficulties in identifying common properties amongst their members and reflect on the hypothesis and the consequences of our findings. Finally, in Section 4 we conclude and discuss our future work.

2 Methods and Used Data Samples

We use the Semantic Differential [Osgood et al., 1957] as our data gathering method. It is an often-used psychometric method that can be used to investigate what connotative meanings apply to an investigated word or concept, e.g., whether a model is typically considered good or bad, a language is considered intuitive or difficult. It is widely used in information systems and modeling research as it is easy to implement, generally produced valid results [Di Vesta and Dick, 1966], stands up to test-retest validity [Peter, 1979], and there are well-researched guidelines and best practices to ensure quality of the results [Verhagen and Meents, 2007].

In order to investigate the semantic understanding of common modeling meta-concepts with a semantic differential it is necessary to determine what connotations to enquire about for the investigated concepts. The connotations essentially serve as different dimensions on which people’s understanding can be discriminated, e.g., whether an actor could be considered human or not, whether a resource could be immaterial or not. As such, we do not aim to comprehensively measure what a concept ‘is’ for someone, but we focus on characterizing it by determining on which dimensions concepts can be discriminated and delineated. This approach is in line with findings in psychology and cognitive science [Malt et al., 2011, Pinker, 2007] which acknowledge that it is infeasible, if not downright impossible to fully ‘measure’ a concept.

The dimensions we need in order to characterize a concept can be gathered by multiple means, for example exploratory research amongst (a sam-
ple of) the participant population (e.g., the repertory grid elicitation technique [Tan and Hunter, 2002]), observing practitioners to see which topics give rise to discussion involving clarification of terminology more often, and analyzing and comparing specifications of modeling languages and methods to find dimensions on which language constructs differ.

To construct a semantic differential we thus follow the simple steps of determining participants, determining the concepts to be investigated and determining the dimensions on which to investigate them. Once we know which dimensions we wish to investigate, we need to gather a set of adjectives for each of them, as the dimensions are enquired about indirectly (e.g., to find out whether something is considered human or not one could use “human – not-human”, “self-conscious – not self-conscious”). The gathering and validation of adjectives is done amongst (a sample of) the target participants in order to ensure the differential is aimed at them and asks what we want to ask (i.e., is semantically valid). A semantic priming task is finally incorporated as well to ensure that the enquired adjectives are targeted at the concepts we wish to investigate.

2.1 Our study setup

The studies reported on in this article investigate the understanding participants have for the concepts ACTORS, EVENTS, GOALS, PROCESSES, RESOURCES, RESTRICTIONS and RESULTS in the context of (conceptual) modeling. These (meta)concepts were derived from an earlier performed analysis, as reported in [van der Linden et al., 2011]. This analysis was specifically focused on finding the common high-level meta-concepts shared between the specifications of a number of languages and methods covering different aspects used in enterprise modeling (e.g., processes, value exchanges, goals, architecture, performance, security). While there is more difference (of opinion and interpretation) to be found when it comes to domain concepts than the listed meta-concepts, the latter are more interesting to look at for our purposes. As we wish to compare a number of modelers in order to establish whether they can be grouped or not, the concepts we investigate should be shared amongst them. This is definitely the case for the meta-concepts, as they are shared by most languages and methods, whereas the highly specialized domain concepts might not be shared amongst them. Furthermore, as the personal understanding of the meta-concepts directly affect the actual semantics of a model (i.e., a meta-concept’s semantics dictates what is, and is not a permissible instantiation for a part of a model), differences in understanding of these concepts can have more of an (unnoticed) effect on the produced model’s semantics.

The dimensions we investigate for each of these concepts are whether they can be considered natural, human, composed, necessary, material, intentional and vague things. These dimensions originate from the same analysis that was used for the meta-concepts [van der Linden et al., 2011]. They were found by establishing when two similar constructs in a language were used for different purposes. For instance, in case two languages had an ACTOR-like construct, and one language assumed this to be human whereas the other assumed it to be
either an abstract entity (i.e., agents) or a non-human physical entity (i.e., computer hardware), we derived a dimension human on which the languages can be discriminated. The remaining dimensions discriminate meta-concepts based on whether they are found to be naturally occurring or not (i.e., a rock versus a man-made tool), composed of multiple things or singular, necessary to adhere to or not (i.e., an alethic condition versus a deontic ‘rule’), material or non-material (i.e., a physically existing object versus an abstract entity like a number), intentional or unintentional and vague or (well-)defined.

Finally, we use markers to analyze whether groups found in the results of our study reflect commonly used grouping approaches by practitioners. These markers originate from workshop sessions with practitioners and companies as detailed in the introduction. They are the following: what modeling languages and methods people use, what sector they operate in, what the focus of their modeling efforts is, and what kind of stakeholders they interact with during the modeling process.

2.2 Our studies

The practitioner study (n=12, see Table 1) was carried out in two internationally operating companies that focus on supporting clients in (re)designing organizations and enterprises. The investigated practitioners all had several years of experience in applying conceptual modeling techniques. Apart from the semantic differential, we explored what modeling languages and methods they use, what sector(s) they operate in, what they model, and what kind of people they mostly interact with in order to see whether these could be used as identifying factors for semantic communities.

The student study (n=19) is an ongoing longitudinal study into the (evolution of) the understanding computing and information systems science students have of modeling concepts. This study was initiated at the start of the involved students’ academic studies. As such, most of them had little to no experience with modeling languages yet. We explored their educational (and where applicable, professional) background, their knowledge of modeling or programming languages and methods, their interests, and career plans. While these students will likely not offer any particularly interesting insight compared to the practitioners, we include them in order to verify whether the phenomena we investigate occur in other groups than just experienced modelers.

While the amount of participants in each study might seem low compared to other scientific studies with different goals and methodologies, both of our studies are large enough to produce useful results for our purposes. As we will test our hypothesis by attempting to falsify it, we need only counter-examples to the practice of naive grouping we described in the introduction. We are confident that accepting the hypothesis is not unrealistic as it is grounded in empirical observations, and its rejection would also not be a trivial matter. Thus, it is most efficient for a first enquiry into the problem matter to use only as many people as deemed necessary to find a counter-example. Given that the practice
Table 1. Participants in the practitioner study and their relevant data. ‘Proprietary’ languages are not publicly available modeling languages or suites, often developed in-house.

<table>
<thead>
<tr>
<th>No.</th>
<th>Used languages</th>
<th>Sector</th>
<th>Focus</th>
<th>Interacts with</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Proprietary, RDF, Financial, Government, OWL, UML, ERD</td>
<td>Government</td>
<td>Context, domain operations</td>
<td>Senior managers, Domain experts</td>
</tr>
<tr>
<td>2</td>
<td>Proprietary</td>
<td>Government</td>
<td>Knowledge systems, processes</td>
<td>Managers, domain experts</td>
</tr>
<tr>
<td>3</td>
<td>Proprietary</td>
<td>Financial, Government</td>
<td>Knowledge rules, Analysts, modelers</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OWL, BPMN</td>
<td>UML, Government, Public, Healthcare, Finance</td>
<td>Knowledge rules, Domain experts, project managers, IT engineers, business and enterprise architects</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>UML, Proprietary, Financial, Government, Protos</td>
<td>Non-profit</td>
<td>Application-specific knowledge, process and IT departments</td>
<td>Domain experts</td>
</tr>
<tr>
<td>6</td>
<td>Meta-modeling, ontologies, taxonomies, spatial planning</td>
<td>Processes</td>
<td>Domain experts, analysts, architects</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Proprietary, UML, Government, Java</td>
<td>Spatial planning</td>
<td>Business processes, Domain experts, IT process structure, specialists</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>UML, OWL, RDF, Government, Mindmap, Rulespeak, Proprietary</td>
<td>Healthcare</td>
<td>Rules</td>
<td>Business professionals, policymakers, lawyers</td>
</tr>
<tr>
<td>9</td>
<td>Proprietary</td>
<td>Government, Financial</td>
<td>Rules, legislation, Domain experts</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Proprietary, XML, Government, XSLT</td>
<td>Finance</td>
<td>Processes, rules, Domain experts, object definitions</td>
<td>Java developers for systems</td>
</tr>
<tr>
<td>11</td>
<td>ArchiMate, UML, ORM, ERD, BPMN, Amber, ‘improvisational’</td>
<td>Government, Strategic,change, organization process owners</td>
<td>Enterprise-wide architecture, strate- agers, architects, change domain experts, organization process owners</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ArchiMate 2.0, Amber Architect, Proprietary</td>
<td>Healthcare, Financial, Telecom</td>
<td>Business processes, Domain experts</td>
<td></td>
</tr>
</tbody>
</table>

we described seems to be wide-spread, it should thus be found in relatively small samples of participants.
2.3 Data processing & analysis

The resulting numerical data from the two studies were processed into a matrix holding the scores for each concept-dimension combination (e.g., whether an actor is a natural thing or whether a result is a vague thing). These scores range from 2.0 to -2.0, denoting respectively full agreement and disagreement that the dimension ‘fits’ with their understanding of the concept.

To find communities of people that shared a certain amount of semantics (i.e., score similarly for given concept-dimension combinations) we initially analyzed the results using repeated bisection clustering. However, we found it not feasible to investigate the existence and borders of communities with this approach, as it was not sensible to a priori estimate parameters like optimal cluster size and similarity cutoffs (i.e., how similar people should score in order to be considered part of the same community), given that we had no realistic prior data. For this reason we used principal component analysis (PCA) and the visualizations of it in order to generate a more manageable way of investigating the communities and the rough semantic distance between them.

These PCA results and their visualizations (see Figs. 1 and 2) demonstrate (roughly) the degree to which people share a semantically similar understanding of the investigated concepts and can thus be grouped together. It has to be stressed that this ‘unit’ of distance is dimensionless and thus should not be used as an objective measure on its own. Instead, it can be seen and used to distinguish groups from groups, while not saying necessarily in detail how objectively far they are from each other. Combining this data with the information we have gathered about the participants (i.e., the markers) we can investigate whether the structure of the found clusters (i.e., semantic communities) reflect what would be expected from the naive grouping commonly performed in practice.

3 General Results & Discussion

Most importantly, the results support the idea that people can be non-arbitrarily clustered based on their personal semantics. As shown in Figs. 1 and 2 there are easily detectable clusters (i.e., communities) for most of the investigated concepts, although they vary in terms of their member size and the semantic difference between the members (i.e., the variance within the clusters).

While there are both clusters of people that share a semantic understanding for practitioners and students alike, they do differ somewhat. Internal variance for a number of concepts is greater for students, i.e., the semantics are more ‘spread out’ (see Table 2). This may be explained by practitioners having more exposure to specific interpretations of some concepts, causing a lower spread of measurable semantics. Nonetheless, both practitioners and students are still easily divided into communities based on their semantic differences.
3.1 Finding communities

To demonstrate the existence and structure of the found communities, we will discuss some of the clusters we found for the understanding practitioners and students have of GOALS, PROCESSES, RESOURCES and RESTRICTIONS. The immediately obvious difference between the practitioners and students is that, where there are clusters to be found amongst the practitioners, they differ mostly on one axis (i.e., component), whereas the students often differ wildly on both axes. Of particular interest to testing our hypothesis are participants 3 & 8, and 2, 7 &
Fig. 2. Principal components found in the data of concept-specific understandings for students. The visualizations represent (roughly) the distance between understandings which individual participants have. The further away two participants are on both axes (i.e., horizontal and vertically different coordinates), the more different their conceptual understanding has been measured to be.

<table>
<thead>
<tr>
<th>Sample</th>
<th>Actor</th>
<th>Event</th>
<th>Goal</th>
<th>Process</th>
<th>Resource</th>
<th>Restriction</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Practitioner</td>
<td>0.38</td>
<td>0.57</td>
<td>0.68</td>
<td>0.93</td>
<td>0.73</td>
<td>0.94</td>
<td>0.92</td>
</tr>
<tr>
<td>Student</td>
<td>0.66</td>
<td>0.77</td>
<td>1.01</td>
<td>0.93</td>
<td>0.82</td>
<td>0.83</td>
<td>0.81</td>
</tr>
</tbody>
</table>
10 from the practitioner data sample. The first community clusters together very closely for their understanding of RESTRICTIONS (and GOALS, albeit to a lesser degree) while they differ only slightly for most other concepts. This means one would expect them to share some real-world properties. Perhaps they are people specialized in goal modeling, or share a typical way of modeling RESTRICTIONS in a formal sense. The second community (participants 2, 7 & 10) cluster together very closely for RESOURCES, fairly close for GOALS and RESTRICTIONS, while being strongly different when it comes to their understanding of PROCESSES. One could expect this to infer that they have some shared focus on RESOURCES, either through a language they use (e.g., value-exchange or deployment languages) which are often strongly connected to GOALS (as either requiring them, or resulting in their creation). Oppositely, one would not necessarily expect there to be much overlap between the participants with regard to PROCESSES, as they are grouped with a wide spread.

For the students, there are several potentially interesting communities to look at. Participants 4 & 8 differ strongly for several concepts (e.g., their strong differentiation on two components for RESOURCES, and for PROCESSES and RESTRICTIONS), but they have an almost exactly similar understanding of GOALS. One would expect that some kind of property shared between them might be used to identify other participants that cluster together for GOALS, but not necessarily share other understandings. Participants 3, 6 & 19 also cluster together closely for one concept – RESOURCES – but differ on their understanding of the other investigated concepts. As such, if (some) experience in the form of having used specific programming and modeling languages is correlated to their conceptual understanding, one would expect to find some reflection of that in the clusterings of these students.

3.2 Identifying communities

However, when we add the information we have about the participants (see Tables 3 and 4) to these clusters, we run into some problems. It is often the case that communities do not share (many) pertinent properties, or when they do, there are other communities with the same properties that are far removed from them in terms of their conceptual understanding. For instance, consider participants 2, 7 & 10 (highlighted with a gray oval) from the practitioner data sample. While they share some properties, (e.g. operating in the same sector, having some amount of focus on RESOURCES, and interacting with domain experts), when we look at other communities it is not as simple to use this combination of properties to uniquely identify them. For instance, participants 3 & 8 (highlighted with a black rectangle) cluster together closely in their own right, but do share some overlapping properties (both operate in the government sector). Thus, merely looking at the sector a modeler operates in cannot be enough to identify them. Another interesting observation is the fact that while participants 2, 7 & 10 cluster together closely for a number of concepts (e.g., GOALS, RESOURCES and RESTRICTIONS), they do not appear to have a similar understanding of what constitutes a PROCESS, even though they all share a strong focus on modeling
processes. Looking at the combination of sector and focus is not enough either, as under these conditions participant 8 and 10 should be grouped closer together because they both have a focus on rules. When we finally look at the combination of sector, focus and interaction we have a somewhat higher chance of uniquely identifying communities, although there are still counter-examples. Participant 9 (highlighted with a gray rectangle), for example, shares all the properties with participants 2, 7 & 10, but is conceptually far removed from all others. The dataset shows a similar trend for most other participants, providing both examples and counterexamples for most of these property combinations, making it generally very difficult, if not impossible to identify communities.

Table 3. Comparison of some practitioners based on investigated properties. The proprietary language is an in-house language used by one of the involved companies.

<table>
<thead>
<tr>
<th>No.</th>
<th>Used languages</th>
<th>Sector</th>
<th>Focus</th>
<th>Interacts with</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Proprietary</td>
<td>Financial, Government</td>
<td>Knowledge rules, analysts, modelers</td>
<td>processes, data</td>
</tr>
<tr>
<td>8</td>
<td>UML, OWL, RDF, Government, Mindmap, Rules, Healthcare peak, Proprietary</td>
<td>Government</td>
<td>Rules</td>
<td>Business professionals, policymakers, lawyers</td>
</tr>
<tr>
<td>2</td>
<td>Proprietary</td>
<td>Government</td>
<td>Knowledge systems, processes managers, domain experts</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Proprietary, UML, Government, Java</td>
<td>Government, spatial planning</td>
<td>Business processes, Domain experts, IT process structure specialists</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Proprietary, XML, Government, XSLT</td>
<td>Finance</td>
<td>Processes, rules, Domain experts, object definitions java developers for systems</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Proprietary</td>
<td>Financial</td>
<td>Rules, legislation, Domain experts policy, processes</td>
<td></td>
</tr>
</tbody>
</table>

We face the same challenge in the student data sample, although even more pronounced on an almost individual level. There are participants that share the same properties while having wildly varying conceptual understandings. There seems to be some differentiation on whether participants have prior experience, but even then this sole property does not have enough discriminatory power. Take for example participants 4 & 8 (highlighted with a black rectangle) and participants 3, 6 & 19 (highlighted with a gray oval). Both these communities cluster closely together for a specific concept, but then differ on other concepts. One could expect this has to do with a small amount of properties differing between them, which is the case, as there is consistently a participant with some prior experience in programming and scripting languages amongst them. However, if this property really is the differentiating factor, one would expect that on the other concepts the participants with prior experience (4 & 6) would
be further removed from other participants than the ones without experience are, which is simply not the case. It thus seems rather difficult to link these properties to the communities and their structure.

Table 4. Comparison of some students based on investigated properties. Profiles are standardized packages of coursework students took during secondary education, nature being natural sciences, technology a focus on physics and health a focus on biology.

<table>
<thead>
<tr>
<th>No.</th>
<th>Study</th>
<th>Profile</th>
<th>Prior experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Computing Science</td>
<td>Nature, Technology &amp; Health</td>
<td>Some programming and scripting experience</td>
</tr>
<tr>
<td>8</td>
<td>Computing Science</td>
<td>Nature &amp; Technology</td>
<td>None</td>
</tr>
<tr>
<td>3</td>
<td>Information Systems</td>
<td>Nature &amp; Technology</td>
<td>None</td>
</tr>
<tr>
<td>6</td>
<td>Computing Science</td>
<td>Nature &amp; Technology</td>
<td>Programming experience</td>
</tr>
<tr>
<td>19</td>
<td>Information Systems</td>
<td>Nature &amp; Health</td>
<td>None</td>
</tr>
</tbody>
</table>

This challenge could be explained by a number of things. First and foremost would be a plain lack in the amount of properties (or their granularity, as might be the case in the student data sample) to identify communities by, while it is also possible that the investigated concepts were not at the right abstraction level (i.e., either too specific or too vague), or that the investigated concepts were simply not the concepts people use to model. We will discuss each of these possibilities.

The simplest explanation is that the properties we attempt to identify communities by are not the right (i.e., properly discriminating) ones. It is possible (especially for the student data sample) that some of the properties are not necessarily the wrong ones, but that they are not discriminative enough. For example, knowing what modeling languages someone uses could be described in more detail because a language could have multiple versions that are in use, and it is possible (indeed quite likely) that a language used is not the same as the ‘official’ language. However, this line of reasoning is problematic for two reasons. The first being that these are properties that are used by practitioners to (naively) group modelers together, the second that there is no clear-cut way to identify reasonable other properties that are correlated to the modeling practice. If these properties are not useful, we would have to reject the hypothesis on grounds of them being a ‘bad fit’ for grouping people. Other properties that could be thought of could include reflections of the cultural background of modelers. However, these are less likely to be of influence in our specific case as the Enterprise Modelers we investigate are all set in a Western European context and there is little cultural diversity (or granularity, as might be the case in the student data sample) in this sense.

Another explanation could be that the meta-concepts we chose are not at the right abstraction level (i.e., concept width), meaning that they are either
too vague or specific. For example, some modelers could typically think on near-instantiation level while others think more vaguely. If concepts are very specific, one would expect to find differences much faster (as the distance between people's conceptual understanding can be expected to be larger), which thus makes it easier to find communities. If they are (too) vague though, people would not differ much because there are not enough properties to differ on in the first place. However, the way we set up our observations rules out the vagueness possibility, as participants were given a semantic priming task before the semantic differential task of each concept. What we investigated was thus their most typical specific understanding of a concept. For this reason it is unlikely that the abstraction level of the concepts was the cause of the challenge of identifying the communities.

Finally, the most obvious explanation could be a flaw in our preliminary work, namely that we did not select the right concepts, irrespective of their abstraction level. Considering the concepts were derived from an analysis of conceptual modeling languages and methods used for many aspects of enterprises, and that there simply does not seem a way to do without most of them, we find it very unlikely this is the case. The unlikely option that what we investigated was not actually the modeling concept, but something else entirely (i.e., someone considering their favorite Hollywood actors over a conceptual modeling interpretation of actor) can also be ruled out as the priming task in our observation rules out this possibility. It is therefore unlikely that these potential issues affected our analysis, leading us to conclude that the identification of communities of modelers based on the investigated properties is not feasible.

While we had admittedly hoped that these observations would yield a positive result to the hypothesis, the lack of support we have shown means that a theory of predicting how modelers understand the key concepts they use, and thus what the additional 'implicit' semantics of a model could be (as alluded to in the introduction) is likely not feasible. Nonetheless, the observations do help to systematically clarify that these different personal understandings exist, can be measured, and might be correlated to communication and modeling breakdown due to unawareness of linguistic prejudice.

4 Consequences

If we wanted to simply discount the possibility of these properties being good ways to identify communities that share a semantic understanding of some concepts with, we would now be done. But as noted in the introduction, the rejection of this hypothesis carries with it certain consequences, especially as these properties are being used to identify communities and group people together in practice (e.g., the earlier discussed workshops within the ASD project [Lankhorst, 2012]). Our findings are thus of direct relevance to groups like model facilitators and enterprise modelers as they can use these kind of findings to support them in determining good and effective modeling strategies (e.g., by having more of an
insight into the basic ‘kinds’ of modelers, being more aware of common differences). More generally, the consequences of our findings are not simply that we should stop grouping modelers together in a naive fashion, but that we should strive to gain a better understanding of why we do so, what else we might do in its stead, and what avenues of research should be explored to deal with the consequences from such practices.

Our research stresses the point that a ‘model’ should not just be regarded solely in terms of its graphical or textual representation. Instead, we need to understand that the actual model underlying whatever form it is represented in contains more information than the representation itself. This includes, for example, the personal understanding the people have of the concepts and meta-concepts, and in particular, the (joint) understanding of the meta-concepts used by the model. To ensure that one does not leave out these personal understandings and their possible effects during the model creation and use, a number of practices can be applied during the modeling process.

Before actually modeling a domain, whether with modelers or stakeholders, it would be prudent to discuss the understandings the involved parties have of the concepts to be used. This should not be relegated to a purely abstract discussion of the types (meta-concepts) used in the modeling language, but should rather focus on exploring what in the universe of discourse needs to be modeled, and as a result, what types are needed for this. As a consequence, one can focus on elaborating how the people involved understand those meta-concepts. For example, when modeling a specific universe of discourse which entails the necessity to model rules and the way they affect people, it can easily be derived that some meta-concept for rules or restrictions is necessary. We can then move towards a discussion concerning what kind of properties this meta-concept should at least be able to distinguish between, e.g., that some rules are logical conditions that cannot be violated (alethic), while some other ones are moral conditions that can, but ought not be violated (deontic). As a result of having done this, we now know before actually starting to model what the modeling language as such should accommodate.

When it is known what conceptualizations the modeling language to be used should accommodate, we can either select an existing language that does so, or create a domain or purpose-specific language, which could entail either creating a new dialect or a completely new language. In the previous example on the modeling of rules which need to explicitly distinguish between alethic and deontic rules, we can for instance choose to use Object Role Modeling (ORM), as its meta-model explicitly includes alethic and deontic distinctions, and thus accommodates these conceptualizations. Another example is a universe of discourse which includes the need to model goals and their level of attainment. It seems fair to assume that a goal-specific modeling language will eventually be selected, but the exact dialect (e.g., i*, GRL or the TROPOS language) will depend on which dialect allows us to express all the conceptual distinctions we need to express. For example, we can have goals for which the level of attainment is quite well defined, while we also have goals where this same level is more
vague. This necessitates a conceptual distinction, often made in goal modeling dialects by distinguishing between hard and soft goals. However, if no suitable language or dialect can be found, it can sometimes be better to simply create a new one. This can be either a new dialect of an existing language (e.g., subdividing the i* meta-model), or a new domain or purpose-specific language (e.g., by stereotyping UML class diagrams into a new meta-model).

As part of the creation process, it is necessary to constantly validate the model and its understanding. This refers to both the meta-concepts and domain concepts. This can be done, for example, by instantiation testing, where we simply instantiate the model with examples and see whether the model forces us to make explicit the conceptual distinctions we want to be explicit. This validation of instantiation is neither focused on the mathematical validity of the model, nor the correctness of constraints in the model (although both are necessary as well), but on ensuring that the conceptualizations discussed with people beforehand can actually be explicitly expressed.

Apart from being more adaptive to different conceptualizations people have in the modeling process, we can also ensure that our modeling languages are inherently more suited to explicitly deal with them. A possible strategy to deal with this could be to ‘upgrade’ the concept of view as used in e.g., the field of enterprise architecture [The Open Group, 2012] or systems and software engineering [IEEE, 2011]. Traditionally, a view provides a model of a domain from a specific (set of related) concern(s). This could be extended with an articulation of all the expressed (and preferably shared) understandings of the modeling concepts used in the view. Even more, one should consider the joint creation (by a group of stakeholders or modelers) of a view as the joint creation of a model of the domain and the meta-model of the modeling concepts used in that view. This is essentially a form of domain/purpose specific modeling language. When modeling a single domain in terms of a ‘swarm of views’, where each view is modeled by a specific group, from the perspective of a (set of related) concern(s), an integrated or joined model of that domain could then be constructed as a shared (and traceable) understanding among the different views. Such approaches to constructing models by integrating views can be found in e.g., [Dijkman et al., 2008, Brandt and Hermann, 2012]. At first this might sound as a laborious task. However, as our research has indicated, when we do not respect the group-based and personal understanding of modeling concepts and or domain concepts, there is a risk of (implicit) misunderstandings. Such misunderstandings can have severe adverse consequences in an enterprise and information systems engineering context.

However, we should not focus exclusively on attempting to solve the issue by engineering our way around it. Attempts to understand more clearly the reasons and challenges in the modeling process as discussed could be undertaken in, for example, the following areas.

Understanding why people become part of a community (in our case, of shared semantics) could help to deal with their conceptualization processes by understanding more clearly how the group dynamics affect them. Sev-
eral drivers (e.g., economical, political and cultural) could drive people to become part of such communities and have received attention already (see e.g., [Huang et al., 2002] or recent work in Enterprise Architecture [Niemietz, 2013]), and is a worthwhile angle of investigation to extend our understanding of such group dynamics. Furthermore, it would be useful to know how specific domain or purpose-specific modeling languages really need to be, and on the other hand, how general general-purpose modeling languages can be while not conflicting with people’s conceptualizations. This correlates with the (limits of) someone’s semantic flexibility, which can be investigated by testing the limits of their conceptualizations. This likely affects their ability to easily use languages that do not accommodate their typically used and needed distinctions (e.g., a modeler who typically only uses the concept of human actors). This can be investigated, for example, through validation by instantiation testing to see to what degree people can accommodate non-matching uses of their conceptualizations as defined in a language’s specification. Finally, related to all these angles of investigation is the question of what causes the success of certain modeling dialects for certain aspects (process modeling, value exchanges, technical design). While factors behind the drive to create specialized dialects for modeling languages are somewhat understood (“ambiguities, contradictions and incompleteness” [Ayala et al., 2005] of their formal specification), the reason for the success of one dialect over another is less well understood. Combined insight into the formation and evolution of semantic communities and insight into cultural and corporate factors affecting their selection and use might join to explain why certain dialects are used intensively, and others wither away.

To summarize, we have shown that the often implicit assumption that “people have strongly comparable semantics for the common modeling meta-concepts if they share an expertise in certain sectors, modeling focus or used languages” cannot be backed up by our empirical investigation.

5 Conclusion

We have shown a way to discover communities that share semantics of conceptual modeling meta-concepts through analysis of psychometric data and discussed the difficulties in identifying them through shared properties between their members. On basis of this we have rejected the hypothesis that modelers with certain shared properties (such as used languages, background, focus, etc.) can be easily grouped together and expected to share a similar understanding of the common conceptual modeling meta-concepts. Furthermore, we have discussed the consequences of these findings for the modeling process and elaborated on what avenues of research might prove fruitful with these consequences.

Acknowledgments. The authors would like to thank Lex Overmars for his constructive discussions and help in extending this article, and the anonymous reviewers for their insightful comments. This work has been partially sponsored by
the Fonds National de la Recherche Luxembourg (www.fnr.lu), via the PEARL programme.

References


