Towards an Expressivity Benchmark for Mappings based on a Systematic Classification of Heterogeneities[*]

M. Wimmer
TU Vienna
wimmer@big.tuwien.ac.at

G. Kappel
TU Vienna
gerti@big.tuwien.ac.at

A. Kusel
JKU Linz
kusel@bioinf.jku.at

W. Retschitzegger
JKU Linz
werner@bioinf.jku.at

J. Schoenboeck
TU Vienna
schoenboeck@bioinf.jku.at

W. Schwinger
JKU Linz
wieland@jku.at

ABSTRACT

A crucial prerequisite for the success of Model Driven Engineering (MDE) is the seamless exchange of models between different modeling tools demanding for mappings between tool-specific metamodels. Thereby the resolution of heterogeneities between these tool-specific metamodels is a ubiquitous problem representing *the* key challenge. Nevertheless, there is no comprehensive classification of potential heterogeneities available in the domain of MDE. This hinders the specification of a comprehensive benchmark explicating requirements wrt. expressivity of mapping tools, which provide reusable components for resolving these heterogeneities.

Therefore, we propose a feature-based classification of heterogeneities, which accordingly adapts and extends existing classifications. This feature-based classification builds the basis for a mapping benchmark, thereby providing a comprehensive set of requirements concerning expressivity of dedicated mapping tools. In this paper a first set of benchmark examples is presented by means of metamodels and conforming models acting as an evaluation suite for mapping tools.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability

General Terms

Measurement

Keywords

Classification of Heterogeneities, Mapping Benchmark

1. INTRODUCTION

With the rise of MDE models become the main artifacts of the software development process [3]. Hence, a multitude of modeling tools is available supporting different tasks, such as model creation, model simulation, model checking, model transformation, and code generation. Seamless exchange of models among different modeling tools increasingly becomes a crucial prerequisite for effective MDE. Due to the lack of interoperability, however, it is often difficult to use tools in combination, thus the potential of MDE cannot be fully exploited. For achieving interoperability in terms of transparent model exchange, current best practices comprise creating model transformations between different tool metamodels (MMs) with the main drawback of having to deal with all the intricacies of a certain transformation language. In contrast to that, first mapping tools [6, 18] have been proposed, allowing to specify a transformation on a more abstract level by means of reusable components. Out of the resulting mapping definitions corresponding executable transformation code can be generated. In the definition of a mapping between MMs the resolution of heterogeneities represents *the* key challenge. Thereby heterogeneities result from the fact that semantically similar metamodeling concepts (M2) can be defined with different meta-metamodeling concepts (M3) leading to differently structured metamodels. As a simple example Fig. 1 shows two metamodels of fictitious[1] domain-specific tools administrating publications. Whereas the MM of `Tool1` models the type of a publication by the attribute `Publication.kind` (e.g., conference, workshop or journal), the MM of `Tool2` represents the same semantic using the class `Publication` which refers to a class `Kind` to determine the kind of the publication.

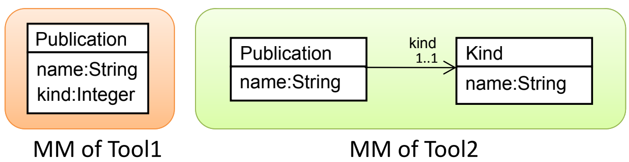

Figure 1: Two Heterogeneous Tool Metamodels

In order to resolve such heterogeneities mapping tools provide certain reusable components. Nevertheless, it is still unclear, which kinds of reusable components are required to provide the necessary expressivity. Therefore this paper provides a systematic classification of heterogeneities occurring in the domain of MDE between object-oriented MMs,

[*]This work has been funded by the Austrian Science Fund (FWF) under grant P21374-N13.

[1]Due to reasons of comprehensibility examples comprising ontological concepts have been preferred over examples comprising linguistic concepts.

thereby adapting and extending existing classifications [2, 4, 10, 11, 12, 13, 15, 17]. Moreover, this classification is used to derive an evaluation suite building an expressivity benchmark for mapping tools. Thereby a first set of examples is presented in this paper. Additional heterogeneity examples can be downloaded from our homepage[2] complementing the expressivity benchmark.

The remainder of this paper is structured as follows. In Section 2 we present the design rationale behind our classification as well as the feature-based classification itself. In the Sections 3-5 we exemplary discuss heterogeneities, thereby presenting six examples of our expressivity benchmark. Related work is discussed in Section 6 and finally, Section 7 concludes the paper together with an outlook on future work.

2. TOWARDS A SYSTEMATIC CLASSIFICATION OF HETEROGENEITIES

This section presents the design rationale behind the proposed classification of heterogeneities as well as the classification itself. Since the classification targets at the domain of MDE, it bases on object-oriented MMs in contrast to existing classifications from the domain of data engineering basing either on the relational or the XML data model. To clearly make explicit the interconnections between heterogeneities we build our classification on a feature model [5].

2.1 Deriving Heterogeneities from Ecore

Heterogeneities result from the fact that semantically similar concepts can be defined with different metamodeling concepts (e.g., Ecore[3]) leading to differently structured tool metamodels. To exemplify this, Fig. 2 depicts the MMs of Fig. 1 as Ecore instances. Thereby, several heterogeneities arise, e.g., the MM of `Tool1` represents the *publication kind* by an `EAttribute` whereas the MM of `Tool2` utilizes an `EReference`, an `EClass` and an `EAttribute` to represent the semantically equivalent information.

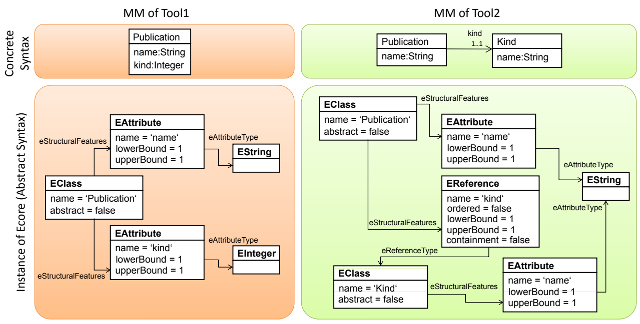

Figure 2: Tool Metamodels as Instances of Ecore

To gain a systematic classification of different kinds of *syntactic heterogeneities*, we investigated potential variation points between two Ecore-based metamodels (cf. Fig. 3). Ecore has been used since it is the prevalent meta-metamodel in MDE and since it comprises the core concepts of semantic data models [9], being *classes*, *attributes*, *references* and *inheritance*. Therefore, the proposed classification can also be applied to other data models comprising these common core concepts, e.g., OWL[4].

[2]www.modeltransformation.net
[3]http://www.eclipse.org/modeling/emf/
[4]http://www.w3.org/TR/owl-features/

In this respect, Fig. 3 depicts the relevant extract of the Ecore meta-metamodel for mappings. When comparing two Ecore-based metamodels, different cases can be distinguished, namely (i) that in the left-hand side (LHS) MM and in the right-hand side (RHS) MM the *same* Ecore concept is used. Thereby differences wrt. the owned attribute settings can arise, e.g., if two `EClasses` are used, one can be set `abstract` whereas the other is not – leading to a *concreteness difference*. Moreover, (ii) in the LHS MM and in the RHS MM *different* Ecore concepts may be used, e.g., an `EAttribute` in the LHS MM and an `EReference`, an `EClass` and an `EAttribute` in the RHS MM (cf. example in Fig. 2). Finally, (iii) both cases mentioned get more complex, if the number of Ecore concepts for modeling a certain MM concept differs. A simple example in this respect is that in one MM two `EAttributes` `firstName` and `lastName` are used whereas in the other MM this information is contained in just one `EAttribute` `name`.

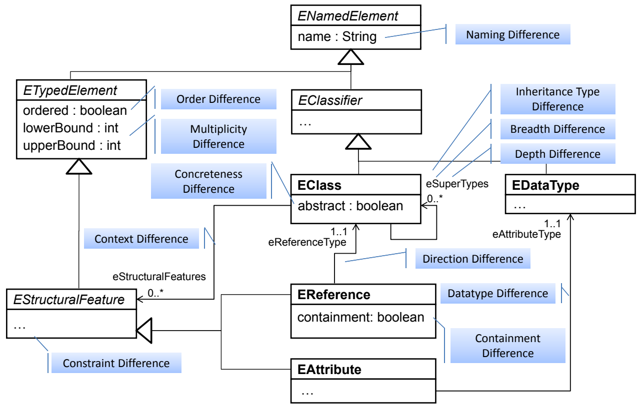

Figure 3: Variation Points in Ecore-based MMs

Besides *syntactic heterogeneities*, comprising all heterogeneities that can be derived from the syntactic definition in Ecore, also *semantic heterogeneities* may arise [15]. They occur when the valid instance set differs – either (i) in the number of valid instances or (ii) in the interpretation of the instance values. An example for the first case is that one MM comprises an `EClass Publication` whereas the other MM comprises an `EClass JournalPublication`, allowing only for journal instances - thus being a subset of the valid instances of the `EClass Publication`. An example for the second case is that one MM comprises an `EAttribute amount` encoding pricing information in Dollar, whereas the other MM also exhibits an `EAttribute amount` but encoding the pricing information in Euro. Thus, semantic heterogeneities can not be derived from the syntax (since in both cases the MMs can be represented syntactically equal) but only by incorporating *interpretation*, i.e., an assignment of a meaning to each piece of data [8].

2.2 Classification of Heterogeneities

Based on this design rationale, we introduce a classification of heterogeneities. It is expressed using the feature model formalism [5], which allows to clearly point out the interconnections between the different kinds of heterogeneities (e.g., `xor` features modeling mutual exclusive features versus `or` features allowing to pick several features at once). Thereby heterogeneities are divided into the two main classes of (i) *semantic heterogeneities*, i.e., heterogeneities wrt. *what*

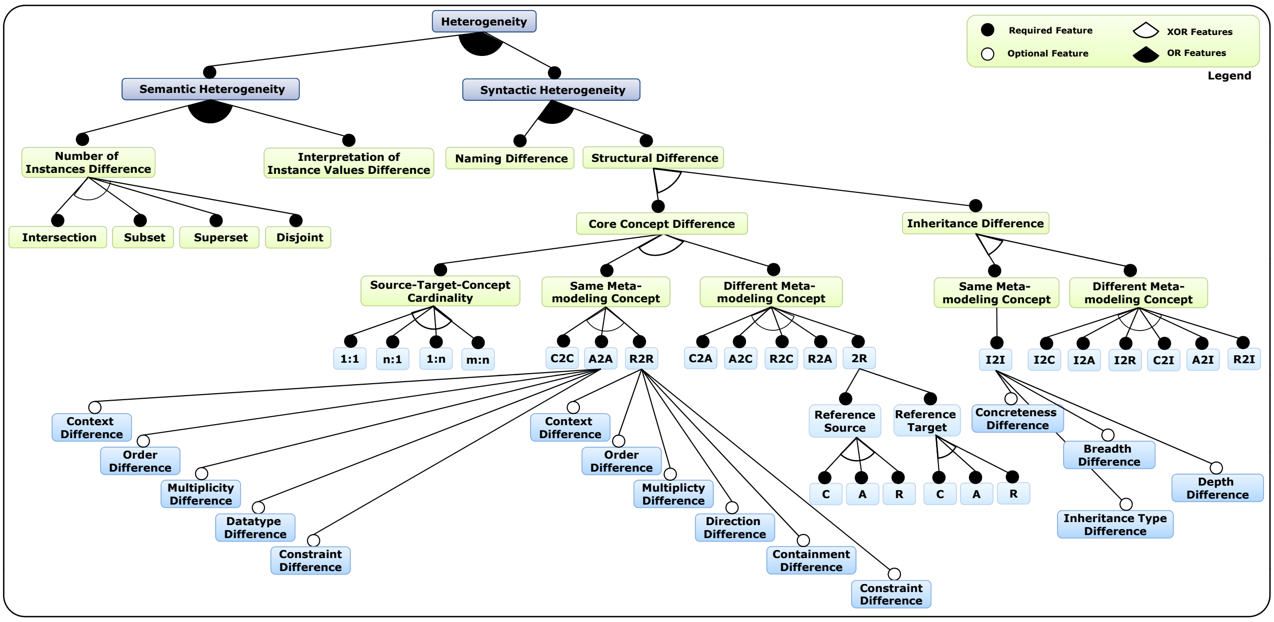

Figure 4: Heterogeneity Feature Model

is represented by a MM and (ii) *syntactic heterogeneities*, i.e., heterogeneities wrt. *how* it is represented (cf. Fig. 4) whereby these two classes might occur jointly as modeled by the `or` relationship in between.

Semantic Heterogeneities. Concerning *semantic heterogeneities* – as mentioned above – two main cases can be distinguished namely (i) differences in the *number of valid instances* and (ii) differences in the *interpretation of the instance values*. With respect to the first case all the set-theoretic relationships might occur as modeled by the corresponding sub-features. Regarding the second case diverse modifications of the values might be necessary to translate the values of one MM to correct values of the other MM such that it conforms to the *interpretation* of the other MM.

Syntactic Heterogeneities. With respect to *syntactic heterogeneities* we distinguish between simple *naming differences* (i.e., a difference in the value of the `name` attribute of `ENamedElement` – cf. Fig. 3) and more challenging *structural differences*. Although names play an important role when deriving the semantics of a certain concept, names do not allow to automatically conclude on the semantics. Thereby, the two cases (i) same semantic and different naming, i.e., *synonyms* and (ii) different semantic and same naming, i.e., *homonyms* can be distinguished.

With respect to structural differences again two main cases can be distinguished – namely *core concept differences* and *inheritance differences*. Thereby, core concept differences are differences that occur due to the different usage of classes, attributes and references between two MMs. In addition, these two main categories can be further distinguished into *same metamodeling concept* heterogeneities and *different metamodeling concept* heterogeneities, differentiating whether the same Ecore concepts have been used in the LHS MM and in the RHS MM or not. In the context of core concept differences additionally a different number of concepts may have been used in the two MMs leading to different *source-target-concept cardinalities*. In the following sections a first set of benchmark examples is given divided into three main packages, comprising (i) core concept heterogeneities with same metamodeling concept heterogeneities, (ii) core concept heterogeneities with different metamodeling concept heterogeneities and (iii) inheritance heterogeneities. Due to space limitations only a subset of all potential heterogeneities is explained in detail by means of concrete metamodels and according model instances but nevertheless examples from each main category are given. In this respect, the benchmark examples are described uniformly comprising (i) a short description, (ii) the main challenges, (iii) the example description, and (iv) a discussion of resolution strategies. Complementary benchmark examples are presented on our collaborative homepage which invites the community to participate in adding and discussing benchmark examples.

3. CORE CONCEPT HETEROGENEITIES – SAME CONCEPTS

Same metamodeling concept heterogeneities are heterogeneities, that occur although the same modeling concept has been used in the LHS MM as well as in the RHS MM as mentioned above. In this respect, two main differences might emerge – either the concepts exhibit different attribute settings (cf. Fig 3) or a different number of concepts has been used in the MMs to express the same semantic concept (cf. *Source-Target-Concept Cardinality* in Fig. 4). In the following two examples of this category are given.

3.1 Benchmark Example 1

This first example (cf. Fig. 5) only exhibits differences wrt. different attribute settings (cf. optional features of ***A****(ttribute)***2A***(ttribute)* and ***R****(eference)***2R***(eference)* in Figure 4) as well as semantic heterogeneities. The main challenges in this example can be summarized as follows:

1. `EAttribute Professor.dateOfBirth` – `EAttribute Prof.bornIn`:
   A2A, Multiplicity Difference, Datatype Difference

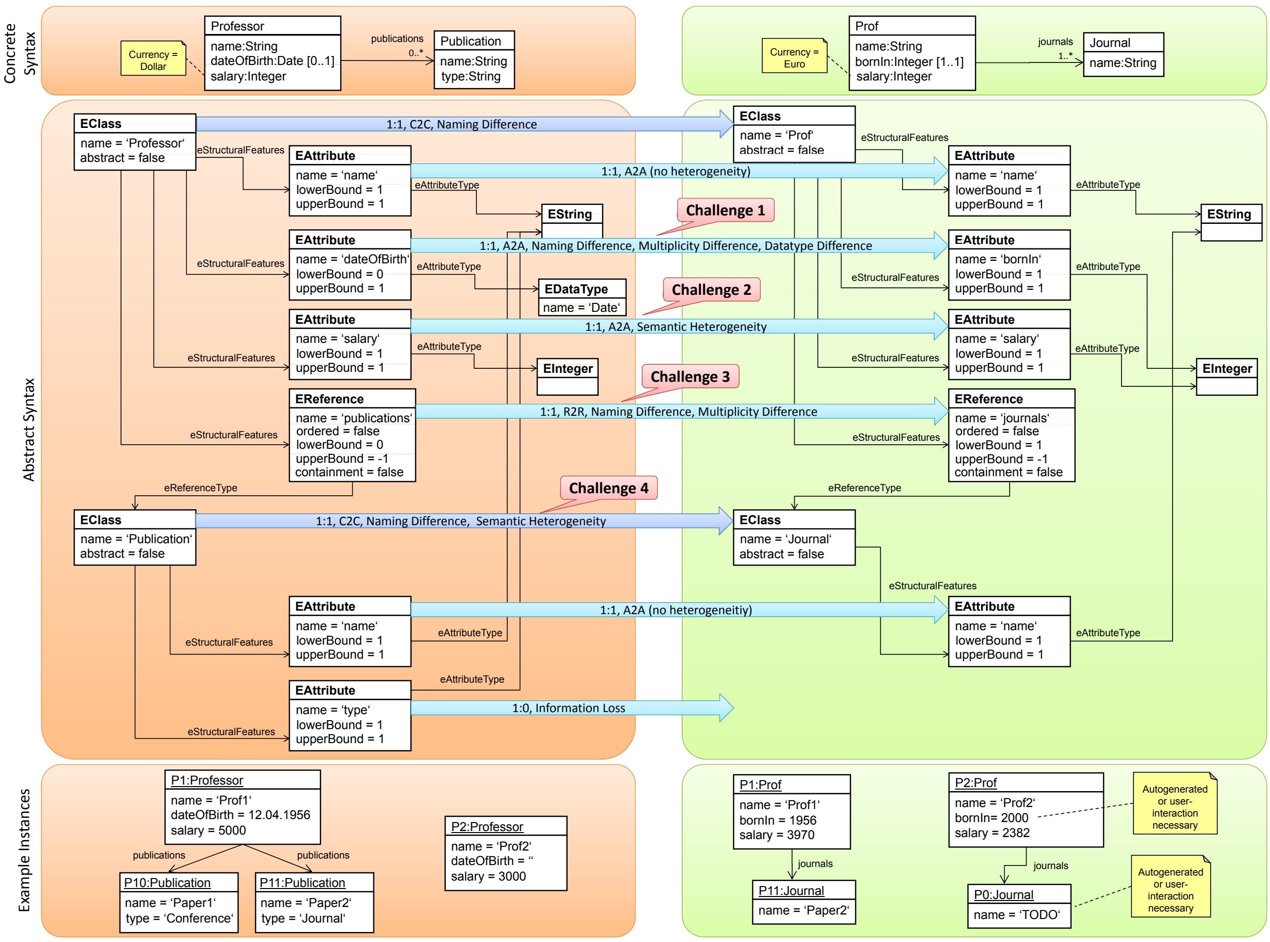

Figure 5: Benchmark Example 1 – Same Metamodeling Concept Heterogeneities

2. `EAttribute Professor.salary` – `EAttribute Prof.salary`: *Semantic Heterogeneity* (Interpretation of Instance Values Difference), *A2A*

3. `EReference Professor.publications` – `EReference Prof.journals`: *R2R*, *Multiplicity Difference*

4. `EClass Publication` – `EClass Journal`: *Semantic Heterogeneity* (Number of Instances Difference), *C2C*

Example Description. This first benchmark example (cf. Fig. 5) exhibits four main challenges. With respect to the *first challenge*, a *multiplicity difference* as well as a *datatype difference* between the `EAttributes Professor.dateOfBirth` and `Prof.bornIn` arise. Concerning the *second challenge* a *semantic heterogeneity* between the `EAttributes Professor.salary` and `Prof.salary` emerges since `Professor.salary` is encoded in Dollars whereas `Prof.salary` is encoded in Euros, i.e., difference in the interpretation of the values. Regarding the *third challenge* a multiplicity difference between the `EReferences Professor.publications` and `Prof.journals` exists. Finally, the *fourth challenge* incorporates again a *semantic heterogeneity* – but this time a difference in the number of valid instances. For resolving the differences of the first three challenges corresponding *functions* are required which either are able to generate values or to transform values. In contrast to that, for resolving the heterogeneity of the fourth challenge a corresponding *condition* is needed, that filters those instances, that are still valid in the context of the RHS `EClass`.

Discussion of Resolution Strategies. When taking a look at the example instances, one can see that a resolution strategy has been chosen to *minimize information loss* and to *achieve valid instances* only. This is since instance `P2` has been kept in the RHS although it does not reference any journal publication in the LHS model. Another potential resolution strategy would be to keep only those `Professor` instances that actually exhibit a journal publication. If this is the case, also a semantic heterogeneity between the `EClasses Professor` and `Prof` would exist, since the valid instance set would be potentially different. Another interesting point in this example is that the RHS MM is more restrictive than the LHS MM since the `EAttribute Prof.bornIn` always requires a value and since each instance of `Prof` requires at least one link to a journal publication. Since these restrictions do not exist in the LHS MM, instances of the LHS MM may not fulfill them. Therefore some resolution strategy is needed – either by auto-generating values or by incorporating user-interaction in order to produce valid instances of the RHS MM.

3.2 Benchmark Example 2

In contrast to the first example which restricts itself to source-target-concept cardinalities of 1:1, this example (cf.

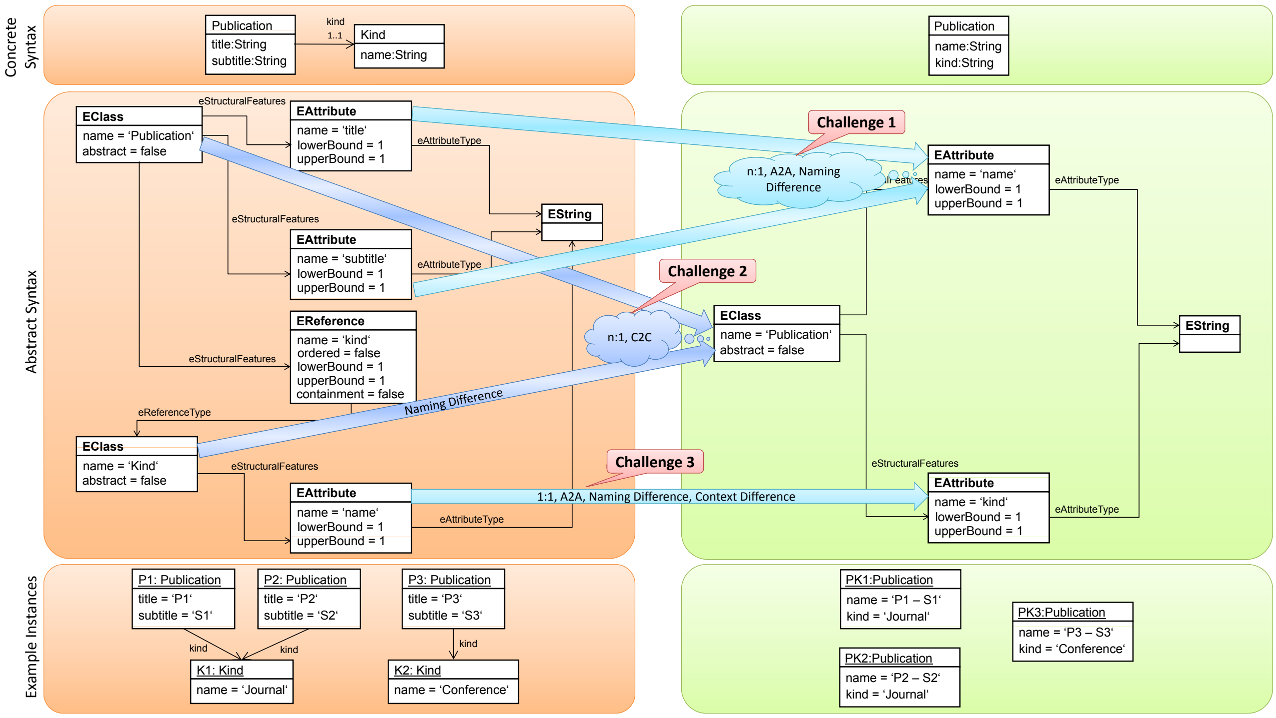

Figure 6: Benchmark Example 2 – Same Metamodeling Concept Heterogeneities

Fig. 6) additionally contains differences wrt. the number of concepts (cf. *Source-Target-Concept Cardinality* in Fig. 4). The main challenges in this example can be summarized as follows:

1. `EAttribute Publication.title`, `EAttribute Publication.subtitle` – `EAttribute Publication.name`: *Source-Target-Concept Cardinality:* n:1, *A2A*

2. `EClass Publication`, `EClass Kind` – `EClass Publication`: *Source-Target-Concept Cardinality:* n:1, *C2C*

3. `EAttribute Kind.name` – `EAttribute Publication.kind`: *A2A*, *Context Difference*

Example Description. This benchmark example (cf. Fig. 6) possesses three challenges. Concerning the *first challenge*, there is a n:1 *source-target-concept cardinality* between the **`EAttributes title, subtitle`** and **`name`**. In order to resolve this heterogeneity, merging functionality is needed, which is basically a *concatenation function* in this case. Concerning the *second challenge*, again a n:1 *source-target-concept cardinality* exists, but this time between the `EClasses Publication, Kind` and `Publication`. Therefore, again *merging functionality* is needed, allowing to merge objects under a certain condition. Finally, the *third challenge* consists in a *context difference* between the `EAttributes Kind.name` and `Publication.kind`. For its resolution the *assignment of values across object boundaries* is needed.

Discussion of Resolution Strategies. When taking a look at the example instances in Fig. 6, one can see, that for each combination of a `Publication` object and the referenced `Kind` object a `Publication` object should be generated. Concerning the merge of the attributes different strategies could be followed, whereby in this case a simple concatenation has been chosen. Other strategies comprise another concatenation order. In case of other datatypes (e.g., numbers) arbitrary calculations could be incorporated.

4. CORE CONCEPT HETEROGENEITIES – DIFFERENT CONCEPTS

Different metamodeling concept heterogeneities result from expressing the same semantic concept with different modeling concepts in the LHS MM and in the RHS MM. In our classification, potential heterogeneities were derived by systematically combining the identified core concepts of semantic data models. To exemplify these heterogeneities two benchmark examples are discussed in the following.

4.1 Benchmark Example 3

The third example (cf. Fig. 7) deals with the fact that a concept is modeled in the LHS MM by means of an `EAttribute` whereas the RHS MM models this concept explicitly by means of an `EClass`. Thus, the main challenges in this example can be summarized as follows:

1. `EAttribute Publication.kind` – `EClass Kind`: *A2C*

2. `EClass Publication`, `EAttribute Publication.kind` – `EReference Publication.kind`: *CA2R*

Example Description. The *first challenge* is that the kind of the publication is represented by means of the `EAttribute Publication.kind` in the LHS MM whereas the RHS MM makes the type explicit by means of the `EClass Kind`, which is therefore classified as ***A****(ttribute)***2C***(lass)* in Fig. 7. In order to link publications with the publication kind, the RHS MM provides the `EReference Publication.kind` for which there is no according counterpart in

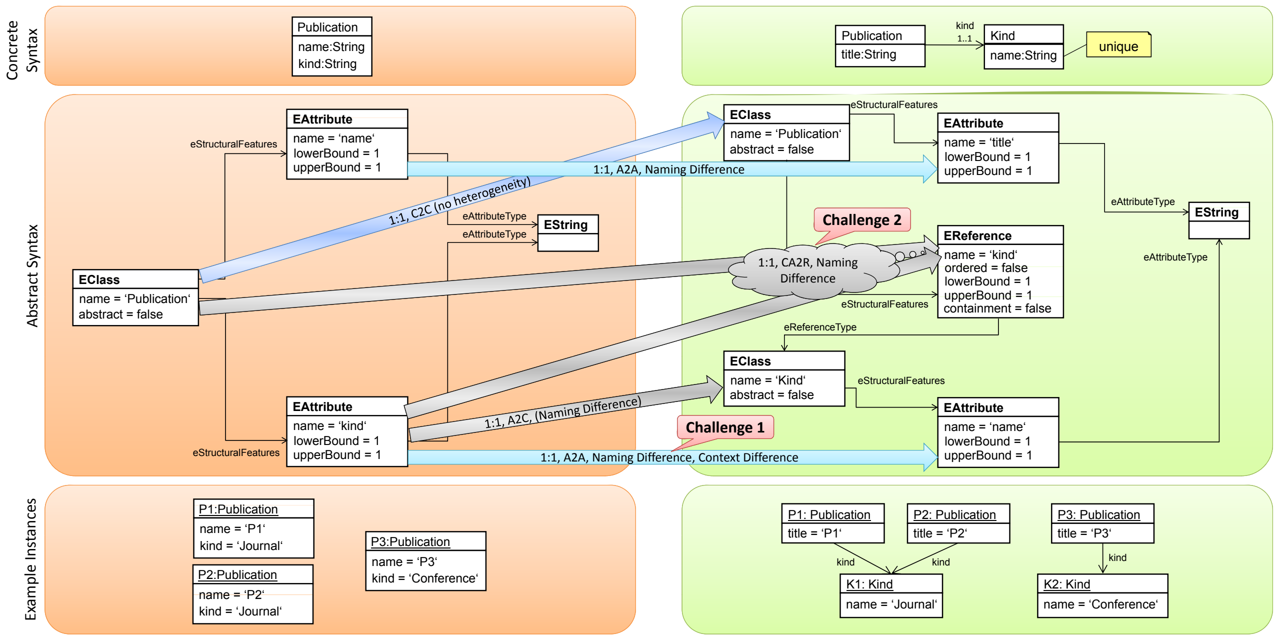

Figure 7: Benchmark Example 3 – Different Metamodeling Concept Heterogeneities (A2C, CA2R)

the LHS MM, i.e., the RHS links have to be generated, representing the *second challenge* in the example. In order to establish such additional links in the RHS, the information is needed in which relation the to be linked concepts have been in the LHS MM. With respect to this example, the source of the `EReference Publication.kind` is represented in the LHS MM by means of the `EClass Publication` and the target of the `EReference` by means of the `EAttribute Publication.kind`. Therefore, this heterogeneity is classified as ***C****(lass)****A****(ttribute)****2R****(eference)*, whereby the first letter depicts the used LHS concept for the source of the to be generated reference and the second letter the used LHS concept for the target of the to be generated reference.

Discussion of Resolution Strategies. When taking a look at the example instances, one can see that the desired intention of an *A2C* heterogeneity is that only for distinct `Publication.kind` attribute values an according `Kind` object should be generated. Therefore, the RHS model exhibits only a single object named `Journal` (cf. `K1` in Fig. 7), which is referenced by the `Publication` objects `P1` and `P2`.

4.2 Benchmark Example 4

Whereas the previous example exhibited the heterogeneity that a LHS concept is modeled by means of an `EAttribute` and the RHS concept by means of an `EClass`, the following example (cf. Fig. 8) exhibits the heterogeneity that a LHS concept is modeled by means of an `EReference` whereas the equivalent RHS concept is again represented by an `EClass`. The main challenges in this example are:

1. `EReference Professor.publications` – `EClass DBLPEntry`: *R2C*

2. `EReference Professor.publications` – `EAttribute DBLPEntry.id`: *R2A*

3. `EClass Professor`, `EReference Professor.publications` – `EReference Professor.entries`: *CR2R*

4. `EReference Professor.publications`, `EClass Publication` – `EReference DBLPEntry.publication`: *RC2R*

Example Description. Whereas the class `Professor` in the LHS MM in Fig. 8 has a direct `EReference Professor.publications`, the LHS MM offers this information only indirectly by means of the `EClass DBLPEntry` and its `EReference DBLPEntry.publication`, representing the *first challenge* in this example (cf. ***R****(eference)****2C****(lass)* feature value in Fig. 4). Concerning the *second challenge*, values for the `DBLP.id EAttribute` have to be generated. Since the containing RHS `EClass` is generated on basis of the LHS `EReference Professor.publications` the according `EAttribute` has also to be generated on basis of this `EReference` (cf. ***R****(eference)****2A****(ttribute)* feature value in Fig. 4). With respect to the third and fourth challenge, the according links have to be established. For this again the information is needed in which relation the to be linked concepts have been in the LHS MM, as described above. Concerning the `Professor.entries EReference`, the source of the `EReference` (`Professor`) is generated on basis of the LHS `EClass Professor` and the target of the `EReference` (`DBLPEntry`) on basis of the `EReference Professor.publications` – thus this heterogeneity is classified as ***C****(lass)****R****(eference)****2R****(eference)*. A similar situation occurs for the RHS `EReference DBLPEntry.publication` but in this case the source of the `EReference` bases on an `EReference` and the target bases on an `EClass` – a heterogeneity classified as ***R****(eference)****C****(lass)****2R****(eference)*.

Discussion of Resolution Strategies. The challenge in this benchmark example is to obtain objects conforming to the RHS `EClass DLBPEntry` (cf. example instances in Fig. 8). These RHS objects have to created on basis of the LHS links since these links encode the information which publications belong to which professor which is also the task of `DBLPEntry` objects. Therefore, Fig. 8 depicts four `DLBPEntry` objects which originate from the four LHS `Professor.publications` links. To set the `DBLPEntry.id`

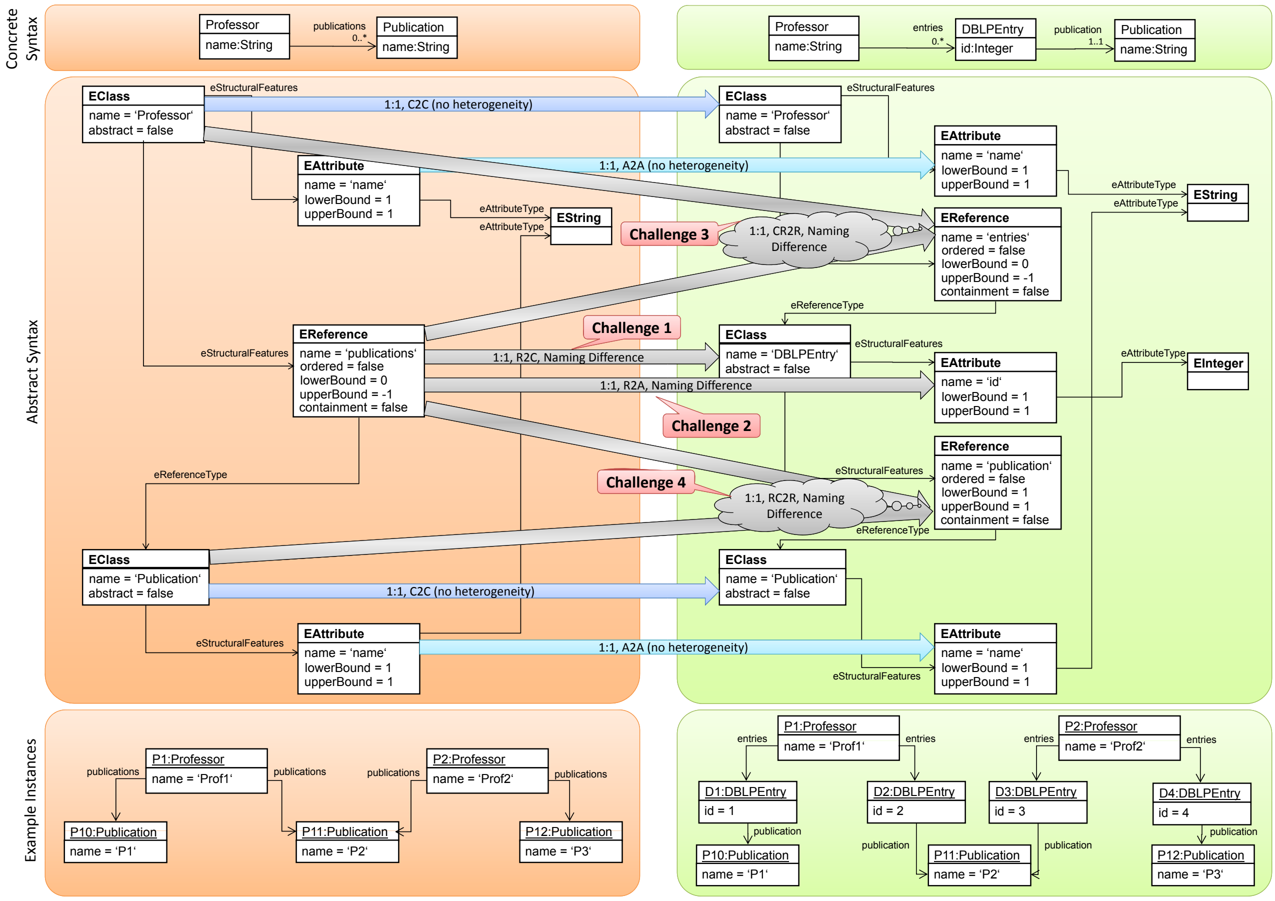

Figure 8: Benchmark Example 4 – Different Metamodeling Concept Heterogeneities (R2C, R2A)

value a function is needed which generates an according id whereby again for every LHS link an according RHS value should be created.

5. INHERITANCE HETEROGENEITIES

In the previous sections we discussed potential heterogeneities when considering the metamodeling concepts of classes, attributes and references. Finally, heterogeneities might be caused by the concept of inheritance. In this respect we again divide into heterogeneities that might occur although both MMs use inheritance (cf. *same metamodeling concept inheritance differences* in Fig. 4) and heterogeneities that occur if only one MM makes use of inheritance (cf. *different metamodeling concept inheritance differences* in Fig. 4). Similar to the afore mentioned same metamodeling concept differences (cf. Section 3), same metametamodeling concept inheritance differences occur due to different attribute values or links in the Ecore MMs (cf. Fig. 3) whereas the latter heterogeneities occur if an inheritance hierarchy in one MM is expressed by other concepts (i.e., classes, attributes, and references) in the other MM. In the following one example per category is given.

5.1 Benchmark Example 5

This example (cf. Fig. 9) belongs to the *same metametamodeling concept category* and therefore both MMs make use of inheritance. Nevertheless certain heterogeneities occur, comprising *breadth differences*, *depth differences* and *concreteness differences*. The main challenges in this example can be summarized as follows:

1. `EClass FullProf`, `EClass AssistantProf` – `EClass FullProf`: *I2I*, *Breadth Difference*
2. `EClass Assistant` – `EClass Assistant`: *I2I*, *Concreteness Difference*, *Depth Difference*
3. `EClass PrePhd`, `EClass PostPhd` – No corresponding `EClass`: *I2I*, *Breadth Difference*

Example Description. Concerning the *first challenge*, a *breadth difference* between the LHS `EClasses FullProf`, `AssistantProf` and the RHS `EClass FullProf` exists. This is since the number of sibling classes in the context of a certain parent class differs. For resolving breadth differences, the strategy can be applied to map instances of some class only existing in the LHS MM to a *concrete* parent class in the RHS MM. Nevertheless, since the parent classes of the `EClass AssistantProf` are abstract, instances of `AssistantProf` get lost. With respect to the *second challenge*, a *concreteness difference* as well as a *depth difference* occurs between the two `EClasses Assistant`. This is since the `EClass Assistant` in the LHS MM is set abstract whereas the corresponding `EClass Assistant` in the RHS MM is concrete. Additionally, a *depth difference* exists, since the longest path of subclasses in the context of the `EClass Assistant` in the LHS MM is 1 whereas it is 0 in the context of the corresponding class in the RHS MM. For resolving the

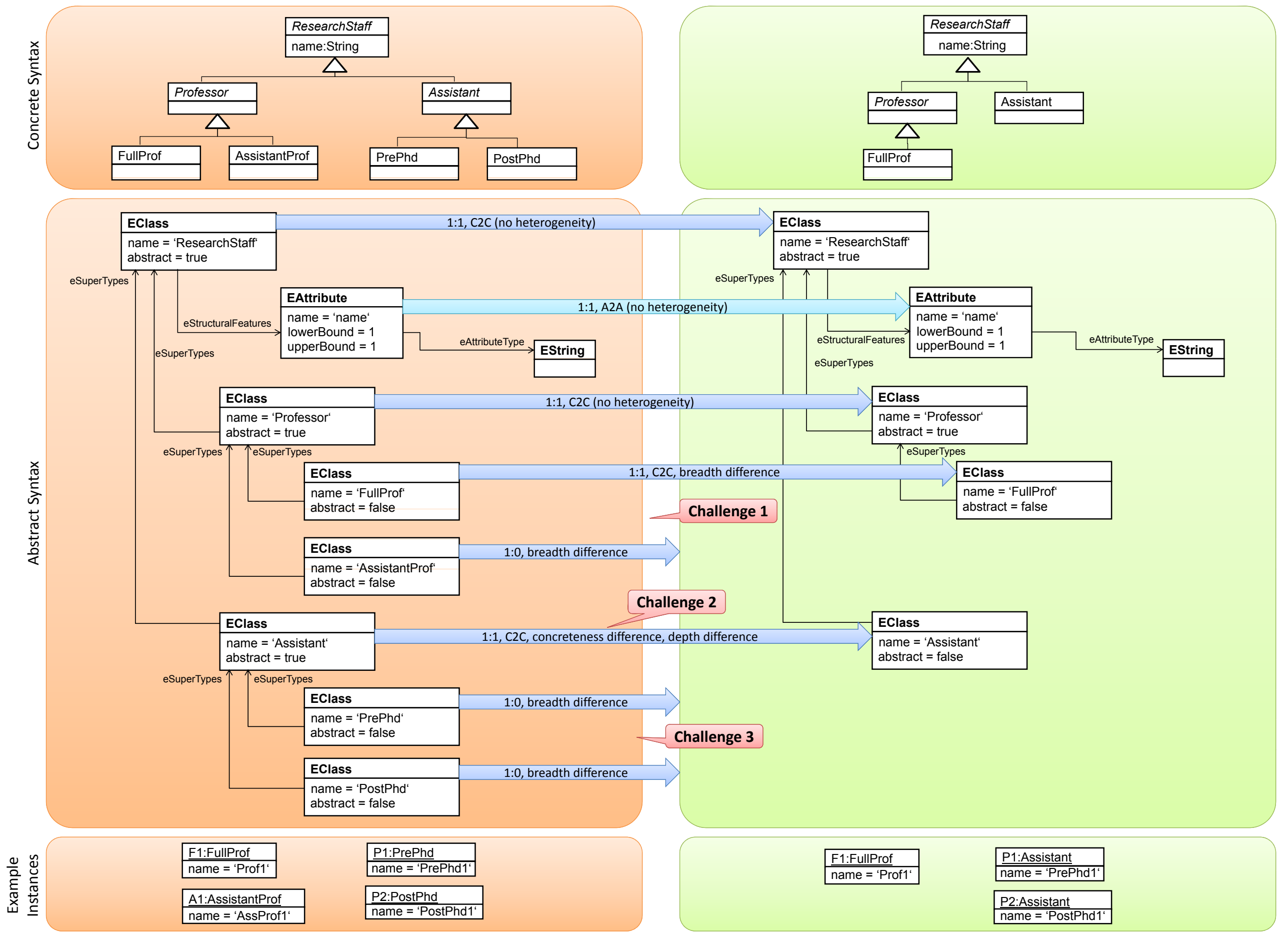

Figure 9: Benchmark Example 5 – Same Metamodeling Concept Heterogeneities

concreteness difference no strategy is needed in this example, since the LHS class is abstract and therefore no instances can exist. The situation would be different, if it would be inverse. Then instances might be lost, if no concrete class in the RHS MM for including those instances might be found. For resolving the depth difference, the strategy can be pursued to map instances of the classes only existing in the LHS MM to some concrete parent class in the RHS MM. Therefore, in this case the instances of the `EClasses PrePhd` and `PostPhd` result in instances of the parent `EClass Assistant` in the RHS MM. Finally, regarding the *third challenge*, a *breadth difference* between the `EClasses PrePhd` and `PostPhd` and the non-existing RHS classes exists. Since in this case the breadth difference overlaps with the depth difference of challenge 2 (being the case since the `EClass Assistant` in the RHS MM exhibits no subclasses at all), no additional resolution strategy is needed here.

Discussion of Resolution Strategies. When taking a look at the chosen resolution strategies, one can see that a strategy has been chosen that tries to minimize instance loss and thus information loss. Therefore instances of a class that only exist in the LHS MM should be kept by mapping them to some concrete parent class due to the is-a relationship between the classes. Nevertheless, the explicit type information and additional features only owned by the subclass are lost. Therefore sometimes also a strategy that omits these instances might be useful.

5.2 Benchmark Example 6

This example (cf. Fig. 10) belongs to the *different metamodeling concept category* and therefore only one MM makes use of inheritance. The main challenge in this example can be summarized as follows:

1. `EAttribute ResearchStaff.kind` – `EClasses ResearchStaff, Professor, Assistant` and `FullProf` in inheritance hierarchy: *A2I*

Example Description. With respect to the *main challenge* in this example, an ***A****(ttribute)****2I****(nheritance)* heterogeneity between the `EAttribute ResearchStaff.kind` and the `EClasses ResearchStaff`, `Professor`, `Assistant` and `FullProf` occurs. For resolving this kind of heterogeneity a condition is needed to divide the instances of the `EClass ResearchStaff` according to the values of the `EAttribute kind` in order to instantiate instances of the corresponding RHS classes. Thereby the problem may arise, that the `EAttribute` of the LHS MM comprises values that do not correspond to any (concrete) `EClass` in the RHS MM. This is the case in the example with the instance `R1`, since the corresponding `EClass Professor` in the RHS MM is abstract and can thus not be instantiated causing information loss.

Discussion of Resolution Strategies. Concerning the resolution strategy chosen in this example again information loss should be prevented whenever possible. Nevertheless, as already discussed above, this may not always be possible.

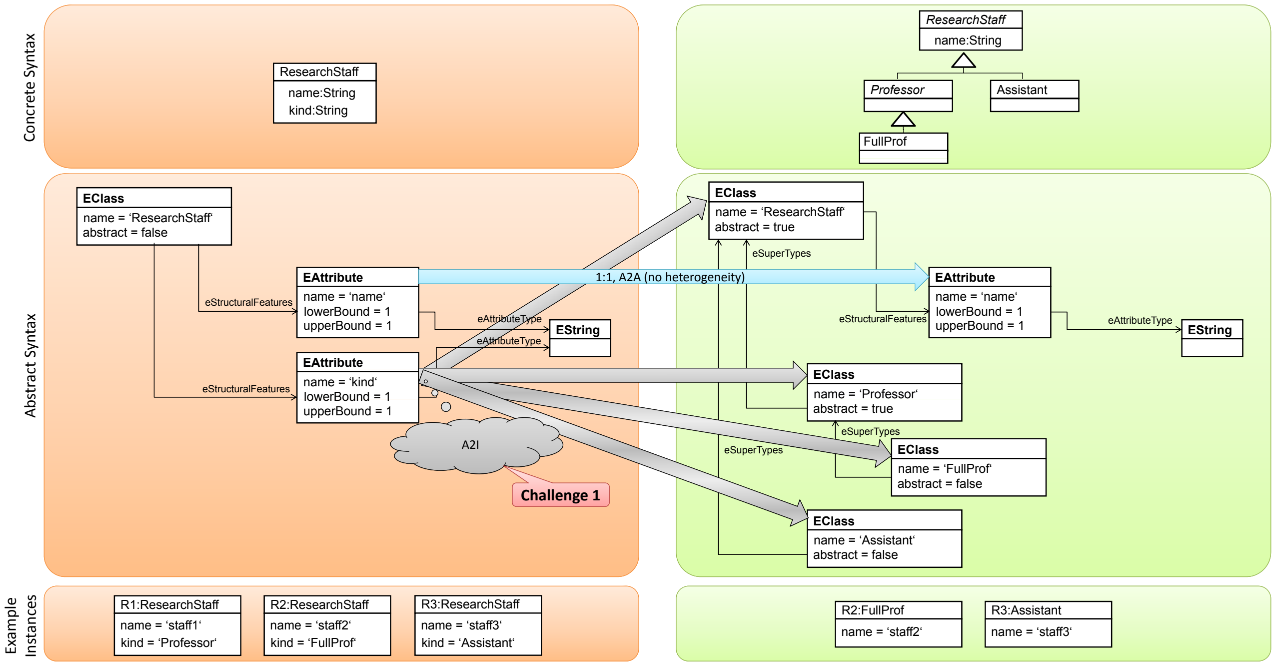

Figure 10: Benchmark Example 6 – Different Metamodeling Concept Heterogeneities (A2I)

6. RELATED WORK

In the following, two threads of related work are considered. First, our feature-based classification is compared to existing classifications. Second, mapping benchmark is related to existing mapping benchmarks. In this respect, at first the most closely related area of model engineering is examined. Moreover, the more widely related areas of data engineering and ontology engineering are investigated.

Existing Classifications.

Model Engineering. Although model transformations and thus the resolution of heterogeneities between MMs play a vital role in MDE, to the best of our knowledge no dedicated survey examining potential heterogeneities exists.

Data Engineering. In contrast to that, in the area of data engineering a plethora of literature exists for decades highlighting different aspects of heterogeneities in the context of database schemata. A first classification of semantic and structural heterogeneities when integrating two different schemas was presented by Batini et al. in [2]. A systematic classification of possible variations in a SQL statement was presented by Kim et al. in [11], detailing *Table-Table* and *Attribute-Attribute* heterogeneities, e.g., wrt. cardinalities. The classification of Kashyap et al. presented in [10] provides a broad overview on possible heterogeneities in a data integration scenario comprising semantic heterogeneities and conflicts occurring between same modeling concepts. The work of Blaha et al. presented in [4] describes patterns resolving syntactic heterogeneities, comprising same metamodeling concept heterogeneities as well as different metametamodeling concept heterogeneities. Finally, the classification of Legler [13] presents a systematic approach for attribute mappings by combining possible attribute correspondences with cardinalities.

Ontology Engineering. Concerning the domain of ontology engineering pattern collections as well as classifications exist. A pattern collection has been presented by Scharffe et al. in [14]. Thereby correspondence patterns for ontology alignments are presented, but on a rather coarse-grained level, e.g., conditional patterns dealing with attribute differences and transformation patterns, vaguely dealing with different metamodeling concept heterogeneities. With respect to existing classifications, Visser et al. [17] and Klein [12] provide a comprehensive list of semantic heterogeneities. Nevertheless, they have a strong focus on semantic heterogeneities, neglecting syntactic heterogeneities.

Summarizing, although there are several classifications available, none explicitly focuses on the domain of MDE. Therefore we systematically analyzed variation points in the Ecore meta-metamodel in order to extend and adapt existing classifications. In this respect, we aligned on the one hand terms of existing classifications, e.g., most classifications introduced terms for the heterogeneities summarized in our classification by same metamodeling concept heterogeneities. On the other hand, we introduced new heterogeneities stemming from the explicit concepts of references and inheritance in object-oriented metamodels in contrast to existing classifications basing either on the relational or the XML data model. Finally, current classifications miss to explicate how different types of heterogeneities relate to each other, which we formalized by means of a feature model.

Existing Benchmarks.

Model Engineering. To the best of our knowledge no benchmark for mapping systems in the area of MDE exists. Nevertheless, a benchmark for evaluating the performance of graph transformations [16] has been proposed.

Data Engineering. In the area of data engineering Alexe et. al. propose in [1] a first benchmark for mapping systems, thereby presenting a basic suite of mapping scenarios which should be readily supported by any mapping system focussing on information integration. In this respect, ten examples are discussed for which the actual transformation functions are given in terms of XQuery[5] expressions. Additi-

[5]http://www.w3.org/TR/xquery/

tional examples are presented on their homepage[6]. Although the benchmark provides a first set of mapping scenarios it remains unclear how the scenarios have been obtained and if they provide full coverage in terms of expressivity. Although XQuery expressions are given to define the semantics, some of the XQuery functions assume the availability of custom functions which are not provided. Since there are also no RHS models given it is hard to get the actual outcome of the transformation. Finally, some scenarios are not clearly specified with the given query (cf. scenario 2 and 17 on their homepage). A further benchmark called THALIA is presented by Hammer et. al in [7]. It provides researchers with a collection of twelve benchmark queries given in XQuery, focusing on the resolution of syntactic and semantic heterogeneities in a data integration scenario. For every query a so-called *reference schema* (i.e., global schema) and a *challenge schema* is provided (i.e., the schema to be integrated) together with instances. Although the paper claims a systematic classification of semantic and syntactic heterogeneities leading to the presented queries, it is merely an enumeration of heterogeneities where the rationale behind is left unclear.

Ontology Engineering. With respect to the area of ontology engineering, no dedicated mapping benchmark exists. Nevertheless, efforts concerning the evaluation of matching tools, i.e., tools for automatically discovering alignments between ontologies have been spent, resulting in an ontology *matching benchmark*[7] whereby these examples could be of interest for a dedicated mapping benchmark as well.

Summarizing, although both benchmarks from the area of data engineering provide useful scenarios in the context of XML they do not provide a systematic classification resulting in a systematic set of benchmark examples to evaluate the expressivity of a certain mapping system.

7. CONCLUSION AND FUTURE WORK

In this paper we presented a systematic classification of heterogeneities occurring between Ecore-based MMs. Nevertheless, this classification of heterogeneities can also be applied to other semantic data models, comprising the common core concepts this classification bases on. Moreover, a first set of benchmark examples has been proposed stating the requirements a mapping tool should fulfill. Additionally, these benchmark examples can be used to compare solutions realized with ordinary transformation languages. Further work comprise the completion of the benchmark examples to fully cover the classification. However, the success of a benchmark heavily depends on the agreement of the community – thus our collaborative homepage invites for discussions. Finally, a tool evaluation on basis of this benchmark is envisioned comparing and evaluating mapping tools from diverse engineering domains wrt. their expressivity.

8. REFERENCES

[1] B. Alexe, W.-C. Tan, and Y. Velegrakis. STBenchmark: Towards a Benchmark for Mapping Systems. *VLDB Endow.*, 1(1):230–244, 2008.

[2] C. Batini, M. Lenzerini, and S. B. Navathe. A Comparative Analysis of Methodologies for Database Schema Integration. *ACM Comp. Surv.*, 18(4):323–364, 1986.

[3] J. Bézivin. On the Unification Power of Models. *Journal on SoSyM*, 4(2):31, 2005.

[4] M. Blaha and W. Premerlani. A catalog of object model transformations. In *Proc. of the 3rd Working Conference on Reverse Engineering, WCRE'96*, pages 87–96, 1996.

[5] K. Czarnecki, S. Helsen, and U. Eisenecker. Staged Configuration Using Feature Models. In *Proc. of Third Software Product Line Conf.*, pages 266–283, 2004.

[6] M. Del Fabro and P. Valduriez. Towards the efficient development of model transformations using model weaving and matching transformations. *Journal on SoSyM*, 8(3):305–324, July 2009.

[7] J. Hammer, M. Stonebraker, and O. Topsakal. THALIA: Test harness for the assessment of legacy information integration approaches. In *Proc. of the Int. Conf. on Data Engineering, ICDE*, pages 485–486, 2005.

[8] D. Harel and B. Rumpe. Meaningful modeling: What's the semantics of "semantics"? *Computer*, 37:64–72, 2004.

[9] R. Hull and R. King. Semantic Database Modeling: Survey, Applications, and Research Issues. *ACM Comput. Surv.*, 19(3):201–260, 1987.

[10] V. Kashyap and A. Sheth. Semantic and schematic similarities between database objects: A context-based approach. *The VLDB Journal*, 5(4):276–304, 1996.

[11] W. Kim and J. Seo. Classifying Schematic and Data Heterogeneity in Multidatabase Systems. *Computer*, 24(12):12–18, 1991.

[12] M. Klein. Combining and relating ontologies: an analysis of problems and solutions. In *Proc. of Workshop on Ontologies and Information Sharing, IJCAIŠ01*, 2001.

[13] F. Legler and F. Naumann. A Classification of Schema Mappings and Analysis of Mapping Tools. In *Proc. of the GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW'07)*, 2007.

[14] F. Scharffe and D. Fensel. Correspondence Patterns for Ontology Alignment. In *Proc. of the 16th Int. Conf. on Knowledge Engineering, EKAW '08*, pages 83–92, 2008.

[15] A. P. Sheth and J. A. Larson. Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.

[16] G. Varro, A. Schürr, and D. Varro. Benchmarking for graph transformation. In *Proc. of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing, VLHCC '05*, pages 79–88, 2005.

[17] P. R. S. Visser, D. M. Jones, T. J. M. Bench-Capon, and M. J. R. Shave. An analysis of ontological mismatches: Heterogeneity versus interoperability. In *Proc. of AAAI 1997 Spring Symposium on Ontological Engineering*, 1997.

[18] M. Wimmer, G. Kappel, A. Kusel, W. Retschitzegger, J. Schönböck, and W. Schwinger. Surviving the Heterogeneity Jungle with Composite Mapping Operators. In *Proc. of the 3rd Int. Conf. on Model Transformation, ICMT 2010*, pages 260–275, 2010.

[6] http://www.stbenchmark.org/

[7] http://oaei.ontologymatching.org/2010/