Towards Semantic-Aware Collaborations in Systems Engineering

Laurent Wouters*, Stephen Creff*, Emma Effa Bella*†, Ali Koudri*
*IRT SystemX, Paris-Saclay, France
firstname.lastname@irt-systemx.fr
†Sorbonne Universités, UPMC Univ Paris 06, Laboratoire d'Informatique de Paris 6, France
emma.effa-bella@lip6.fr

***Abstract*—The design of nowadays complex systems requires a collaboration between a diversity of stakeholders: from domain experts to customers. For a collaboration to be efficient, the relevant information have to reach the right stakeholder at the right time in a format that is understandable to him/her. We propose a formal framework to integrate the meaning projected by stakeholders onto their data (the denotation), so that it can be unambiguously used by others. An implementation of this framework, relying on the existing language xOWL (extension of OWL2 with behavioral constructs), is then provided to perform the semantic integration of the captured denotations in an MBSE approach.**

***Index Terms*—Collaborative Engineering, Semantic Integration, Modeling, Domain Intention, Ontology Mapping, Behavioral Semantics**

I. Introduction

In this day and age, the design of a complex system requires the collaboration of a vast range of stakeholders. This includes engineers (mechanics, electrics, human factors, ergonomics, etc.) but also the clients, legal and regulation experts, business strategists and more. They are not all required to collaborate with every other experts all the time; but they all have a say in the system's design.

As summed-up in [1], Collaborative Engineering is concerned with various fields, such as the technological, social, socio-technical, decisions making, and negotiation ones. In the context of Model-Based Systems Engineering (MBSE), [2] pointed out some prerequisites for a successful collaboration. One of them is: the exchange of relevant information between the right people at the right time.

One of the root cause of misinterpretations and misunderstandings during the design of a system is that stakeholders may associate different meanings to the same signs [3]. To improve this situation in the collaboration, we propose to capture the surface meaning stakeholders associate to their data, so that it can be understood by others, i.e. their denotation [4]. Doing so, the interpretation of the incoming signs by the receiving stakeholder can be made less ambiguous through the use of the captured denotation.

In a MBSE context, dealing with structured languages and considering the use of SysML [5], in a given tool, one may interpret the sign *"Brake"* in the Figure 1, as the concept *"System Function Brake"* of the domain *"Functional Architecture"* (i-a), another as a "SysML::Block" (i-b), and finally as a "UML::Class" with a "Block" stereotype (i-c). From a Systems Engineering perspective, the expert from the *"Functional Architecture"* domain would capture its denotation: "*System Function is represented by SysML::Block*". A collaborator (an expert from another domain), would then be able to use the captured denotation to extract the relevant information and use it in its domain. The original meaning in the language's specification, here SysML, then becomes irrelevant for our purpose.

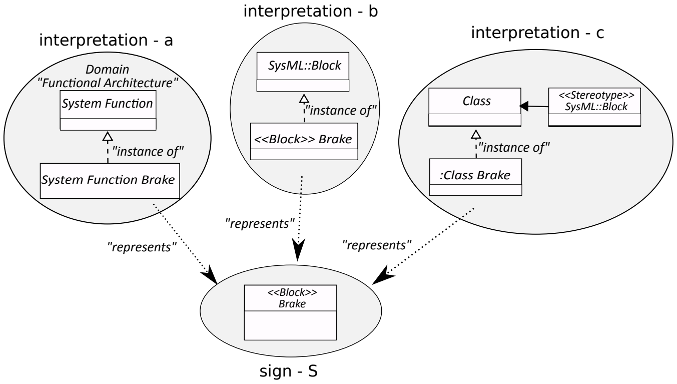

Fig. 1. Interpretations of signs - a SysML example

As presented above, a required element is the capture of the stakeholders' original denotation for their data, i.e. the semantics they project onto their data. Although the capture and sharing of the denotation is necessary, it is not sufficient. The recipient of the information still need to relate the concepts from the sender's denotation to his own concepts [2]. We argue in this paper that a possible way to reduce the risk of misinterpretation, and thus to improve the quality of the collaborations during the design of a complex system, is to support the meaningful exchange of engineering data at the system level. Indeed, this approach differs from the usual dogma in the MBSE community by refuting the need for a central common vocabulary encompassing all domains and advocating for the point-to-point mediation of meaning during focused collaboration. The use of a common language does not imply a shared understanding.

Therefore, we propose to formally capture the meaning projected by the stakeholders onto their data (the denotation) so that it can be unambiguously used by others. More specifically, this paper focuses on the aspect of the collaboration regarding the semantically consistent use of the exchanged information in conjunction with others; i.e. how can we relate the meaning projected by multiple stakeholders so that they can later exchange information that is meaningful to them. An implementation of this formal framework, relying on xOWL [6] (extension of OWL2 [7], [8] with behavioral constructs), is then provided to perform the semantic integration of the captured denotations in an MBSE approach.

The remainder of the paper is organized as follows. Section II formalizes our approach for this particular issue. Section III presents some related works. Finally, the last Section draws our conclusion and narrows down possible future works.

II. Formal Framework for Semantic-Aware Collaborations

Our approach is to provide a way to relate the denotations of the stakeholders for their data, i.e. a way to integrate the meaning projected by the stakeholders onto their data. This approach is really twofold: i) capturing the denotations of the stakeholders as languages, ii) the integration of their semantics through a federation of languages.

A. Some background on modeling and languages

Here, we build upon the linguistic approach to metamodeling of [9] where a modeling language is defined by its abstract syntax, which specifies its linguistic constructs. Declarative and/or operational semantics define the structural and behavioral aspects of the language's domain. Consequently, the semantics of a modeling language is a mapping from the abstract syntax onto the semantic domain, as defined in [10], [11]. When formal semantics is desired, the semantic domain is a mathematical theory such as Description Logic; it can also be simply expressed in a natural language, as noted in [10].

Leaving out the concrete syntax and its mapping, a formal language is noted $L = \langle A, S, M \rangle$ where A is the language's abstract syntax, S a semantic domain (Description Logics for example) and M is a semantic mapping, i.e. a function from A to S ($M : A \rightarrow S$).

A model O expressed in the language L is said to be a linguistic instance of L. It is defined by saying that all elements of O are expressed using elements of the abstract syntax A. Mathematically, every element o of O is mapped to an element t of A. Following [9], this is noted $o \lhd^l t$ and is extended to models $O \lhd^l L$. The function mapping elements of the model to their corresponding abstract syntactic construct is denoted I_O:

$$I_O = \{(o \mapsto t) \mid o \in O \wedge t \in A \wedge o \lhd^l t\}$$

The semantics of a model O, S_O maps each model element to an element of the semantic domain S ($S_O : O \rightarrow S$) can then be computed by combining M with I_O. For all o in O, $S_O(o)$ is equal to $M(I_O(o))$, thus:

$$S_O = M \circ I_O$$

B. Intention-Specific Modeling Language (ISML)

In this paper, we hypothesize that a stakeholder's denotation can be captured as a formal language, here designated by Intention-Specific Modeling Language (ISML), because it is specific to the stakeholder's intention regarding its data. An ISML capturing a stakeholder's denotation will be composed of the same concrete syntax as the original language (same signs). However, the abstract syntax will contain the concepts projected by the stakeholder onto its data and the semantic domain will give their mathematical interpretation.

Back to the example of the functional architecture, the original language for the model is SysML. The ISML that captures the system engineer's denotation will then have: i) the sign "SysML::Block" in its concrete syntax; the concept of "System function" in its abstract syntax; and the mathematical interpretation of "System function" in its semantic domain. It is important to note that ISMLs are specific to a particular stakeholder, or class of stakeholders in a collaboration. They are also specific to original structured languages (signs) used by the stakeholders. This means that different classes of stakeholders can use the same original structured language to express their data, but associate different meanings to the signs. This case is a very common one when general-purpose modeling languages such as UML [12] are used, or when the information is structured in implicit forms such as spreadsheets.

Defining an ISML L_i using a meta-language L_{meta} is done by expressing its abstract syntax using L_{meta}. This is the case, for example, when we use Meta-Object Facility (MOF) [13] to specify the ISML. In this framework, if $L_{meta} = \langle A_{meta}, S_{meta}, M_{meta} \rangle$ and $L_i = \langle A_i, S_i, M_i \rangle$, this is defined by:

1) $A_i \lhd^l L_{meta}$, the abstract syntax A_i is a model expressed in L_{meta},
2) $S_i = M_{meta} \circ I_i$, its semantic domain is the set of couples mapping each syntactic construct of L_i to its meaning in L_{meta},
3) $M_i : A_i \rightarrow M_{meta} \circ I_i, M_i(a) = (a \mapsto M_{meta} \circ I_i(a))$, an element a of the ISML's abstract syntax A_i is mapped to the corresponding couple $(a \mapsto s)$ of the semantic domain of the ISML.

Once the ISML L_i is defined, it can be used to express a model O_i ($O_i \lhd^l L_i$). The function I_{O_i} captures this linguistic instantiation of A_i into O_i:

$$I_{O_i} : O_i \rightarrow A_i$$
$$I_{O_i} = \{(o \mapsto a) \mid o \in O_i \wedge a \in A_i \wedge o \lhd^l a\}$$

The semantics of the model O_i, S_{O_i} is easily obtained using the previous definition of the semantics of a model: $S_{O_i} = M_i \circ I_{O_i}$. It can be made further explicit as:

$$S_{O_i} = \left\{ (o \mapsto (a \mapsto s)) \mid \begin{array}{r} o \in O_i \wedge \ a \in A_i \\ \wedge \ o \lhd^l a \\ \wedge \ (a \mapsto s) \in M_{meta} \circ I_i \end{array} \right\}$$

The function S_{O_i} associates a model element $o \in O_i$ to an element of the semantic domain of L_i, i.e. a couple of the form

$(a \mapsto s)$ where o is the linguistic instance of a $(o \triangleleft^l a)$. It is interesting to note that the semantics of a model element o is not simply related to an element of the semantic domain of the meta-language, but to the association of an element of the abstract syntax (A_i) and its corresponding element of the semantic domain S_{meta}. By consequences, different ISMLs expressed in the same meta-language L_{meta} can use the same abstract syntax elements of A_{meta} but still have different semantics. For example, a first ISML can be expressed using MOF (the meta-language) and define a class x in its abstract syntax, and a second ISML still expressed using MOF defines a class y. In the semantic domain of the first ISML, we will have $(x \mapsto \text{class})$; and in the second we will have $(y \mapsto \text{class})$. The consequence is that models expressed in the first ISML can have elements expressed with x that have the first couple as semantic; and models expressed in the second ISML have elements expressed with y that have the second couple as semantic. Because the semantics of the two ISMLs are different, the models are not trivially compatible.

C. ISMLs federation

Once the stakeholders' denotations have been captured as ISMLs, we need to find out the relations between them in order to be able, in the end, to say that sign A in domain X may correspond to sign B in domain Y. Hereafter, we present a formal federation framework for the **syntactic** and **semantic** integration of structured languages. Because we capture the denotations as languages, we propose to rely on techniques to do so.

First, the **syntactic** integration corresponds to the mapping of the concepts in the ISMLs' abstract syntaxes (ISMLs Federation). Then, the **semantic** integration means that the mapping of the **syntactic** elements (concepts) must be consistent with their associated **semantics** in the respective semantic domains, i.e. their associated mathematical interpretations must be equivalent. For structural semantic integration, we aim at relying on ontology mapping approaches. Finally, an extension of these approaches is described to take into account behavioral semantics when integrating ISMLs. The objective of this framework is to demonstrate how to achieve such an integration and to position how ontology mapping approaches can be leveraged at system level.

The goal is to relate the different ISMLs corresponding to stakeholder's denotation of their data. This is to be achieved through the **syntactic** and **semantic** integration of the ISMLs, here noted L_i. This integration is performed in the form of a federation in this paper. The federation must produce a syntactically and semantically integrated access to the federated languages L_i. For this purpose, a federation relies on the production of a federation language L_f.

$$\forall i, L_i = \langle A_i, S_i, M_i \rangle$$

A federation of these ISMLs is then specified as a federation language L_f associated to a collection of morphisms φ_i, one for each ISML of interest.

$$L_f = \langle A_f, S_f, M_f \rangle$$

The federation language L_f essentially acts as the blue-print for the federated models representing the results of the realization of the federation on models expressed in the ISMLs of interest, i.e. the stakeholders' data. A morphism φ_i captures the relations between the abstract syntax elements of L_i and the abstract syntax elements of L_f:

$$\forall i, \exists \varphi_i : A_i \leftrightarrow A_f$$

As defined above, φ_i is a relation that is able to associate any number of elements of A_i to any number of elements of A_f. This means that some particular φ_i may not map the complete abstract syntax of the associated ISML. The ISMLs federation is realized when the semantics associated to the mapped part of a ISML is equivalent to its associated counter-part in the federation language L_f. This is expressed as:

$$\forall i, \exists m_i : S_i \leftrightarrow S_f, \tag{1}$$

$$\forall \psi \subset \varphi_i, \psi = \text{dom}(\psi) \triangleleft \varphi_i \;\wedge\; \psi = \varphi_i \triangleright \text{ran}(\psi), \tag{2}$$

$$\text{ran}(\text{ran}(\psi) \triangleleft M_f) = m_i(\text{ran}(\text{dom}(\psi) \triangleleft M_i)) \tag{3}$$

In this definition, for all language L_i a relation m_i must exist between the semantic domains S_i and S_f (1) such that for all self-contained subset ψ of the morphism φ_i (2) the semantics of the abstract syntax elements $\text{dom}(\psi)$ is equivalent to the transformed semantics of its mapped elements in A_f (3). According to line 2, a self-contained subset of φ_i, is a subset ψ such that there is no other couple $(a_i \mapsto a_f) \in \varphi_i$ not in ψ such that a_i is in the domain of ψ or a_f in its range. This means that all the considered mapped elements from A_i and A_f in this subset are not referred to by mappings outside the subset. In line (3), $\text{dom}(\psi) \triangleleft M_i$ is the subset of the semantic mapping M_i for the abstract syntax elements considered in the current subset of the morphism. The range of this partial relation is then a subset of the semantic domain S_i. In the same way, $\text{ran}(\psi) \triangleleft M_f$ is the subset of the semantic mapping M_f for the abstract syntax elements in A_f considered in current ψ. The transformed subset of S_i must be equal to the corresponding subset of S_f for the federation to be semantically sound. In practice, only the φ_i morphism would have to be made explicit for the actual federation to take place. It suffices to demonstrate that the relation m_i exists. The relation can be as trivial as the identity function if the semantic domains S_i and S_f are the same.

It can be noted that whereas the φ_i morphism performs the **syntactic** federation of L_i, the **semantic** federation is ensured by the relation m_i. The above definition does not tell, however, how exactly the **syntactically** and **semantically** sound federation should be achieved.

D. ISMLs federation with ontology mapping

The mapping of ontologies is a collection of techniques for the **semantically** sound integration of different ontologies expressed in an ontology description language. In this context, a possible solution for the realization of the federation of ISMLs would be to express the abstract syntaxes of the addressed ISMLs in ontologies and rely on one of these techniques to

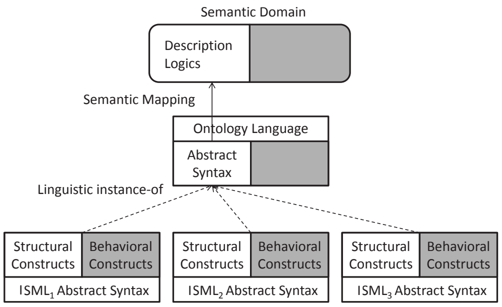

Fig. 2. Semantic Federation of ISMLs with Ontologies

perform the required **syntactic** and **semantic** federation of the languages. Figure 2 illustrates this approach. An important point, though, is that the behavioral aspects of the different ISMLs will have to be ignored whenever the abstract syntaxes are expressed using an ontological language that only maps to declarative semantics. To represent this problem, the relevant parts are grayed in Figure 2. Still, the rest of this sub-section demonstrates how an ontology mapping approach can be used to perform a federation of ISMLs, the following one addressing the gray parts.

The mapping of ontologies is defined in [14] as “the task of relating the vocabulary of two ontologies that share the same domain of discourse in such a way that the mathematical structure of ontological signatures [...] are respected.” More formally, the ontology mapping of O_1 to O_2 expressed in the language $L_O = \langle A_O, S_O, M_O \rangle$ is the morphism f such that:

$$f : O_1 \rightarrow O_2 \tag{4}$$

$$\wedge \; \forall \; (o_1 \mapsto o_2) \in f, \exists \; s \in S_O, \tag{5}$$

$$(o_1 \mapsto s) \in S_{O_1} \wedge (o_2 \mapsto s) \in S_{O_2} \tag{6}$$

This means that (4) the morphism f is a function associating elements from O_1 to elements of O_2 so that (5) for all such mapping $(o_1 \mapsto o_2)$ defined in f, there is s in the semantic domain of L_O and (6) o_1 and o_2 are associated the same semantic s.

Through the formal framework defined above, we use an ontology description language as the meta-language L_{meta} in which the abstract syntax of the ISMLs of interest will be expressed. We then have the meta-language:

$$L_{meta} = \langle A_{meta}, S_{meta}, M_{meta} \rangle$$

A particular ISML L_i specified with this meta-language is then noted, as presented previously:

$$L_i = \langle A_i, M_{meta} \circ I_i, M_i \rangle$$

The function I_i capturing the linguistic instantiation relations is still the same as herebefore. The same goes for the M_i mapping the abstract syntax A_i onto the semantic domain of L_i, i.e. $M_{meta} \circ I_i$.

To perform the federation, the federation language L_f shall also be specified using the same meta-language L_{meta}.

$$L_f = \langle A_f, M_{meta} \circ I_f, M_f \rangle$$

The **syntactically** and **semantically** sound federation is then realized for the condition given previously:

$$\begin{aligned} \forall \; i, \exists \; m_i : M_{meta} \circ I_i \leftrightarrow M_{meta} \circ I_f, \\ \forall \; \psi \subset \varphi_i, \psi = \text{dom}(\psi) \lhd \varphi_i \;\; \wedge \;\; \psi = \varphi_i \rhd \text{ran}(\psi), \\ \text{ran}(\text{ran}(\psi) \lhd M_f) = m_i(\text{ran}(\text{dom}(\psi) \lhd M_i)) \end{aligned}$$

Here, we make the assumption that φ_i is an adequate morphism performing an ontology mapping as defined above. With this assumption, we have:

$$\begin{aligned} &\varphi_i : A_i \rightarrow A_f \\ &\wedge \; \forall \; (a_i \mapsto a_f) \in \varphi_i, \exists \; s \in S_{meta}, \\ &\quad (a_i \mapsto s) \in M_{meta} \circ I_i \wedge (a_f \mapsto s) \in M_{meta} \circ I_f \end{aligned}$$

Using the definition of the semantic mapping of language expressed in a meta-language given above, we also have:

$$\begin{aligned} \forall \; (a_i \mapsto a_f) \in \varphi_i, \exists \; s \in S_{meta}, \\ (a_i \mapsto (a_i \mapsto s)) \in M_i \wedge (a_f \mapsto (a_f \mapsto s)) \in M_f \end{aligned}$$

Because the morphism φ_i is a function under the current assumption, the smallest self-contained subsets of φ_i are single couples. They can be trivially recombined to form more complex subsets. The federation condition can be rewritten as:

$$\begin{aligned} \forall \; i, \exists \; m_i : M_{meta} \circ I_i \leftrightarrow M_{meta} \circ I_f, \forall \; (a_i \mapsto a_f) \in \varphi_i, \\ \text{ran}(\{a_f\} \lhd M_f) = m_i(\text{ran}(\{a_i\} \lhd M_i)) \\ \Leftrightarrow \\ \forall \; i, \exists \; m_i : M_{meta} \circ I_i \leftrightarrow M_{meta} \circ I_f, \forall \; (a_i \mapsto a_f) \in \varphi_i, \\ \exists \; s \in S_{meta}, \{(a_f \mapsto s)\} = m_i(\{(a_i \mapsto s)\}) \end{aligned}$$

The federation condition under the current assumption is then that it must exist a relation m_i so that for each couple in φ_i mapping an element of the abstract syntax A_i to an element of the abstract syntax A_f both elements are associated the same semantic s. This condition is always verified by a function m_i trivially derived from the morphism φ_i:

$$m_i = \left\{ \begin{array}{l} \quad ((a_i \mapsto s) \mapsto (a_f \mapsto s)) \mid \\ \quad a_f = \varphi_i(a_i) \\ \wedge \; (a_i \mapsto s) \in M_{meta} \circ I_i \\ \wedge \; (a_f \mapsto s) \in M_{meta} \circ I_f \end{array} \right\}$$

The conclusion is that under the assumption that all φ_i represent ontology mappings, the condition for a **syntactically** and **semantically** sound federation of ISMLs is verified. This means that ontology mapping approaches can be directly used to perform the integration, i.e. the mapping of stakeholders' denotations, under the condition that the behavioral aspects are ignored.

E. Adding behavior

From the previous conclusion, one possible way to realize the mapping of the stakeholders' denotation is to express the abstract syntaxes (concepts) of the associated ISMLs as OWL2 [7] ontologies. Doing so, we can rely on existing ontology mapping approaches to perform the **syntactic** and **semantic** integration that realizes the mapping of the denotations. A major drawback of using pure OWL2, however, is its inability to formalize behavioral semantics. For this purpose, we propose to rely on the xOWL language introduced in [6]. xOWL is an extension of OWL2 that supports the expression of behavioral constructs and capture their formal semantics. These are defined in conjunction with the standard model theoretic semantics (also called direct semantics) of OWL2 defined in [8]. Structural Operational Semantics have been chosen as the semantic domain of xOWL for the behavior because they enable the compact expression of the semantics and are easily integrated with Description Logics, the semantic domain of the OWL2 language. The behavioral semantics of xOWL are then defined with an abstract interpreter [15].

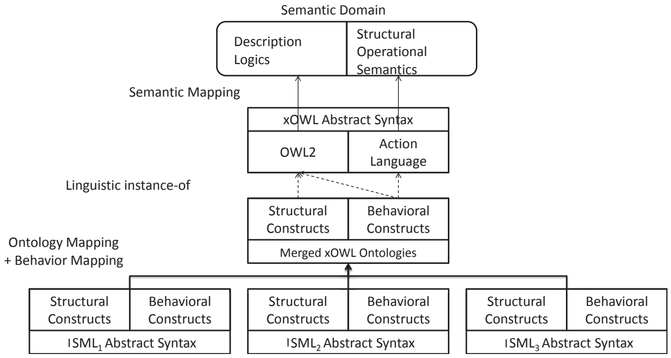

Fig. 3. Semantic Federation of ISMLs with xOWL

Using xOWL, the structural and behavioral aspects of the ISMLs can be expressed. The structural aspects will be expressed using pure OWL2. For example, the part of a domain about state machines will be expressed in xOWL as follow. The *State* and *Transition* concepts are expressed as OWL2 classes with the corresponding relations. The algorithms describing how to execute state machines are then expressed in the xOWL ontology using Clojure[1] code. This is the purpose of the *smNextState* and *smSimulate* function:

Listing 1. State machines in xOWL

```
Ontology( <http://holons–hub.com/domains/syseng>
    Declaration(Class(:State))
    Declaration(Class(:Transition))
    Declaration(Class(:SystemEvent))
    FunctionalObjectProperty(:origin)
    ObjectPropertyDomain(:origin :Transition)
    ObjectPropertyRange(:origin :State)
    FunctionalObjectProperty(:target)
    ObjectPropertyDomain(:target :Transition)
    ObjectPropertyRange(:target :State)
    FunctionalObjectProperty(:symbol)
    ObjectPropertyDomain(:symbol :Transition)
    ObjectPropertyRange(:symbol :SystemEvent)

  FunctionDefinition(:smNextState (fn [state symbol]
    (xowl/firstValueOf
      (xowl/sparql (str
            ”PREFIX : <http://holons–hub.com/domains/syseng
                #>”
            ” SELECT ?s WHERE { GRAPH ?g { ”
            ” <” event ”> :origin <” state ”> .”
            ” <” event ”> :target ?s . } }”))
          ”s”)))

  FunctionDefinition(:smSimulate (fn [init events]
    (if (nil? init)
      nil
      (if (empty? events)
        init
        (smSimulate
          (smNextState init (first events))
          (next events))))))
)
```

In this approach, summarized in Figure 3, the mapping of the denotations of stakeholders in a collaboration is achieved by 1) the capture of the stakeholders' denotations as ISMLs, 2) the expression of the abstract syntaxes of the ISMLs as xOWL ontologies, 3) the expression of a federated xOWL ontology representing the abstract syntax of federated ISMLs and 4) the use of an ontology mapping approach for the specification of the φ_i morphisms performing the federation.

III. Related Works

Some related approaches come from the MBSE community, others from the Semantic Web one.

The MBSE community often distinguish between “formal” and “informal” models. Whereas “formal” models are intended to be unambiguously interpreted by machines and still readable to humans, “informal” models are solely intended for humans. Not only do they have a greater risk for misinterpretation, but also is this fuzziness actually embraced by the standardization organization as a feature. For example, the UML language includes the notion of “semantic variation point” [12] that paves the way for stakeholder-specific interpretations of a UML model. However this particular trait of the language was originally intended as a compromise for disagreeing future tool vendors for the language. Its purpose was not to recognize the variability in the end-users' interpretations.

The failure to recognize this variability is a common symptom of the MBSE community. MBSE practitioners usually value the identification of a common vocabulary in order to help stakeholders from different domains to collaborate [16]. This way of thinking takes a concrete form in tools and standards that emphasize a central repository of knowledge using a single vocabulary for all considered domains [17]. This fails to recognize the variability in the meaning associated to the vocabulary and do not help at reducing the chance for misinterpretations. It has to be noted that some effort in the MBSE community goes toward the use of ontologies in order to build common vocabularies, integrate and share data [18], [19]. These approaches leverages ontologies as a way to provide a holistic representation of the system under study across domains so that consistency analyses can be performed. They still enforce the dogma of a centralized repository of knowledge with a single common vocabulary. The stakeholders still have to project their domain's concepts onto the foreign vocabulary in the central repository.

As presented in Section II, ontology mapping approaches can be used to realize the **semantic** federation of the ISMLs

[1] https://clojure.org/

representing the denotations of various stakeholders in a collaboration. With broad strokes, ontology mapping approaches can be classified into two categories: First, Ontology Merging, which consists in merging all the considered ontologies into a single one [20]. The approaches in this category perform a **syntactic** and **semantic** federation by resulting in a single ontology representing the integrated abstract syntaxes of all the considered DSMLs. Second, Semantic Bridges, which consists in the definition of semantic mappings (or bridges) between the considered ontologies [21]. The approaches in this category only perform a **semantic** federation because they only focus on the semantic alignment of the ISMLs.

In the Model-Driven Engineering community, several works also have identified and addressed the problem of the **syntactic** and/or **semantic** federation of multiple languages. The earliest work in this regard is the introduction of Semantics Units in [22]. In this approach, the authors define the notation of Semantic Unit as a gateway for the specification of formal declarative and operational semantics for languages. Semantics Units are always mapped to the same semantic domain, thus realizing their semantic integration. Given a set of languages, the approach consists in building a set of Semantic Units, which semantics can be merged. How these Semantic Units are built completely depends on the given languages, as noted in [22].

IV. Conclusion

In systems engineering, collaborations between stakeholders from different domains involved in the conception of a complex system is ever more necessary. In this paper, we argue that the collaboration shall be facilitated and made more efficient by reducing the misunderstanding and misinterpretation of the stakeholders' data when exchanged. Our approach to tackle this issue is to formally capture and integrate the meaning projected by the stakeholders onto their data (the denotation), so that it can be unambiguously shared. A formal framework is consequently proposed, and an implementation of it, relying on xOWL, is provided to perform the semantic integration of the captured denotations in an MBSE approach.

A first limitation of this approach is that it requires the capture of the stakeholders' denotations for their data. Although we propose a way to formalize the captured denotations as languages, the capture itself can be challenging. At this time, it is possible to manually realize the capture through interviews with the experts; but this process should be at least partially automated. This constitutes one of the research tracks for our future works.

Acknowledgements. This research work was supported by the IRT SystemX, Paris-Saclay, France, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir".

References

[1] S.-Y. Lu, W. Elmaraghy, G. Schuh, and R. Wilhelm, "A scientific foundation of collaborative engineering," *CIRP Annals - Manufacturing Technology*, vol. 56, no. 2, pp. 605 – 634, 2007.

[2] L. Wouters, S. Creff, E. E. Bella, and A. Koudri, "Collaborative systems engineering: Issues & challenges," in *Proceedings of the 2017 IEEE 21st CSCWD Conference*, W. Shen, P. Antunes, N. H. Thuan, J.-P. Barths, J. Luo, and J. Yong, Eds., pp. 486–491.

[3] I. M. Sayer, "Misunderstanding and language comprehension," *Procedia - Social and Behavioral Sciences*, vol. 70, pp. 738 – 748, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877042813001195

[4] R. Barthes, *Elements of Semiology*. Hill and Wang, 1967.

[5] OMG, *Systems Modeling Language*, OMG Std., Rev. 1.4, 2015.

[6] L. Wouters and M.-P. Gervais, "xOWL an Executable Modeling Language for Domain Experts," in *Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference*. IEEE, 2011.

[7] W3C, "OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax," W3C, Tech. Rep., 2009. [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/

[8] ——, "OWL 2 Web Ontology Language Direct Semantics," W3C, Tech. Rep., 2009. [Online]. Available: http://www.w3.org/TR/2009/REC-owl2-direct-semantics-20091027/

[9] T. Kühne, "Matters of (Meta-) Modeling," *Software and Systems Modeling*, vol. 5, pp. 369–385, 2006.

[10] D. Harel and B. Rumpe, "Meaningful Modeling: What's the Semantics of "Semantics"?" *Computer*, vol. 37, pp. 64–72, 2004.

[11] T. Clark, A. Evans, S. Kent, and P. Smmut, "The MMF Approach to Engineering Object-Oriented Design Languages," in *Proceedings of the Workshop on Language Descriptions, Tools and Applications*, 2001.

[12] OMG, *Unified Modeling Language*, OMG Std., Rev. 1.5, 2003.

[13] ——, *Meta Object Facility (MOF) 2.4.2*, OMG Std., Rev. 2.4.2, 2014.

[14] Y. Kalfoglou and W. M. Schorlemmer, "Information-Flow-Based Ontology Mapping," in *CoopIS/DOA/ODBASE*, ser. LNCS, vol. 2519. Springer, 2002, pp. 1132–1151.

[15] L. Wouters, "Multi-Domain Expert-User Modeling Infrastructure," Ph.D. dissertation, Université Pierre et Marie Curie, 2013.

[16] M. Schmit, S. Briceno, K. Collins, D. Mavris, K. Lynch, and G. Ball, "Semantic Design Space Refinement for Model-Based Systems Engineering," in *2016 Annual IEEE Systems Conference (SysCon)*. Institute of Electrical and Electronics Engineers (IEEE), apr 2016. [Online]. Available: http://dx.doi.org/10.1109/SYSCON.2016.7490579

[17] J.-L. Voirin and S. Bonnet, "Arcadia: Model-Based Collaboration for System, Software and Hardware Engineering," in *Complex Systems Design & Management, poster workshop (CSD&M 2013)*, 2013.

[18] C. Hennig, A. Viehl, B. Kmpgen, and H. Eisenmann, "Ontology-Based Design of Space Systems," in *Lecture Notes in Computer Science*. Springer Nature, 2016, pp. 308–324. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46547-0_29

[19] E. Kharlamov, B. C. Grau, E. Jiménez-Ruiz, S. Lamparter, G. Mehdi, M. Ringsquandl, Y. Nenov, S. Grimm, M. Roshchin, and I. Horrocks, "Capturing Industrial Information Models with Ontologies and Constraints," in *Lecture Notes in Computer Science*. Springer Nature, 2016, pp. 325–343. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46547-0_30

[20] A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy, "Ontology Matching: A Machine Learning Approach," in *Handbook on Ontologies*, ser. International Handbooks on Information Systems. Springer, 2004.

[21] A. Maedche, B. Motik, N. Silva, and R. volz, "MAFRA - A MApping FRAmework for Distributed Ontologies," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, vol. 2473. Springer, 2002, pp. 235–250.

[22] K. Chen, J. Sztipanovits, S. Abdelwahed, and E. K. Jackson, "Semantic Anchoring with Model Transformations," in *ECMDA-FA*, 2005.