

MATHEMATICAL MODELS TOWARDS SELF-ORGANIZING FORMAL FEDERATION LANGUAGES BASED ON CONCEPTUAL MODELS OF INFORMATION EXCHANGE CAPABILITIES

Andreas Tolk, PhD

Engineering Mgt & Systems Engineering
Old Dominion University
Norfolk, VA 23529, USA

Saikou Y Diallo
Charles D Turnitsa

Virginia Modeling Analysis & Simulation Center
1030 University Blvd
Suffolk, VA 23435, USA

ABSTRACT

Conceptual models capture information that is crucial for composability of legacy solutions that is not formally captured in the derived technical artifacts. It is necessary to make this information available for the selection (or elimination) of available solutions, their orchestration, and their execution. Current standards barely address this class of problems. The approach presented in this paper is the first step towards self-organizing federation languages. The system interfaces are described in form of exchangeable data. The context of information exchange (syntax, semantics, and pragmatics) is captured as metadata. These metadata are used to identify the elements of a formal federation language that links model composability and simulation interoperability based on conceptual model elements. The paper describes the formal process of selection, orchestration, and execution and the underlying mathematics for the information exchange specifications that bridge conceptual and engineering levels of the federation process.

1 INTRODUCTION

Most textbooks on systems engineering in general and Modeling and Simulation (M&S) development in particular assume that systems are built from scratch. Based on the problem to be solved, conceptual modeling helps to shape the outline and constraints of a system. Engineers take these conceptual ideas and build the system specification artifacts, using methods supported by UML, SysML, IDEF, and others, and implement the system.

In practice however very few systems start from scratch. Legacy systems that already perform desired functionality need to be reused. New bundles of functionality have to be provided by systems of systems, which by themselves require new system engineering methods (Keating et al., 2003). In general, reuse of existing solutions is often mandated and part of the constraints for the systems development process. Theoretically, the integration of existing systems into a system of systems is sup-

ported by the documentation developed in parallel to the system engineering process: architectures, interface descriptions, process, and data model. In practice, this is not always the case. But even if these engineering artifacts are available, the underlying conceptual model is seldom captured with the same rigor and often conceptual decisions are insufficiently documented for reuse.

In this paper, we will first look at the information that needs to be provided by models to support the reuse of services in general and the reuse of M&S services in particular. Next, we will look into the process that can be applied to re-engineer the information exchange resulting in support of selection, orchestration, and execution. Finally, we will document a mathematical model to capture information exchange and show how this leads towards self-organization of formal federation languages based on the conceptual models derived in the re-engineering process.

2 LAYERED MODELS FOR REUSE

Models are purposeful abstractions of reality. The abstraction is mainly driven by the intent when building the system. Brachman (1979) used the work of McCarthy and Hayes (1969) on artificial intelligence and what is needed to model understanding to introduce the following layers leading from linguistics down to implementation:

- Linguistic level: the level of arbitrary concepts, words, and expressions of natural languages describing the system.
- Conceptual level: the level of semantic relations, linguistic roles, objects, and actions describing the system.
- Epistemological level: the level of defining concept types and subtypes, inheritance and structuring relations, objects with type information to describe the system
- Logical level: the level of symbolic logic with propositions, predicates, variables, quantifiers, and operators, i.e., algorithmic description of the system.

- Implementation level: the level of data structures and language specific notions.

As recently documented by Federici et al. (2006), these abstractions takes place on several layers in current developments as well, starting with the conceptualization of the real world referents (the system or part of the real world that should be modeled) and resulting in coding decisions for the final implementation. They introduce a layer of models as follows:

- The Target System is the first level of abstraction. It is constituted by the object of study that is determined by a specific point of view on a portion of reality, considered “isolated” from the rest of the universe.
- Abstract Model: the second level is the abstract model of the target system.
- Computational Model: the third level is the specific computational model that has been adopted to represent the abstract model. It is a formal model.
- The software model: the fourth level is the software translation of the Computational Model.

Both contributions make clear that a lot of conceptual work is needed and conducted in a number of stages before the state of actually building or coding a system starts. We cannot ignore all these decisions, assumptions, and constraints when we reuse the solutions or compose them into a new solution. Wache et al. (2001) already showed that the aspects captured by ontology are needed to integrate information (data in context). Gnägi et al. (2006) used these concepts to set up conceptual models for data centric applications. Current simulation standards are missing these dimensions. Available standards allow sharing concepts, but not their roles in the applications; they allow sharing concepts, but not the implementation details of entities; they allow sharing semantics (meaning of data) to some degree, but not pragmatics (intention). In order to contribute to a better taxonomy of terms, Oren et al. (2007) distinguish between various categories of understandings, which are

- Lexical understanding (recognizing the symbols and tokens),
- Syntactical understanding (recognizing the elements describing an entity),
- Morphological understanding (recognizing the structure of the properties describing the underlying concepts or attributes of the entities),
- Semantic understanding (recognizing the meaning of entities and their relations), and
- Pragmatic understanding (recognizing the intention behind using the identified entities).

Tolk et al. (2006) show that similar structures are needed on the metadata level in order to support compos-

able M&S services. The Levels of Conceptual Interoperability Model (LCIM) distinguishes the following levels

- Technical Interoperability (exchanging bits and bytes),
- Syntactic Interoperability (using common protocols, exchanging data),
- Semantic Interoperability (using common data models, understanding meaning),
- Pragmatic Interoperability (using common workflow models, understanding the intent of sending and context of using of data);
- Dynamic Interoperability (using common execution models, understanding the effects of exchanging data in the sending and receiving services), and
- Conceptual Interoperability (using common conceptual models, being aware of abstractions, constraints, concepts, relations, and roles).

Dobrev et al. (2007) show how this model can be used to support interoperation in general applications. Zeigler and Hammonds (2007) use it to compare it with their ideas on using ontological means in support of interoperation. Tolk et al. (2006) documents use cases.

All these layered models agree that the artifacts used for implementation are the results of an important conceptualization process. All the results should be captured formally in a conceptual model. However, the community is still in the process of agreeing on what a conceptual model it. The authors propose to understand the conceptual model as the “*bridge between the real world observation and the high-level implementation artifacts.*”

Robinson (2008) defines the conceptual model as “*a non-software specific description of the simulation model that is to be developed, describing the objectives, inputs, outputs, content, assumptions, and simplifications of the model.*” He furthermore points out there is a significant need to agree on how to do develop conceptual models and capture information formally. The authors not only agree, but identify the urgent need to capture the conceptualization decisions in form of metadata accessible and understandable by machines in support of selection (what services to use), orchestration (which service provides which functionality when), and execution (running the system).

As stated by Tolk (2006), *Interoperability of Simulation Systems requires Composability of Conceptual Models!* We use the definitions based on the proposal of Page et al. (2004). They distinguish between the following categories of interoperation:

- *Integratability* contends with the physical/ technical realms of connections between systems, which include hardware and firmware, protocols, etc.
- *Interoperability* contends with the software- and implementation details of interoperations, including exchange of data elements based on a common data interpretation, etc.

- *Composability* contends with the alignment of issues on the modeling level. The underlying models are purposeful abstractions of reality used for the conceptualization being implemented by the resulting simulation systems.

As stated in the introduction, reuse and composition of legacy systems into a service-oriented architecture belong to the standard tasks to be performed by system engineers. When developing these systems originally, many important conceptual decisions were made reflecting the intended use and operational requirements that resulted in conceptual decisions that ultimately resulted in the implemented solution. However, if at all the implementation artifacts may have survived. In order to compose systems, we must be aware of the conceptual decisions, the “*objectives, inputs, outputs, content, assumptions, and simplifications*” that are driving the engineering artifacts. On the long term, conceptual models as envisioned by Robinson (2008) will support this task. On the short term, processes are needed to support the consistent re-engineering or at least the conceptual constraints of legacy systems.

In the next section, we will present a formal process to use the conceptual modeling ideas and artifacts to identify the information exchange needs (what receiving systems expect) and information exchange capabilities (what producing systems can provide). This results in a formal language that can be spoken within such a federation. If the re-engineered artifacts are followed consistently, they can be used to lead towards self-configuring languages of federations. As the focus of this paper is on data and processes, supported operational activities (business models, concepts of operations) are not yet included in sufficient detail. The approach described here is only a first step. Although this is not sufficient, it is necessary and so far has proven to be a solid foundation.

We will start with the operationally motivated information exchange modeling process, mainly described in section 3. This will be the basis to motivate the mathematical model presented in section 4. This model allows generalization and – as such – becomes the basis for self-configuration of languages for information exchange based on all capabilities and needs, including but not limited to the operationally motivated information exchange requirements. However, without having the artifacts describing the conceptualization available, this is not possible, which connects the following sections with the ideas of layered models in support of conceptual modeling described in this section.

3 USING CONCEPTUAL MODELING FOR SYSTEM SELECTION

The conceptual models and layered approaches can be supplied to support the selection of available systems – or

services – based on their conceptual specifications. In this section, we will first describe the formal process to identify the information exchange aspects and will afterwards add some additional remarks.

It should be pointed out that we will focus on the mathematics for information exchange in the following section. Similar formalisms are ultimately needed for all aspects of interoperability, in particular processes and axioms for assumptions and constraints. We present the data view only as a necessary first step.

3.1 Information Exchange Requirements, Needs and Capabilities

When an organization or enterprise has some goal in mind, it is likely to be supported by some conceived operation that will achieve the goal. To achieve this goal, some rigor and methodological approach must be made to ensure that participating systems towards realizing the operation can be federated. The first step is for the operation itself to be described by a conceptual model defining it as a system. Following the layered definitions from section two, several requirements for this conceptual model can already be seen. Following Brachman (1979), the model needs only progress as far as the logical layer, given that the Robinson (2008) definition of conceptual model grants that it is not necessarily implemented. Treating the operation as a system for purposes of conceptual modeling has implications for the identified elements of a conceptual model. The *input* to the operation would be the pre-conditions, the *output* would be the results of the operation, the *assumptions* would provide the context and framework for the operation, the *content* would identify individual actions making up the operation, and the *simplification* of the model would identify what must be addressed by the actions, and what can safely be ignored. The conceptual model of the operation should be addressed as being crucial to the success of satisfying the implementation of the operation. This follows after Federici, et al. (2006), Wache et al. (2001), and Gnägi et al. (2006) as described in the earlier section.

Once the operation is described with a conceptual model, that model can be used to determine which contents of the operation will be accomplished by which systems. Each of the candidate systems should themselves be described by conceptual models, describing the system at least to the logical level, as supported by Brachman (1979). From the information presented in these conceptual models – the operation to be supported as well as the systems available to support the operation –, selection of the systems that accomplish the actions of the operation can follow. It is likely that no single system will represent all of the actions, resulting in coverage of the operation by several systems. Figure 1 exemplifies this process.

On the left side of the figure, the operational concept of the operation to be supported is captured in form of

processes that are connected by data flows. On the right side, systems with matching concepts are used to show how the desired operation can be covered. It should be pointed out that the method described here is not only applicable to identify supporting system, but that it can also be used to identify gaps and specify the needs accordingly, as the main elements of the conceptual model for a system filling the gap are a subset of the conceptual model of the operation to be supported.

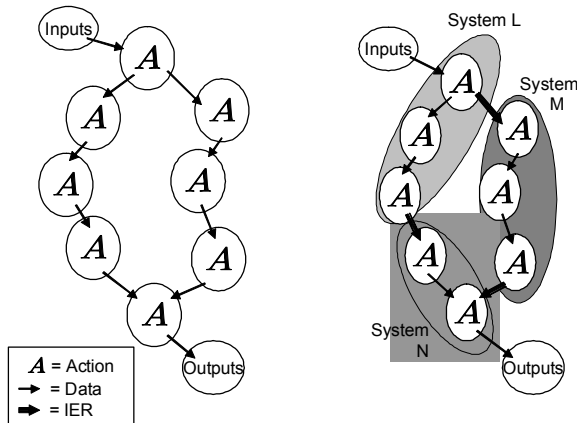


Figure 1: Conceptual Models

Once the systems are identified and mapped to the operation, the *Information Exchange Requirements* (IER) follow by evaluating the mapping: Wherever the system boundary is crossed by the information exchange flow, the results (or output) of one action lead to pre-conditions (or input) to the next action. This information describes the operationally motivated IER. The source system must be able to produce these data and the target system must be able to understand these data. These steps result in identifying the *Information Exchange Needs* (IEN) and *Information Exchange Capabilities* (IEC) with some additional rigor. This requires to address all aspects of the conceptual models to ensure the correctness of the mapping.

- First, the *objectives* determine what the contribution to the operational goal of the system is, and therefore, whether or not it is suitable to provide the information required – evaluating this is an exercise in affirming that the results of the first step (identifying the systems to be part of the information exchange) satisfy the operational need for exchange.
- Second the *inputs* and *outputs* give an enumeration of the capabilities for each system to exchange.
- Finally, the *content*, *assumptions*, and *simplifications* determine the intended use, resolution, scope, and structure of the inputted and outputted data, as well as how that data will be manipulated internally.

By following the operational need and then following the conceptual model of the systems involved, the final steps can be accomplished by identifying all of the information output capabilities and input requirements that the systems will use for exchange. We will define all three terms more formally in section 4.

In section 4, we will also develop a mathematical model for the information exchange (*input* and *output*) and show, how this model can be used in support of interoperation description and prescription. We need similar mathematical model for all aspects mentioned here: *objectives*, *content*, *assumptions*, and *simplifications*. In this paper, we only focus on the information exchange part as an example. More research and discussion are necessary.

3.2 Conceptual Alignment

Once the information exchange is being addressed in order to satisfy IER for the operation, it is worth reconsidering some of the layered models presented in the earlier section in support of higher levels of interoperation. By establishing the IEC and IEN based on the conceptual models of the systems, the conceptual alignment of the systems represented by those models can be assured. This is identified within the LCIM as the highest level of interoperability: conceptual interoperability. The LCIM demonstrates that for conceptual interoperability to be realized, all supporting levels must be realized. Therefore, IEC and IEN should be aligned in such a way so that syntactic, semantic, pragmatic, and dynamic levels of interoperability are supported. Tolk et al. (2006) show that this demonstrates IEC and IEN alignment in light of common protocols (syntactic interoperability), common data models (semantic interoperability), common workflow models (pragmatic interoperability), and common execution models (dynamic interoperability). Only by such a multiple-layered approach to alignment between IEC and IEN can the conceptual alignment be fully realized. King (2007) introduced the term “conceptual linkage” to include in particular conceptual artifacts, such as assumption and constraints.

Using the example shown in Figure 1 it can be seen that IER have been established between *system L* and *system M*, between *system L* and *system N*, and finally between *system M* and *system N*. Considering the first of these IER, it can be seen that the IEC of *system L* must be aligned with the IEN of *system M*. In order to fully realize conceptual interoperability for this IER, the following must be aligned between the IEC of *system L* and the IEN of *system M*:

- protocol of exchange (syntactic interoperability is satisfied by having an aligned means of exchange)
- data model (semantic interoperability is satisfied by having an aligned method of identifying data)

- workflow model (pragmatic interoperability is satisfied by having an aligned method for representing processes)
- execution model (dynamic interoperability is satisfied by having an aligned method for representing context)
- conceptual model (conceptual interoperability is satisfied by having an alignment of concepts, abstractions, relations and roles)

Pair wise comparison for all sets of IER can be supported and even partly conducted by intelligent agent, as objectives, inputs, outputs, content, assumptions, and simplifications are captured in form of data understandable to such intelligent decision support technologies.

4 MATHEMATICAL MODEL FOR INFORMATION EXCHANGE

So far, the recommended solutions have been abstract. In this section, we will give a concrete example on how to derive mathematical expressions that can be used to directly support the unambiguous modeling of data and underlying concepts. The research team of the Virginia Modeling Analysis and Simulation Center of Old Dominion University started with data, as data is the current focus of most interoperability solutions.

Interoperability is defined as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged” [IEEE90]. Before a federation of systems can be setup, it is therefore important to capture the information that needs to be exchanged conceptually. As motivated in section 3, we define the information thus captured as the *information exchange requirements (IER)* of a federation. Based on the IER, candidate systems are selected as possible federates. In order to evaluate to suitability of a system for a given federation, each candidate system must describe conceptually information it can produce and information it can process. Based on this assessment, we must distinguish between the *information exchange capabilities (IEC)* of a systems (what it can produce) and its *information exchange needs (IEN)*. Given a federation and a set of IER, a system is a federate if its IEC and IEN are a subset of the IER. However, the question of conceptually capturing the IER, IEC, and IEN has yet to be addressed. In practice, information is captured in a logical model.

4.1 Formalizing the Information Exchange

Logical Models and their physical implementations have been extensively studied over the last thirty year and formalized by Codd’s relational model (1970). However, the relational model and its modern counterparts such as the XML based data models as described by Bhargava et al. (2005) fail to formally capture both the underlying assump-

tions that lead to a logical model and its context of validity. In order to formalize information exchange, each system must be described in a way that captures not only entities and their relationships but also the conceptualization of each entity and the context of validity of each relationship. The context of validity of relationships refers to the unique meaning of a relationship as intended by the modeler. While this meaning might be ambiguous, it must be captured unambiguously. It is important to stress this aspect of the conceptual model because failing to capture this information often leads to misalignment between the conceptual model and logical model and/or the logical model and the physical model.

In order to capture the information exchange needs and capabilities of a system as well as the information exchange requirements of a federation, we refer to the ISO/IEC 11179 (Part 3) for metadata registry. This standard uses four components to identify a data element:

1. the conceptual domain captures the results of the conceptualization phase and represents to what concept the data element contributes;
2. the data element concept, which is the property of the concept that is represented by the data element;
3. the value domain, which is the range of possible values to express the data element concept; and
4. the data element itself.

Alternatively, we introduce the following definitions:

- *Properties* are the specifying characteristics of an entity. They are representing the data element concept information of ISO/IEC 11179.
- *Property values* are the allowed values for a specifying characteristic. This is the information captured in the value domain as specified in ISO/IEC 11179.
- *Propertied concepts* are a collection of specifying characteristics for an entity or concept in the domain of knowledge. It may be of help to think about entities as the “things” that can be differentiated in the represented world view. They are captured in the conceptual domain specified by ISO/IEC 11179.
- *Associated concepts* are semantic entities in which data comprising more than one propertied concept are given in a common context. This concept is given by associations connecting the respective propertied concepts. It is of help to think about associated concepts as more complex elements, such as plans or list with more than one “thing” in it.
- The *context of data* is the specification of a data element’s property, propertied concept, and associated concept.

In order to formalize information exchange, we introduce the following relations.

Let S be the non empty, finite and countable set of all propertyed concepts and P the non empty, finite and countable set of properties. We define a relation “*hasProperty*” as the set of ordered pairs: $hasProperty = \{s \in S, p \in P / s \text{ has property } p\}$. (1)

Let us now consider V as a non-empty, finite and countable set of property values, a relation “*hasProperty-Value*” is defined as the set of ordered pairs: $hasProperty-Value = \{p \in P, v \in V / p \text{ has value } v\}$.(2) In addition, we define a non-empty, finite and countable set of value domains D and a relation $hasDomain = \{v \in V, d \in D / v \in d\}$.(3)

Finally, let X and Y be two sets of propertyed concepts, we define a relation “*isAssociatedWith*”: “*isAssociatedWith*” = $\{x \in X, y \in Y / x \text{ has a semantic link with } y\}$. (4)

A semantic link in this case refers to any form of semantic relation including categorizations, requirements, etc...The study of the nature of this link lies in the domain of ontology and therefore is outside the scope of this paper.

Using the relations defined earlier, we can formally describe a data element v that is susceptible of being exchanged as the quadruple $(s, p, v, d) \ s \in S, p \in P, v \in V, d \in D$ such that $s \text{ hasProperty } p, p \text{ hasValue } v$ and $v \text{ hasDomain } d$. Otherwise stated, the set of all data elements susceptible of being exchanged is the subset of the Cartesian product $S \times P \times V \times D$ that obeys the first three relations defined above. The set of data elements in context is formally defined as the subset of the Cartesian product $S \times P \times V \times D$ that obeys all four relations. This subset represents the IER or IEN of a system. It can also represent the IER of a federation.

4.2 Application Example

As a simple example, consider figure 2 showing how two data elements can be specified using the formalism described here.

The set $X = \{\{Refrigerator\}, \{Temperature\}, \{25, Celsius\}, \{\{1, 2, \dots, 99\}, \{Celsius, Fahrenheit\}\}\}$ captures the statement “*The temperature of the refrigerator is 25 degrees Celsius.*” In addition, the value domain indicates that the only other alternative for describing the element is “*Fahrenheit.*” Similarly, the set $Y = \{Appliance, Make, GE, \{Philips, GE\}\}$ captures the statement “*The make of my appliance is GE.*” Taken separately these two sets fail to capture the complete picture which is “*The appliance is a GE refrigerator which has a temperature of 25 degrees Celsius.*” The set $Z = X \cup Y$ captures the previous sentence. It is worth noting that other valid subsets do exist for this example.

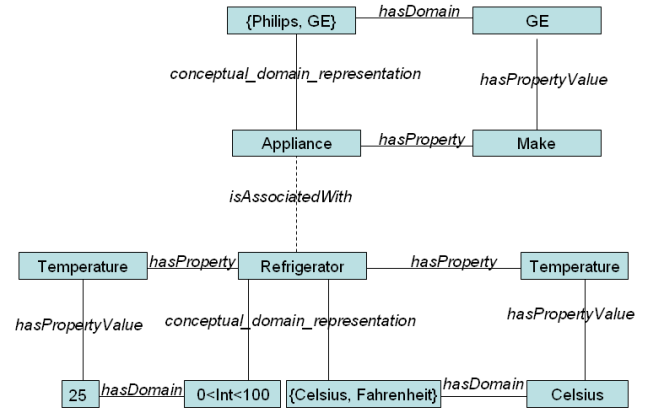


Figure 2: Formal Representation of Data Elements

4.3 Some Implications

In practical terms the quadruple defined in this paper serves as metadata that must be exchanged at least once, normally during initialization. At run time, systems need only exchange data (v) and reference the set to which it is a member. The message exchanged will therefore be constituted of a header that contains information about the set membership (name, tag, or symbol) and the body will contain the datum itself. If we let $H = \{S, P, D\}$ and $B = \{V\}$, we can define a message m being exchanged as $m = h \times b$ with $h \in H, b \in B$. The set M of all messages exchanged in a federation is $M = H \times B$.

Based on this formalism, we can define the IER of a federation as the set M , the IEC of a system as the subset of M that it can generate and IEN the subset of M it can consume. This formalism can be used to define a self-organizing federation language following the algorithm described in the next section.

One of the phrases often used in the context of web-based systems is *mediation*. The mathematical model shows some ideas of direct practical value:

- If two values v are identical, they are only conceptually equivalent if and only if their quadruples $S \times P \times V \times D$ are identical as well.
- If two quadruples $S \times P \times V \times D$ are identical beside the value v , and a reversible function exists that transforms v into v' , than a semantically lossless mediation exists between both representation.

Several other implications regarding resolution, scope, and structure are directly exist that can be directly derived from this model. Some of them will be mentioned in the following section.

As mentioned at the beginning of this section, the research team started

5 TOWARDS SELF-ORGANIZING FORMAL FEDERATION LANGUAGES

The main challenges to be addressed by formal federation languages are the same that every federation developer faces: as models are purposeful abstractions of reality, the viewpoints of different models differ regarding

- the resolution chosen by the models for the properties that are exposed (multiple resolution),
- the concepts that are characterized by the properties that are modeled (multiple structure),
- the scope defined by the union of all concepts and their properties (multiple scope), and
- the underlying paradigms and metaphors used to model the concepts (multiple paradigms).

The recommended solutions to deal with these challenges, such as formulated among other by Zeigler and Oren (1986), Reynolds et al. (1997), Davis and Tolk (2007), and Yilmaz et al. (2007), remain valid and are part of the applied body of knowledge enabling the self-organization described in this section. They need to be formalized in order to be supportable automatically. Many practical application ignore this problems and assume that two models use the same scope, structure, and resolution, only because the model the same aspects of the real world. The mathematical model helps not only to identify potential misalignments, it also helps to solve some challenges. We used the Cartesian product $S \times P \times V \times D$ to define the propertied concepts (S) in a given domain (D) exposing properties (P) with assigned values (V). This reflects the metadata needed for a data element as also identified in ISO/IEC11179.

- If S and D are identical, but the P and V of one model are aggregates of the other model (where aggregation means that several values of one model are combined into one or more values of the other model in a process that consumes information – such as building of the sum or the mean), we observe *multiple resolution*.
- If P and V are identical, but they are combined into different S and D , we observe *multiple structure*.
- If two models share only a subset of S and D , but for those shared S and D all P and V are identical, we observe *multiple scope*.

These observations help us to define mathematical expressions that can be used to categorize potential misalignments, including identifying mixed cases (such as multiple scope and resolution, if only parts of S and D overlap, and for those, aggregates need to be defined).

As pointed out before, current simulation interoperability standards do not address these challenges adequately. By supporting standardized protocol data units for the information exchange (IEEE 1278) or agreeing on a common federation object model (IEEE 1516), these standards introduce another model to the federation, that can only be a compromise among the federates. As mentioned at the end

of section 4.1, such a “common” federation model introduced its own quadruple $S \times P \times V \times D$. Currently, this mathematical model is the result of a conceptualization process as described before. Our recommendation is to replace this process with self-organizing formal federation languages that use the artifacts capturing the conceptual modeling decisions.

So far, we used the conceptual model of the operational concept to be supported to guide the development of the IER. Each data set captured in the conceptual model of the operational concept supported by the chosen systems itself can be described by the $S \times P \times V \times D$. One possible interpretation of the quadruple is to capture all possible expressions in form of an XML schema. Another possibility is to define a generative grammar with production rules resulting in the enumeration of all elements of this set. Another possibility is to use the sets to define an object model to be used within the federation, etc. In summary, the union of all these sets makes up all sentences that need to be exchanged within the federation in order to support the operational concept to be supported. The conceptual artifacts developed when defining the actions and data are the common nominator and give the meaning to the terms. Therefore, we use S , D , and P to define the elements of the common language by deriving them from the operational concept and using the results on mediation described at the end of section 4 to map the values of the language to the different representations in the participating systems.

However, real self-organization should not be limited by the operational concept to be supported. Using the ideas presented so far, the information exchange must not be limited to the interfaces identified in Figure 1. However, we need to assume – and this assumption is not trivial – that descriptions of domains (D) and propertied concepts (S) are unambiguous and machine understandable, which can be supported by capturing the artifacts of the conceptual modeling phase in machine understandable form. If this is the case, each interface of each system participating in the federation can be unambiguously described by two characteristic quadruples $S \times P \times V \times D$ describing all sets that can be provided as *Information Exchange Capability* of this interface as well as all sets that can be consumed as *Information Exchange Need* for this interface. Starting with these descriptions of each interface as a data producers as well as a data consumer, the elements of the describing quadruples can be mapped, e.g.

- All sets with identical $S \times P \times V \times D$ in the IEC and IEN of two different systems can be exchanged and are valid sentences in the language.
- All sets with identical domains, propertied concepts and properties of which the values can be functionally be mapped to each other can be mediated to valid sentences of the language.
- Multiple scope sets with no challenges in resolution or structure contribute the overlapping sets to the common language (if no axioms or rules require the exchange of all information elements).

These examples are just a minor subset of resulting theorems that can be used to self-configure the information exchange. As pointed out before, the biggest assumption is that we have the information for all elements of the quad-

rule available assuring that we identify if two things are the same (even if they may have been named differently in the participating systems) or if two things are different (even if they are called the same in two participating systems). Such results are only possible due to rigorous mathematical models based on conceptual modeling capturing all important results in machine understandable artifacts.

6 SUMMARY

The main findings and recommendations of our research conducted of several years are that

- conceptual modeling must be conducted based on engineering principles resulting in artifacts,
- rigorous mathematical modeling is necessary to capture this artifacts,
- composition of systems or services into systems of systems or service-oriented architectures must use the results of the conceptual modeling phase of the development phase, or conceptual misalignment is unavoidable,
- in order to support the composition with self-organization or artificial intelligence means, such as intelligent agents, conceptual modeling artifacts are required.

The paper only gives details for information exchange challenges. To describe a system, not only data is needed. In addition, the participating processes are needed as well. Furthermore, axioms for data as well as for processes need to be captured. All three elements describing a system – data, processes, and axioms – can have different scope, structure, and resolution in different systems, leading to some already known challenges. Again, conceptual model elements can help to identify such areas of possible misalignment. To support composition of models, we need mathematical models for data, processes, and axioms and the interplay between them. This models will help to find out which artifacts are needed to support composability.

The second application of the ideas described in this paper is to describe information exchange capabilities and needs instead of prescribing information exchange requirements. While the idea of operationally driven IER was appropriate for the traditional system development paradigm, it is only limited applicable for net-centric environments in which legacy systems are to be reused in support of already available functionality. In such environments, efficient methods for mapping and mediation are needed. The definition of a common exchange model that every service has to use as a mandate is no longer feasible.

The current research results are just a first hint at what may become possible if we capture all aspects of conceptual models – *objectives, inputs, outputs, content, assump-*

tions, and simplifications – in rigorous mathematical model to support really composable services and systems based on self-configuring languages to exchange information and self-orchestrating services providing the available functionality to the user.

ACKNOWLEDGMENTS

Significant parts of the underlying research work were conducted supported by the US Joint Forces Command (JFCOM), the US Army Test and Evaluation Command (ATEC), and the Program Executive Office (PEO) Soldier.

REFERENCES

- Bhargava, P.; Rajamani, H.; Thaker, S.; Agarwal, A. 2005. XML Enabled Relational Databases. Report. The University of Texas at Austin, Texas
- Brachman, R.J. 1979. On the Epistemological Status of Semantic Networks. In *Associative Networks: The Representation and Use of Knowledge by Computers*, ed. N.V. Findler, pp. 3-50. Academic Press New York
- Codd, E.F. 1970. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM* 13 (6): 377–387.
- Davis, P.K. and A. Tolk. 2007. Observations on new developments in composability and multi-resolution modeling. *Proceedings of the 2007 Winter Simulation Conference*. eds. S. G. Henderson, B. Biller, M.-H Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 859-870. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Dobrev, P., O. Kalaydjiev, and G. Angelova. 2007. From Conceptual Structures to Semantic Interoperability of Content. *LNCS Vol. 4604*, 192-205, Springer-Verlag Berlin Heidelberg
- Federici, M.L., S. Redaelli, and G. Vizzari. 2006. Models, Abstractions and Phases in Multi-Agent Based Simulation. *Proceedings CEUR Workshop 204*: 144-150
- Gnägi, H.R., A. Morf, and P. Staub. 2006. Semantic Interoperability through the Definition of Conceptual Model Transformations. *Proceedings 9th AGILE Conference on Geographic Information Science*, 128-138
- IEEE 1278. Standard for Distributed Interactive Simulation
- IEEE 1516. Standard for Modeling & Simulation High Level Architecture
- Keating, C., R. Rogers, R. Unal, D. Dryer, A. Sousa-Poza, R. Safford, W. Peterson, and G. Rabadi. 2003. System of Systems Engineering. *Engineering Management Journal* 15 (3): 36-45
- King, R.D. 2007. Towards Conceptual Linkage of Models and Simulations. *Proceedings Fall Simulation Interoperability Workshop*, IEEE CS Press

- Mascari, J.-F. and G.A. Cavarretta. 2007. Complex adaptive services. *Business Process Integration and Management*, 2 (1): 3-8
- McCarthy, J. and P.J. Hayes. 1969. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine Intelligence 4*, ed. D. Michie and B. Meltzer, 463-502. Edinburgh University Press
- Oren, T., N. Ghassam-Aghaee, and L. Yilmaz. 2007. An Ontology-based Dictionary of Understanding as a Basis for Software Agents with Understanding Abilities. *Proceedings 2007 Spring Simulation Multiconference*, 25-29. IEEE Press
- Page, E.H., R. Briggs, and J.A. Tufarolo. 2004. Toward a Family of Maturity Models for the Simulation Interconnection Problem. *Proceedings IEEE Spring Simulation Interoperability Workshop*, IEEE CS Press
- Reynolds, F. P., A. Natrajan, and S. Srinivasan. 1997. Consistency maintenance in multiresolution simulation. *ACM Transactions on Modeling and Simulation*, 7 (3): 368-392
- Robinson, S. 2008. Conceptual modelling for simulation Part I: definition and requirements. *Journal of the Operational Research Society*, 59: 278-290
- Tolk, A. 2006. What comes after the Semantic Web? PADS implications for the Dynamic Web. *20th ACM/IEEE/SCS Principles of Advanced Distributed Systems*, 55-62, IEEE CS Press
- Tolk, A., and S.Y. Diallo. 2008. Model-based Data Engineering for Web Services. In *Evolution of the Web in Artificial Intelligence. Environments*, eds. R. Nayak et al., SCI 130, 137-161, Springer, Berlin Heidelberg
- Tolk, A., S.Y. Diallo, C.D. Turnitsa, and L.S. Winters. 2006. Composable M&S Web Services for Net-centric Applications. *Journal for Defense Modeling & Simulation*, 3 (1): 27-44
- Wache, H., T. Vogege, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner. 2001. Ontology-based Integration of Information – a Survey of Existing Approaches. *Proceedings of the IJCAI-Workshop Ontologies and Information Sharing*, 108-117, Seattle, WA
- Yilmaz, L., A. Lim, S. Bowen, and T. Oren. 2007. Requirements and design principles for multisimulation with multiresolution, multistage multimodels. *Proceedings of the 2007 Winter Simulation Conference*. 823-832. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Zeigler, B.P., and P.E. Hammonds. 2007. *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*. Academic Press New York
- Zeigler, P. B. and T.I. Oren. 1986. Multifaceted, multi-paradigm modeling perspectives: Tools for the 90's. *Proceedings of the 1986 Winter Simulation Conference*. 708-712.

AUTHOR BIOGRAPHIES

ANDREAS TOLK is Associate Professor for Engineering Management and Systems Engineering at Old Dominion University (ODU). He is also a Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC). He holds a M.S. in Computer Science (1988) and a Ph.D. in Computer Science and Applied Operations Research (1995), both from the University of the Federal Armed Forces of Germany in Munich. He is a member of SCS and SISO. His e-mail address is <atolk@odu.edu>.

SAIKOU Y. DIALLO is a Project Scientist and a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Engineering (2003) and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of Model Based Data Engineering and web services for M&S applications. His e-mail address is <sdiallo@odu.edu>.

CHARLES D. TURNITSA is a Project Scientist and a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Science (1991) from Christopher Newport University (Newport News, Virginia), and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of dynamic and fractal ontology models for M&S interoperability. His e-mail address is <cturnits@odu.edu>.