Spring 2010

Towards a Formal Theory of Interoperability

Saikou Y. Diallo
Old Dominion University

Recommended Citation
Diallo, Saikou Y.. "Towards a Formal Theory of Interoperability" (2010). Doctor of Philosophy (PhD), Dissertation, Computational Modeling & Simulation Engineering, Old Dominion University, DOI: 10.25777/tk13-2064
https://digitalcommons.odu.edu/msve_etds/27

TOWARDS A FORMAL THEORY OF INTEROPERABILITY

by

Saikou Y. Diallo

B.S. Computer Engineering, May 2003, Old Dominion University
M.S. Modeling and Simulation, May 2006, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirement for the Degree of

DOCTOR OF PHILOSOPHY

MODELING AND SIMULATION

OLD DOMINION UNIVERSITY
May 2010

Approved by:

Andreas Tolk (Director)

Rasha Morsi (Member)

John Sokolowski (Member)

Eric Weisel (Member)

ABSTRACT

TOWARDS A FORMAL THEORY OF INTEROPERABILITY

Saikou Y. Diallo
Old Dominion University, 2010
Director: Dr. Andreas Tolk

This dissertation proposes a formal theory of interoperability that explains 1) what interoperability is as opposed to how it works, 2) how to tell whether two or more systems can interoperate and 3) how to identify whether systems are interoperating or merely exchanging bits and bytes. The research provides a formal model of data in M&S that captures all possible representations of a real or imagined thing and distinguishes between existential dependencies and transformational dependencies. Existential dependencies capture the relationships within a model while transformational dependencies capture the relationships between interactions with a model. These definitions are used to formally specify interoperation, the ability to exchange information, as a necessary condition for interoperability. Theorems of interoperation that capture the nature and boundaries of the interoperation space and how to measure it are formulated. Interoperability is formally captured as a subset of the interoperation space for which transformational dependencies can be fulfilled. Theorems of interoperability that capture the interoperability space and how to measure it are presented.

Using graph theory and complexity theory, the model of data is reformulated as a graph, and the complexity of interoperation and interoperability is shown to be at least NP-Complete. Model Based Data Engineering (MBDE) is formally defined using the model of data introduced earlier and transformed into a heuristic that supports interoperability. This heuristic is shown to be more powerful than current approaches in that it is consistent and can easily be verified.

This dissertation is dedicated to my beloved family all over the world.

ACKNOWLEDGEMENTS

There are many people who have contributed to the successful completion of this thesis. I extend many, many thanks to all my colleagues at VMASC/ODU for their insights discussions and enlightening comments. I would like to thank my friends at the fireside chat Jose Padilla, Ipek Bozkurt, and many others for their patience, help and reviews during the process. A friendship that will last forever was born from this labor of love. Thanks to Mfikeyi for always pushing me and giving me the motivation to be an example for her. I hope you go farther than I ever will. Thanks to my parents for the example they set for me every day, their untiring love, and for always believing in me and pushing me to always go for my dreams and not settle for less. The untiring efforts, guidance and support of my major advisor Dr. Andreas Tolk deserve special recognition. As a mentor and friend he guided this dissertation with a masterful and wise hand. The lessons I learned from watching him every day will stay with me for the rest of my life. Special thanks to Dr John Sokolowski for making it possible to attain this goal not only as a committee member but also as a part of the VMASC leadership that encourages young researchers to grow and expand. Thanks to Dr Mike McGinnis for giving me the support needed to finish this work. Finally a special thanks to my many sponsors especially JFCOM for the chance they provided to learn about the issues described in this work which allowed me to provide a theory of interoperability. I am grateful to all my committee members for their guidance and generosity with their time and advice. I would like to mention Dr. Roland Mielke who is the best teacher I have ever had at any level for the example that he set, and for sparking an interest in learning, and M&S.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

1 INTRODUCTION

1.1 Problem Statement

Interoperating heterogeneous systems remains a great challenge in Modeling and Simulation (M&S) due to the nature of model based systems, the existence of a variety of engineering methods and the lack of a unified formal specification of interoperability. Interoperability is understood as the ability of two or more systems to exchange information and use the information thus exchanged (IEEE, 1990)[1]. Interoperable solutions are highly sought after not only in M&S but also in the business and military world. Benefits of those solutions include adaptability and reuse which ultimately results in savings in time and money. Interoperability has been identified as a necessary step for composability which is the ability to use and reuse components (Weisel, 2004). As a result, some issues of composability cannot be answered until they are addressed in interoperability. A formal specification is therefore needed to identify, classify and solve problems that are inherent to interoperability. Such a specification would make it possible to assess areas in science where identical problems have been solved and would allow M&S practitioners to consistently predict the time, level of effort and cost of interoperating N systems. Current approaches to interoperability focus on both the technical aspects through the development of technical standards and frameworks and the semantic aspects through the development of common reference models (CRM), ontology or federated schemas to capture the semantics of data.

The term interoperability is used to mean different but closely related things depending on the application area. Nations and companies are ready and willing to invest in “interoperable solutions” even though it is not clear what that really means or what an interoperable solution represents. However, it is clear that just like composition and validation, interoperability is more a practice than a science. As a result it is difficult to come to a consensus on what the problems inherent to interoperability are, how they are categorized and whether they can be solved. Issues such as multi resolution, multi scope and multi structure that are not central in software interoperation take on a new

[1] The dissertation uses the APA style for citation.

dimension in model based interoperation because each system is a purposeful abstraction of reality (Davis & Anderson, 2004). Consequently solutions dealing with technical issues such as connectivity or syntactical alignment while sufficient for software and hardware interoperability fall short when it comes to model based systems (Hofmann, 2004).

One of the main roadblocks to interoperability for model based systems is the issue of semantic inaccessibility. As stated in (Benjamin, Akella, & Verna, 2007):

"*The semantic rules of the component simulation tools and the semantic intentions of the component designers are not advertised or in any way accessible to other components in the federation. This makes it difficult, even impossible, for a given simulation tool to determine the semantic content of the other tools and databases in the federation, termed the problem of semantic inaccessibility. This problem manifests itself superficially in the forms of unresolved ambiguity and unidentified redundancy. But, these are just symptoms; the real problem is how to determine the presence of ambiguity, redundancy, and their type in the first place. That is, more generally, how is it possible to access the semantics of simulation data across different contexts? How is it possible to fix their semantics objectively in a way that permits the accurate interpretation by agents outside the immediate context of this data? Without this ability—semantic information flow and interoperability—an integrated simulation is impossible".*

The challenge is therefore to make data in systems semantically accessible to other systems so that they make use of it. The state of the art in interoperability currently deals with the symptoms of this problem but do not provide a cure. At the root, there is a lack of a formal specification of interoperability or a formal theory of interoperability that separates problems from their symptoms, identifies problems that have been solved in other arenas and need not be readdressed and formulates theorems and well formed formulas that hold true in the general case.

1.2 Thesis Statement

Engineering methods such as Data Engineering and MBDE (Tolk, Diallo, King & Turnitsa, 2009) attempt to answer the question of semantic inaccessibility but focus on best practice recommendations instead of formal models so that in their current form they are insufficient to serve as the basis for a general theory of interoperability. Other approaches such as using protocols, standards, common reference models and ontology are all facets of the same solution, namely MBDE. In fact, this dissertation will show that they are all equivalent and differ only in the type of artifact and algorithm used. The goal of this dissertation is to provide the first step towards a formal theory of interoperability that focuses on machine to machine interoperability in a distributed environment. The theory will explain interoperability and make systems more semantically accessible by identifying the requirements for interoperability. The formalism will provide a basis for comparing existing frameworks and approaches that support interoperability. While the findings in this dissertation are focused on machine to machine interoperability in distributed environments they are applicable to other areas of interoperability.

1.3 Research Question

A review of the current state of the art shows that interoperability has many working definitions. The lack of a formal definition of interoperability and the existence of multiple definitions for related terms such as model and simulation indicate that the community is not exactly on the same page when it comes to making systems interoperate. The consensus based on a review of M&S literature shows that interoperability happens in levels and is characterized by the exchange of useful information between simulated models. There are different approaches to interoperability and they are all equivalent in that they result in a common language that is either prescribed or emerges from an agreement. As a result, the problem of semantic inaccessibility is not completely solved by these approaches. The research question is:

How do we formally specify a theory of interoperability that explains what interoperability is?

The proposed answer presented in this work is a formal specification of interoperability to:

1. explain what interoperability is;
2. identify the requirements for interoperability;
3. provide a basis for comparing interoperability frameworks and approaches.

Each of these steps is developed into subsets that are discussed in the next section.

1.4 Approach

In order to create a theory of interoperability, this dissertation first reviews the state of the art to establish how interoperability is defined and classified. This review establishes the characteristics of interoperability and serves as the basis for the questions that the theory should be able to answer. These questions are concerned with what interoperability is and how to recognize it. A review of current approaches to interoperability provides insights into how interoperability is practiced. This review provides an additional set of questions focused on how interoperability works and how to measure it. The reviews provide a basis for determining the usefulness and applicability of the theory.

Since interoperability is concerned with the meaningful exchange of information, it is essential to formally define what a referent is and how to capture it in a data model. A formal definition of a data model in M&S serves as the basis for the theory. Rather than redefine what a model is, this dissertation reviews relevant related domains in which the nature of data models and information exchange has been studied. The review focuses on formal specifications to the exception of all others because the goal of this dissertation is to provide a formal theory and therefore must be rooted in formal theories. If the review shows that an existing theory satisfactorily explains what interoperability is, it is directly used to formulate theorems of interoperability if they are not already formulated. The theory in question is further used to formulate MBDE and compare current approaches to interoperability. If such a theory does not exist, elements that are formal and relevant to M&S are captured and used as a basis for the new theory of interoperability.

This dissertation uses Set Theory, a branch of Mathematics or Logic that is concerned with the study of sets which are defined as collections of objects(Hein, 2002), to formally define a referent (real or imagined thing) and a model of a referent. Using the properties of sets and logic, interoperability is formally defined along with its characteristics. A deductive reasoning system is applied to derive theorems of interoperation and interoperability. These theorems are used to investigate how interoperability works. **Error! Reference source not found.** shows the details of the approach used in this work. Each box represents a major milestone from which the next step is built and contains the body of work being examined or to which the dissertation contributes.

The idea of critical reading implies not only a classical review of the literature but one in which each work reviewed is also evaluated with respect to how it contributes in answering the research question. This evaluation includes classifying similar works, identifying general trends and assessing relevance to the domain of M&S.

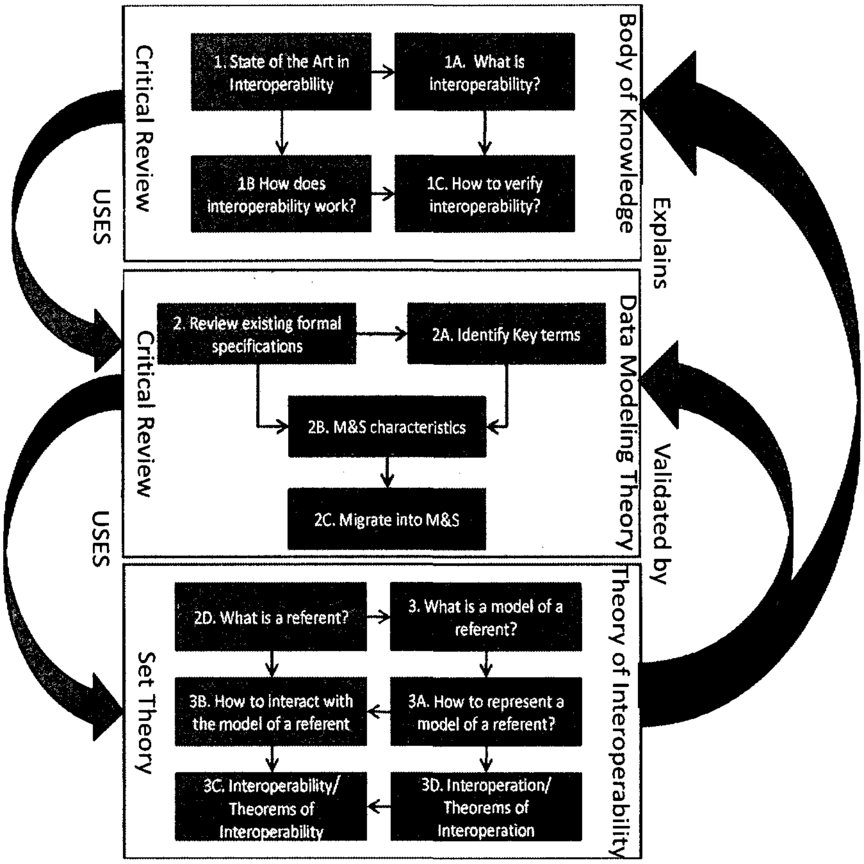

Critical Review
1. State of the Art in Interoperability
1A. What is interoperability?
1B How does interoperability work?
1C. How to verify interoperability?
Body of Knowledge
USES
Explains
Critical Review
2. Review existing formal specifications
2A. Identify Key terms
2B. M&S characteristics
2C. Migrate into M&S
Data Modeling Theory
USES
Validated by
Set Theory
2D. What is a referent?
3. What is a model of a referent?
3B. How to interact with the model of a referent
3A. How to represent a model of a referent?
3C. Interoperability/ Theorems of Interoperability
3D. Interoperation/ Theorems of Interoperation
Theory of Interoperability

Figure 1: An approach to Theory Building

Critical reading establishes the basis from which to generate the theory but it also serves as the basis for evaluating the theory. While the theory itself is developed using Set Theory, it must not contradict the formal theories from which it was derived. That is to say that a new theory of interoperability must be at least as powerful as the theories from which it borrowed while being capable of answering the fundamental questions for which it was developed. For example, the specification of a data model in M&S in section 3 borrowed elements from Data Modeling Theory and therefore must be at least as expressive as formal specifications found in that body of work. Further, the data model must have additional usefulness in the development of a theory of interoperability.

Once a theory is developed and its consistency is demonstrated, its usefulness must be tested. The usefulness of the theory is determined not only by its ability to explain the phenomenon it was developed to support but also by the additional insights into the subject matter. Figure 2 shows how the theory developed here is used to answer additional questions about interoperability. These questions were not derived from the initial critical review of the Body of Knowledge but rather were identified by the author as new contributions to the field of M&S. The set of questions were derived prior to the development of the theory as a test for its power and usefulness.

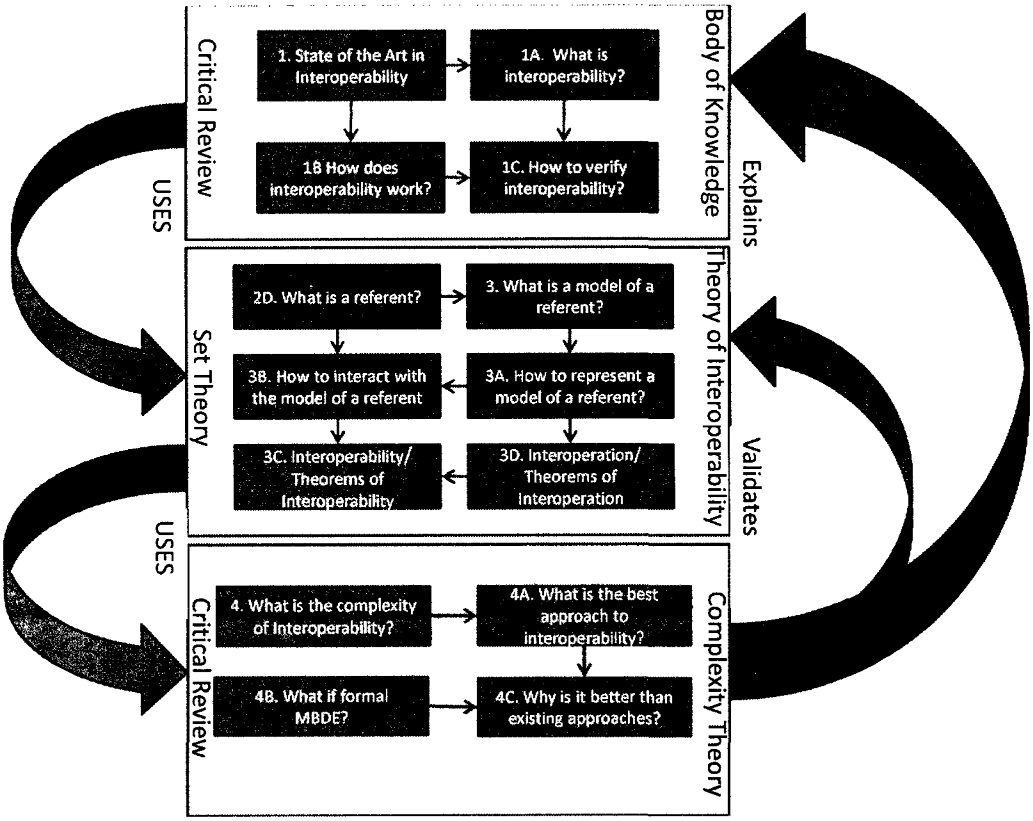

Critical Review
1. State of the Art in Interoperability
1A. What is interoperability?
1B How does interoperability work?
1C. How to verify interoperability?
Body of Knowledge
Explains
USES
Set Theory
2D. What is a referent?
3. What is a model of a referent?
3B. How to interact with the model of a referent
3A. How to represent a model of a referent?
3C. Interoperability/ Theorems of Interoperability
3D. Interoperation/ Theorems of Interoperation
Theory of Interoperability
Validates
USES
Critical Review
4. What is the complexity of Interoperability?
4A. What is the best approach to interoperability?
4B. What if formal MBDE?
4C. Why is it better than existing approaches?
Complexity Theory

Figure 2: Testing the Usefulness of a Theory

Using Complexity Theory (Fortnow & Homer, 2003), the dissertation evaluates the complexity of deciding whether two or more systems are interoperable in order to compare current approaches to interoperability. Formal MBDE, a heuristic to describe and capture the interoperability space between models is presented. The heuristic is compared to current approaches and the dissertation explains why it is better in terms of consistency and verifiability.

The result of the overall process is a useful theory of interoperability that explains what interoperability is, how it works and how to verify it. This theory is used to study the computational complexity of interoperability and derive a formal heuristic that enables interoperability. The next section provides a summary of the objectives and goals of each section in the balance of this work and shows how they are related.

1.5 Dissertation Guide

Section 2 is a critical review of interoperability in order to establish the common features and expected outcomes as they are currently understood. The review first establishes that interoperability is the ability to exchange information and use the information once it is exchanged. One school of thought holds that interoperability is inherent to systems while another explains that interoperability should be defined with respect to a group of systems. These seemingly contradictory views are explained and reconciled when interoperability is formally defined in section 4.1. The distinction between system and model is discussed in section 2.3.

The notion of information exchange is also debated within the literature, and the review shows that though there is a distinction between data and information, it is not clear whether the exchange of data and its use constitute interoperability. This issue is reviewed in detail in section 2.4 which presents the levels of interoperability as they have evolved over time.

Section 2.3 explores terms related to interoperability namely composability, model and simulation in order to generate a common definitional ground and establish the context of this work. The focus is on data models and the exchange of information between them. This section motivates the need for a review of formal data models in section 3 and ultimately the specification of a data model that represents the general case in section 3.2 and serves as the basis for the specification of interoperation and interoperability in section 4.

Section 2.4 reviews existing approaches to interoperability and groups them into two broad categories: one that focuses on standardizing terms and relationships into a model for all to use and another that focuses on standardizing the technical exchange of data. These two approaches are shown to be equivalent in section 4. In terms of computational complexity (complexity heretofore) all approaches are shown to be heuristics. The issue with current approaches is that they do not provide a way to verify whether models are interoperable. This issue is also addressed is section 5.

The questions identified in section 1.2 deal with the nature of interoperability (i.e. what is it, how does it function and what are its characteristics) and form the initial set of question the theory should answer. These questions are answered in section 3 and section 4 when a formal specification of data in M&S is presented along with the resulting theorems. Interoperability is separated from interoperation to distinguish between the exchange of information (interoperation) and the use of the information (interoperability).

Section 5 formalizes MBDE using the model specified in section 3.2. MBDE is adapted as a heuristic that enables interoperability. Section 6 summarizes the work and proposes future lines of research. Annex one presents the detailed MBDE algorithms.

In summary, this dissertation introduces the problem domain in section 1 and identifies the gap in section 2 through a review of the state of the art in interoperability. In section 3 a model of data in M&S is formally specified and used to formally define interoperation and interoperability along with their respective theorems. Section 4 shows that, in general, there is no efficient algorithm to determine whether two or more systems are interoperable. In section 5, a heuristic that approximates interoperability between two or more systems is presented and compared with existing heuristics presented in section 2. Section 6 presents a summary and proposes additional topics of research based on the work presented here.

2 LITERATURE REVIEW

Interoperability is an essential aspect of not only M&S but also of Software Engineering and System of System Engineering. The term is frequently used in the literature but is defined differently depending on the domain in which it is applied.

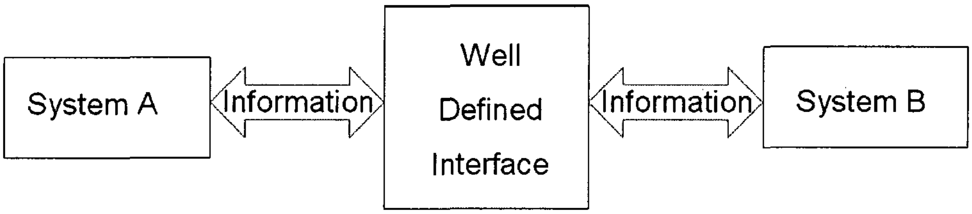


Figure 3: Conceptualized View of Interoperability

Figure 3 shows a conceptualized view of interoperability in which systems exchange information through a well defined interface. Despite the multitude of domains and definitions, there is a common concept of systems or organizations sharing information and cooperating to achieve a given goal. The review presented here shows the state of the art and uses it to identify the requirements for a useful theory of interoperability. The requirements are the necessary questions or issues that the theory must explain before it is able to explain interoperability. The goal of this section is to establish a starting point for a theory of interoperability by first examining definitions of interoperability and establishing whether they can serve as a basis for a first step towards a theory of interoperability. A good starting point must be formal, unambiguous and general in that it covers all aspects of interoperability. If the definitions cannot be used as such, the review will focus on the classifications of interoperability as the next logical starting point. If the classifications cannot be used, the review will investigate the things that are interoperating whether they are called model, system or simulation and attempt to use their descriptions as the starting point. If that is not sufficient, the review will investigate how interoperability is supported in the state of the art in order to determine if it can constitute an initial point for a theory of interoperability. In the end, the review will propose a departing point for a theory of interoperability based on the discussions in this section.

2.1 Definition of Interoperability

Webster's online dictionary defines interoperability as

"the ability to exchange and use information (usually in a large heterogeneous network made up of several local area networks)" (Webster Online, 2008)

The Department of Defense (DoD) defines interoperability as

"the condition achieved among communications-electronics systems or items of communications-electronics equipment when information or services can be exchanged directly and satisfactorily between them and/or their users. The degree of interoperability should be defined when referring to specific cases"(Department of Defense, 2008)

The Institute of Electrical and Electronics Engineers (IEEE) defines interoperability as

"the ability of two or more systems or components to exchange information and to use the information that has been exchanged"(IEEE, 1990)

Another definition of interoperability is proposed by the International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) as

"The capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units"(ISO/IEC, 2003)

While these definitions are not formal, they emphasize two points that are recurrent in all of the working definitions of interoperability found in the literature:

- *Information Exchange*: Interoperable systems are characterized by their ability to exchange information. The DoD definition stresses the fact that information exchange must be direct which conforms to the view depicted in figure one. It is also clear from these definitions that interoperability is at the system level and more precisely systems that are implemented on a computer. This definition also takes the position that interoperability is a condition that must be achieved which implies that systems are interoperable when they are interoperable. The IEEE definition defines interoperability as inherent to a system (its ability to exchange information) which implies that systems are interoperable if they are interoperable. This distinction is reflected in the way interoperability is solved in the state of the art as will be shown in the next section. The DoD definition also points to a third factor by introducing the notion of a measurable or observable "degree of interoperability" that is determined on a case by case basis. This is an important aspect of interoperability that is discussed in more details in the next section. The ISO definition focuses solely on the technical side of interoperability and ignores the semantics and pragmatic aspects of data i.e. its usefulness. However, it is still an applicable definition because it talks about data instead of information.
- *Usability of Information*: The other aspect stressed in these definitions is the notion of usability or usefulness of the information exchanged. The natural question that arises immediately is who determines what is useful and is this determination done before, during or after the information exchange. In the case of IEEE, the use of information is determined by the receiving system as the wording indicates, which implies that the receiving system is not only able to process information but also determine which information it can use and which it should throw out. It also points to the fact that there is a direction of information flow and it is important to identify it during interoperation. The other two definitions do not point to any direction of information flow but repeat the same aspect of usefulness and satisfactory information exchange. This issue is addressed in section 3.

The distinction between data and information is important because all information is data but the converse is not true. According to Ackoff (1989), data are simply a set of symbols, facts or figures, while information is data that are processed to be useful and provide answers to questions such as "who, what". The differentiation between data and information is essential because typically computers deal with data while humans deal with higher levels of abstraction such as information, knowledge and wisdom (Rowley, 2007). Nonetheless, it is possible to represent information in a computer by relating data in context. The term interoperability as described in the earlier definitions point to not only information but useful information. In Ackoff (1989), useful information is knowledge and gaining understanding through knowledge is the goal of M&S. However, there must be a transition between data, information and knowledge that allows a formulation of a useful formal definition of interoperability (Tolk, 2005). A theory of interoperability must include a formalism that describes data to turn it into information on one hand, and relate information in such a way that knowledge can be gained or inferred.

Based on this review, the goal of interoperability is to exchange useful information. Information is understood to be data in context, but the notion of a context is not formally defined. A theory of interoperability should be capable of formally defining what data is, what information is and what context is. The next important question is the determination of what is useful.

In the current state of the art, the determination of what is useful is completely dependent on the system receiving the data and the sending system is not able to determine whether what is exchanged is useful. The current approach is to introduce an interface that will ensure usefulness. The interface qualifies as a model whose role is to broker information between models. However, the introduction of an interface does not answer the question as to what is useful and actually leads to a paradox:

Paradox 1: *Let's consider two models A and B which are interoperable through an interface I. If I is a model then A and I are also interoperable, therefore, an interface I_1 is required between A and I. I_1 is also a model, therefore an interface I_2 is required. To* *generalize, interface I_n and A are interoperable requires an interface I_{n+1} which leads to an infinite sequence.*

The same logic can be applied to *I* and *B*. Consequently, the introduction of an interface is an outcome of interoperability not interoperability itself. A theory of interoperability must be able to explain interoperability without falling into an infinite recursion. This issue is directly addressed in section 4.

In all three definitions, systems that merely exchange information are not interoperable unless the information is deemed useful. Inherent to these definitions is the notion that the right information is exchanged which by definition implies some type of context sharing between the systems, i.e. there is a common understanding of the meaning and context of use of the information. This observation leads to the idea that interoperability should be considered in levels rather than a whole as will be discussed in the next section.

2.2 Levels of Interoperability

The two sets of definitions discussed in the previous section provide an informal view of interoperability. Those definitions including the ones from standardization bodies point to a continuum of interoperability. The ISO/IEC definition focused on the exchange of bits and bytes entirely, while the other definitions address the exchange of information between systems. The fact that interoperability exists in levels has been remarked upon in the body of knowledge (Clark, Numrich, Howard, & Purser, 2001), and this section reviews the relevant classifications of interoperability.

Dahmann, Salisbury, Barry, Turrell and Blemberg (1999) distinguish between *technical* and *substantive* interoperability. Technical interoperability deals with protocol, connectivity, time management, and other hardware and software related issues. In other words, technical interoperability ensures that information exchange is possible. Substantive interoperability on the other hand focuses on aligning the underlying models. It deals with differences in scope, resolution and structure between models. The goal of substantive interoperability is to make sure that the information exchange is meaningful (Clark et al., 2001).

Petty and Weisel (2003) extend the two layer approach by distinguishing between the alignment of models, the interoperability protocol, the communication protocol connecting systems and the underlying implementation hardware. Using this classification, Tolk & Muguira (2003) further refine the notion of substantive interoperability first introduced in Dahmann et al.'s work to form the Level of Conceptual Interoperability Model. The LCIM comprises seven levels of interoperability:

- Level 0: The systems are not connected, no interoperability.
- Level 1: The systems can exchange bits and bytes. A physical connection based on a communication protocol is established. The systems are on the level of *technical interoperability.*
- Level 2: The systems share a common data format and agree on a common syntax. At the level of *syntactic interoperability*, the bit and bytes exchanged can be grouped to form symbols. At this level, systems share a common reference physical data model instance.
- Level 3: The systems harmonize the meaning of the symbols they exchange. The level of *semantic interoperability* implies agreement on the definition of terms through a process of disambiguation. The systems share a common reference physical model.
- Level 4: The systems are aware of the context in which the symbols they exchange are used. At this level the systems are aware of all the possible groupings of symbols and how they are related. The level of *pragmatic interoperability* implies the awareness and sharing of a common reference logical model.
- Level 5: The systems understand the processes that will use the symbols they exchange. At the level of *dynamic interoperability* the assumptions and constraints of processes are described unambiguously, and the behavior of systems is predictable during interoperation.

- Level 6: The underlying concepts represented by the symbols are described unambiguously. The level of *conceptual interoperability* implies the alignment of the models represented in systems. The systems share a common reference conceptual model that captures the assumptions and constraints of the corresponding real or imaginary object.

Page, Briggs, and Tufarolo (2004) introduce an alternative view that distinguishes between issues of integratability related to networks and connectivity, issues of interoperability, which is the realm of simulation or implemented models, and issues in composability, which is the domain of modeling. The LCIM has been used as a maturity model of interoperability in diverse fields including the Gridwise Interoperability Workshop (2007) and its applicability has been shown by Page et al. (2004) and Tolk, Turnitsa, and Diallo (2007).

Classifying interoperability is useful in examining systems once they are interoperable. The main assumption in this case is that systems are interoperable which does not explain what it means to be interoperable. All classification reviewed in this section refer to the alignment of the underlying models. This alignment results in an interface whose purpose is to ensure that the underlying the models are aligned. Consequently it leads back to the paradox identified earlier. The classification of interoperability in level while very useful in determining what needs to be aligned still does not formally answer the questions posed earlier, namely what is meaningful or useful and how is it measured.

The other key problem in terms of the layered approach is that there is no clear distinction between 1) representation issues whether they are logical and/or conceptual, 2) representation issues dealing with the inclusion or exclusion of assumption in processes and 3) technical issues dealing with the use of a common syntax and the alignment of unstructured and semi-structured data. The layered classification can be applied to any of these issues. As an example the underlying conceptual model has many equivalent conceptualizations which in turn have many equivalent dynamic models all the way down to the technical representation. Conversely, issues that are deemed technical such as the representation of a byte follow a standard that is by definition an underlying agreement on a concept (a byte is eight bits, a bit can take only two values etc.). A theory of interoperability should be able to explain what interoperability is as a whole.

In summary this section reviewed the notion of interoperability and provided a set of definition ranging from informal to semi-formal. From these definitions, interoperability appears to have three defining characteristics 1) it deals with information exchange, 2) the information has to be usable, useful or meaningful depending on the definition and 3) it happens in levels. All three parts have been studied to some extent and dealt with through standardization efforts in the case of information exchange and maturity models have been developed to address the idea of a continuum of interoperability (Page et al. (2004). This section also addressed the difference between data information and knowledge and showed that the challenge in formally specifying interoperability is directly related to the specification of a formalism that relates data to information and information to knowledge. The next section will review terms used in conjunction with interoperability. Some of the terms such as model and simulation have to be formally defined before interoperability is formalized; other terms such as composability will be affected by formal interoperability, but that discussion is outside the scope of this dissertation.

2.3 Model, System and Referent

Interoperability is informally defined as the exchange of meaningful information between systems during execution. In order to provide a theory of interoperability, it is important to first define what is interoperating. In previous sections, the terms "system" and "model" have been used to refer to what is interoperating. The focus in those sections was on understanding what interoperability is, and therefore, it was not important what the things were called. System seems to be synonymous with model in M&S and the execution of a model over time is called a simulation (Weisel, 2004). However, in order to formally define what interoperability is and how it works, the formalism must extend to what is interoperating and how is it captured. The goal of this review is to investigate what is interoperating, what it is called and how it is captured. Rather than provide an extensive list of definitions, the next section will go over the definitions found in three main bodies of work that have focused on providing formal definitions of model and simulation.

2.3.1 Discrete Event System Simulation

First introduced twenty five years ago by Zeigler, the Discrete Event System Specification (DEVS) is a systems engineering based formalism used to specify models. DEVS has three main components which are the Discrete Event System Specification (DESS), the Discrete Time System Specification (DTSS) and the Discrete Equation System Specification. The main components of this framework for M&S are source system, experimental frame, model and simulator. The following definitions are offered:

- *Model*: According to Zeigler, Praehofer, and Kim (2004) a model is "*a system specification, such as a set of instructions, rules, equations, or constraints for generating input/output behavior. Models may be expressed in a variety of formalisms that may be understood as a means for specifying subclasses of dynamic systems.*" A model must be understood with respect to a *source system* and an *experimental frame.* A source system is "*the real or virtual environment that we are interested in modeling. It is viewed as a source of observable data in the form of time-indexed trajectories of variables.*" Data of the source system is obtained within a frame of interest to the modeler which in the Discrete Event System Specification (DEVS) is the experimental frame. An experimental frame is "*a specification of the conditions within which the system is observed or experimented; it is also the operational formulation of the objectives that motivate an M&S project.*"
- *Simulation*: In order to introduce the notion of simulation, Zeigler et al. (2004) first define a simulator as "*any computation system (such as a single processor, or a processor network, or, more abstractly, an algorithm), which is capable of executing a model to generate its behavior.*" Consequently, a simulation is the execution of a model to replicate its behavior.

2.3.2 Formal Theory of Composability

Petty and Weisel's (Petty & Weisel, 2003) work is rooted in an attempt to develop a formal theory of composability. In contrast with the system theoretic approach of DEVS, computability theory and mathematical logic serve as the basis for formally defining a model and a simulation.

The following definitions for model and simulation are proposed (Petty & Weisel, 2003):

- *Model:* A model is a computable function that maps state/input pairs to state/ output pairs. States are a non-empty set and the inputs and outputs are sets of vectors of integers
- *Labeled Transition State:* In order to formally define simulation and validity, it is important to first define a Labeled Transition State (LTS). In theoretical computer science an LTS is defined as the tuple T = (S, Σ, →) where

  S is a set of states, Σ is a set of labels, and $\rightarrow \subseteq S \times \Sigma \times S$ is the transition relation.

  An LTS is deterministic, commonly referred to as a Deterministic Labeled Transition (DLTS) if s→ s ′ and s → s″ implies s′ = s″. A DLTS is the basis for formally defining simulation.
- *Simulation:* A simulation is formally defined as the sequential execution of a model and is represented by a deterministic labeled transition system.

  L (M) = (S, I, M_S) where M is a model, and M_S is the state model of M.

A more detailed discussion of composability theory can be found in (Weisel, 2004).

2.3.3 RAND Report

In an effort to address the challenges of composability and component reuse within the Department of Defense, the Rand Corporation put out a monograph discussing ways to improve the composability of future models and simulations. The following definitions are proposed for model and simulation:

- *Model:* According to Davis and Anderson (2004), a model is "*a representation of a system, entity, phenomenon, or process—the model's referent*". The authors further introduce the notion of a *dynamic model* which can be described as the referent's behavior over time.

- *Simulation*: A simulation is "*the act of using a simulation engine (i.e., a simulator) to execute a dynamic model in order to study its representation of the referent's behavior over time.*" (Davis & Anderson, 2004)

Davis and Anderson (2004) make a distinction between a *conceptual model* which is the "*overview of the referent's behavior at a high level,*" and a *specified model* which implies a specification of input/output pairs at a minimum.

When a model of a real world system (referent) is created, it is implemented via a simulator and executed. Modeling and simulating a referent is usually done within a specified experimental frame. Davis and Anderson (2004) distinguish three experimental frames namely the modeling relation, the simulation relation and the effective modeling relation. While all three frames are applicable to this dissertation the focus is on the simulator because it is the part that is implemented.

2.3.4 Implications for M&S

In reviewing these three bodies of work, it has become clear that rather than talk about models versus non-models, it is more pertinent to introduce the notion of a *model continuum,* where a model is designated according to its level of specificity. Davis and Anderson (2004) lay the groundwork for this continuum by distinguishing between conceptual models, specified models and simulated models with each model adding an increasing level of specificity. The simulated model is executable by a digital computer while the conceptual model is for human consumption. It appears that DEVS and the formal theory of composability refer to a simulated model in their definitions of the term. The hierarchy of model starts with a conceptual model which is an "overview of the referent's behavior at a high level," and a specified model which implies a specification of input/output pairs at a minimum (Davis & Anderson, 2004). Diallo, Tolk and Weisel (2007) add the notion of a static model to encompass the class of data models that capture the structure and relationships of data elements independent of behavior. A dynamic model represents the behavior of the referent over time.

So far in this dissertation, the term system has been used to mean a model implemented on a digital computer. This definition of system is contradicts the use of the term in this section. A system as mentioned in previous sections is similar to a simulated model whereas a system as defined in Davis and Anderson (2004) and Zeigler et al. (2004) is equivalent to the referent within the experimental frame. Regardless of how it is understood, a system fits into the idea of a continuum whether as a conceptualization or as an implementation. In general terms, the model continuum shows that the distinction is not between the level at which a model is specified, rather, the distinction is between what is modeled (the referent) and how it is captured (conceptual model, dynamic model, etc). Consequently, all these models are equivalent in that they reflect an aspect of the referent and none is more powerful than the others especially if one considers each specification as a collection of statements that are true about the referent (Weisel, 2004). This distinction is also reflected in Davis and Anderson (2004) even though the idea that all these models are equivalent is not stated. A theory of interoperability should reflect that separation and be applicable to all models, independent from the level of specificity. Section 3 will introduce a formal model of data from which all the definitions provided in this review can be derived. Until a model is formally defined, a system is a simulation model while a model is any other model in the continuum.

Another observation that emerges from this review is the one-worldview assumption. This assumption is stated formally as follows:

***One worldview assumption**: Given a referent and an experimental frame there is one and only one model of the referent.*

This observation is valid regardless of the level at which the model is specified. The one worldview assumption is based on the fact that the referent is assumed to be real and observable in only one way. Consequently, all observations of the same referent in a given experimental frame are assumed to be equivalent. While this assumption holds true within a given model, it is in general false for a referent when one considers all possible ways to describe and capture the same observation. Consequently, a theory of interoperability should not assume a specific description but rather should hold for all possible descriptions of a model. The implications of this assumption are discussed in more detail in the next section.

The dissertation has reviewed current definitions of interoperability and shown that the current understanding is that interoperability is the ability to exchange information and the ability to use the information thus exchanged. This understanding has been explored and shown to be ambiguous as it is not clear what information is and how it differs from data. Further, the idea of having an interface between models has been shown to lead to an infinite recursion. The dissertation also reviewed the classifications of interoperability and showed that while useful, they assume interoperability and therefore cannot be used as a starting point for a theory of interoperability. The next logical step in trying to define interoperability is to explore what is interoperating. The terms "system," "model" and "simulation" are often used seemingly interchangeably to describe what is interoperating. This dissertation reviewed these terms and showed that a distinction should be made between real or imagined things (referent) and how they are captured (model). The levels at which they are captured while important are all equivalent. Based on all these elements it is now possible to start investigating what interoperability is. However, before this investigation, the dissertation will review the state of art in interoperability practices to gain insight into how interoperability works.

2.4 State of the Art in Interoperability

The main goal of interoperability is to exchange useful information between simulation systems. In order to achieve this goal, it is essential to provide a way to exchange information and come to a common understanding of the information being exchanged. This common understanding results in a specification often standardized that provides a technical protocol to support the exchange of bits and bytes and a description framework to capture the information. The result is an agreement on how to exchange information and how to describe it. The information to be exchanged varies with respect to the available models and what information they can provide and/or consume. Another way to achieve interoperability is to agree on what information should be exchanged in a given domain and agree on how to describe it. How the information should be exchanged can either be agreed upon or left to the users. The result is a common model often standardized that describes the domain of interest. In this dissertation the first approach is called the common framework approach and the second is called the common language approach.

These two approaches are not competing or mutually exclusive. For instance, it is possible to have information in the common framework approach that is used so often that it becomes a de facto standard. Conversely, there might be a description framework that so many models support that the common model is always expressed using that framework. The differentiation is not on the outcome of the approaches themselves since they share a common goal; it is mainly on what the focus of the approaches is. In fact, they will be shown to be equivalent in section 4. This section reviews the protocol and the common language approach and provides a rationale for the establishment of a theory of interoperability that will serve as the foundation for frameworks and future tools.

2.4.1 The Common Framework Approach: Interoperability Standards

One approach in creating interoperable solutions is the use of a standardized way to describe and exchange information. The benefits of this approach are the use of a common technical infrastructure and a flexible way to capture information. In this approach the one worldview assumption is violated and a multiple worldview assumption is adopted instead. The multiple worldview assumption is formulated as follows:

Multiple worldview assumption: *Given a referent and an experimental frame there is a potentially infinite number of models of the referent.*

The common framework approach supports this assumption by providing a common way to capture parts of the model of the referent that is relevant to exchange. Standards such as the High Level architecture (HLA) (IEEE, 2000), the Base Object Model (BOM) (SISO Base Object Model Product Development Group, 2006), the Distributed Interactive Simulation (DIS) (IEEE, 2002), the Test and Training Enabling Architecture

(TENA), (United States Department of Defense, 2002) the Aggregate Level Simulation Protocol (ALSP) (Fisher, 1994) specifically address the levels of technical and syntactic interoperability by offering a technical framework for the exchange of information. Semantic interoperability is reached by agreeing on the meaning and structure of information by either enumerating all possible statements (DIS) or through a harmonization process (Federation Object Model in HLA).

Other technical standards such as the Extensible Markup Language (XML) (World Wide Web Consortium, 2004) enable systems to exchange data and metadata at run time but assume adherence to a common schema to reach the level of semantic interoperability. Structured Query Language (SQL) is also used when the information is structured in the form of a relational model that is logically specified. The effective use of SQL for semantic interoperability assumes that systems share a common logical model describing entities and their relationships and that the meaning of terms in the model has been agreed upon. Service Oriented Architectures (SOA) in the form of Web Services has been used as an alternative to HLA, DIS and other traditional standards (Hieb, Pullen, Sudnikovich, & Tolk, 2004; Brutzman, Zyda, Pullen, & Morse, 2002; Blais, Brutzman, Drake, Moen, Morse, & Tolk, 2004).

Web Service standards promoted by the World Wide Web Consortium (W3C) address the technical and syntactic level of interoperability with the added focus on reuse, flexibility and portability. The semantic level of interoperability has been addressed through additional standards such as the Ontology Web Language (OWL) and OWL for services (OWL-S). OWL provides a standardized framework for describing information using descriptive logic while OWL-S is a standard that aims at facilitating automatic and accurate service discovery and use. While Web Service related standards enforce a common set of metadata to describe interfaces they also assume that the meaning of the terms described by the metadata is harmonized. The various standards and their corresponding level of interoperability are discussed in detail in Tolk , Diallo, King, and Turnitsa (2009).

The common framework approach enables technical and syntactic interoperability by establishing a common framework and syntax to describe information. In HLA, there is a Federation Object Model (FOM) that contains the objects and attributes to be exchanged based on the HLA protocol. In DIS systems exchange Protocol Data Units (PDU) using standardized communication services. PDUs provide information about simulated entity states, simulation management and entity management among other things (IEEE, 2002). Web Service related standards describe information in terms of inputs and outputs that are exchanged through the Simple Object Access Protocol (SOAP). The common framework approach while successful in providing a technical framework is insufficient to guarantee semantic interoperability between systems (Dahmann, Salisbury, Barry, Turrell, & Blemberg, 1999).

This observation establishes the need for a more formal approach to deal with semantic alignment. With the emergence of XML, the interoperability community started developing metadata registries to store information and facilitate its discovery. One such registry is the Department of Defense Metadata Registry and Clearinghouse (DoD Metadata Registry, 2008) whose goal is to provide:

"software developers access to data technologies to support DoD mission applications. Software developers can access registered XML data and metadata components, (...) and reference data tables and related meta-data information such as Country Code and US State Code"(DoD Metadata Registry, 2008).

Metadata registries, while useful in storing information, do not have the structure necessary for a formal semantic alignment simply because they do not maintain the context of information once it is store or become too complex to be useful (Connors & Malloy, 2007). As a result, the structure and context of information is lost and instead of being solved, problems such as homonyms, synonyms and namespace identification become harder to identify. In order to harmonize the meaning of data with respect to a domain, communities of interest (COI) started developing data management working groups to identify define and classify all terms relevant to its domain. The next section discusses the COI approach and its contribution to interoperability.

The common framework approach provides a technical infrastructure to exchange bits and bytes and capture the description of information. However, due to the one worldview assumption not holding true, the same referent can be described differently by different systems which leads to the need to 1) capture the referent 2) align the descriptions. The process of aligning descriptions leads directly into the paradox of infinite recursion except in this case it is not apparent due to the basic agreement on the technical protocol. Consequently, technical interoperability can be easily supported, but semantic interoperability is a recurring challenge. In order to address this challenge, metadata registries were introduced to capture the referent and help facilitate the alignment of descriptions. This approach is a step closer to establishing a common model of information, but the commitment to such a model would go against the multiple worldview assumption under which the approach operates. Having reviewed the common protocol approach and shown that while it provides great flexibility, it still faces the challenge of misaligned semantic, this dissertation will discuss the common model approach which starts with a mandated model.

2.4.2 The Common Model Approach: Common Reference Models

The goal of the common model approach is to provide a model that captures the domain of discourse of an area of interest. The common model approach abides by the one worldview assumption and assumes that since all models of the same referent are equivalent they can be mapped. Therefore, the common model focuses on the description of the information exchange requirements for a given domain. The common model is often established by a community of interest. A community of interest is defined as a "collection of people that are concerned with the exchange of information in some subject area" (Renner, 2001). In the military domain for example, there are currently fifty-five COI in DoD (Connors & Malloy, 2007). One of the main goals of a COI is to provide a dictionary of terms, taxonomy and a CRM representing the domain of discourse. As an example, the Joint Command Control and Communication

Interexchange Data Model (JC3IEDM) can be considered the CRM for the North Atlantic Treaty Organization (NATO) Command and Control COI.

The CRM approach is a formalization of the informal semantic alignment process used in the common framework approach. It mandates a model for information exchange that explicitly captures the data elements, relationships and applicable business rules of the COI. This approach is very similar to the federated database (FDB) approach (Spaccapietra, Parent, & Dupont, 1992) and the federated ontology approach (Benjamin, Akella, & Verna, 2007). The FDB approach creates a CRM in the form of a federated scheme merging all existing systems while the federated ontology approach mandates a CRM by creating or using a domain level or upper level ontology that captures the entities and relationships of the domain or COI. Consequently, the same advantages and drawbacks of the CRM approach are applicable to those two approaches. It is worth pointing out that the federated database and ontological approaches require systems to provide a structured description of their internal representation in the form of a schema in the former and ontology in the latter as a prerequisite to participating in the federation.

The introduction of a formal model for information exchange has the caveat of forcing a certain view of the world to all participating models thus reverting to the one worldview assumption. It also introduces a mandated abstraction of reality focused on the information exchange characteristics of the COI which do not necessarily correspond to those of any particular model. By definition, the CRM captures the information exchange requirements of the COI based on relevance while individual models capture information based on an objective. What is relevant to a model with respect to a given objective is not necessarily relevant to the COI and even if it were, it might be captured in a different context. The CRM is therefore not necessarily aligned with any particular model and in fact operates in a different context than any model. Other drawbacks and challenges of a CRM are discussed in more detail in (Lasschuyt, van Hekken, Treurniet, & Visser, 2004). The CRM can be recursively aligned with the models with a bottom up /top down approach that will result in extensions in the CRM until alignment is achieved to satisfaction as discussed in Tolk, Turnitsa, and Diallo (2007). However, this process is equivalent to interoperability as it involves the meaningful exchange of information between individual models and the CRM. Therefore the proposed alignment falls under the purview of the challenges reviewed in the previous sections.

The CRM approach, while providing semantic accessibility to models, introduces a new set of problems for the federation. Semantic alignment of a different kind is still needed. The problem is no longer how to agree on terms and their relationships but how to align the information exchange requirements for a COI as specified in the CRM, with the information exchange needs and capabilities of models in a COI. In fact, the equivalence assumption under the one worldview assumption leads to a multiple equivalent worldview assumption which in turn leads to the infinite recursion problem. As a result, neither the protocol nor the CRM approach offers a set of consistent rules to deal with the technical, syntactic and semantic levels of interoperability. In fact, once the technical aspects are solved, both approaches rely heavily on consensus, agreement and engineered interfaces on the semantic level. The resulting federation is a reflection of these agreements rather than the capabilities and needs of models based on a set of requirements. Section 4 will remove these inconsistencies by formally defining interoperability independently from the LCIM and showing that all levels must be supported in order to have interoperability.

This section reviewed the state of the art in interoperability approaches. One approach is to provide a technical framework and a common way to describe information. This approach (common framework) provides a flexible way to support interoperability but requires semantic alignment in order to reach the semantic level of interoperability. The alignment process leads directly into an infinite recursion at least at the semantic level. Metadata registries are introduced in order to avoid the infinite recursion by providing a common description of basic elements. The registries, however, are not well structured or organized enough to provide the intended support. The logical next step is to organize COI to provide a well structured CRM that will serve as the basis of information exchange for a given domain. This is the common language approach. The CRM, while avoiding the infinite recursion, reverts back to the one worldview assumption which is not true in interoperability. Both approaches are complementary and support the technical level of interoperability but they fall short in terms of the semantic level of interoperability and above.

In attempting to provide a first step towards a formal theory of interoperability, this dissertation has endeavored to establish a starting point for its investigation. A review of the definitions of interoperability has shown that the term is ambiguously defined, leads to an infinite recursion and requires a formal definition of additional terms which disqualifies them as a good starting point. A review of the classification of interoperability has shown that interoperability is assumed to exist first and therefore provides a good way to measure interoperability but is not a good starting point to understanding what it is. This dissertation shifted focus to investigate what is interoperating as a way to gain insight into what is interoperability. This review showed that a key distinction has to be made between a referent and how it is captured but the different ways to capture a referent are all equivalent. Further, this review showed that current descriptions function under a one worldview assumption that is broken in interoperability. This dissertation next looked into how interoperability works as a basis for a theory of interoperability and informally showed that current approaches support technical interoperability and are equivalent in terms of semantic interoperability. Consequently, the approaches lack generality and cannot be used as a starting point for a theory of interoperability.

Based on the reviews, the starting point of a theory of interoperability is a formal description of a referent and a model that does not fall into a recursion and does not function under the one worldview assumption. Since interoperability is concerned with the exchange of information which implies the exchange of data first, the dissertation will focus on data models (static models) without excluding any other form of models. The next section will first review formal approaches of data modeling found in Data Modeling theory and show how they can be applied to derive a formal description of data in M&S.

3 USING SET THEORY TO DESCRIBE DATA IN M&S

This section reviews formal models of data found in Data Modeling Theory. The goal of this review is to identify the descriptions of data that can be used in M&S and explain why some aspects of Data Modeling Theory do not apply to M&S. This section serves as the basis for a formal model of data in M&S presented in section 4. It is important to note that the focus is on formal models of data specifically because the goal of this dissertation is to contribute to a formal theory of interoperability.

3.1 Formal Models of Data

In order for M&S to develop its Body of Knowledge it must clearly delineate theories from frameworks and frameworks from languages. Theories lead to the development of frameworks which are implemented in a language. As the review in section 2 shows, M&S has focused mainly on frameworks and languages, but not enough on theories. Section 2 also shows that the basis of interoperability is data and therefore a formal theory of interoperability is not possible without a formal description of data. This section will first review existing formal descriptions of data and show how they can be extended to support a formulation of data that is useful in accomplishing the goal of this dissertation. An extended review of semantic data models can be found in (Peckham & Maryanski, 1988); the review presented in this section focuses solely on formal models of data.

3.1.1 Review of Existing Formal Descriptions of Data

The first formal model of data in a system called the relational model (RM) was presented in Codd (1970). The stated goal of the RM is to provide a formal description that separates data, the representation of data in data structures and implementations of those data structures. Codd (1970) formally defines a relation as follows:

"Given sets S_1, S_2,..., S_n (not necessarily distinct) R is a relation on these n sets if it is a set of n-tuples each of which has its first element from S_1, its second element from S_2, and so on. We shall refer to S_j as the jth domain of R. As defined above, R is said to have degree n. Relations of degree 1 are often called unary, degree 2 binary, degree 3 ternary, and degree n n-ary."

A relation in this sense is identical to its mathematical definition that is a subset of the Cartesian product of *n* sets. The RM defines five properties of a relation R when represented as an array or a table namely that:

1) Each column represents an *n*-tuple of R

2) The ordering of rows is not significant

3) All rows are distinct

4) The ordering of columns is significant and

5) The significance of each column is partially carried by its label.

The RM defines set operations on relations that allow insertion, deletion or updating on existing relations and operations on relation such as join and projection to support the derivation of new relations based on existing ones.

The RM serves as the basis but is distinct in many regards from modern Relational Database Management Systems (RDBMS) such as Oracle or SQL. Though SQL is the most prominent implementation of RM it is not equivalent to RM. A discussion of the shortcomings of SQL with respect to the RM can be found in Darwen and Date (1995) and Date (1984).

Schmid and Swenson (1975) argue that the RM supports only syntactic aspects of data and does not sufficiently take into account semantic components specifically when modeling real world objects. The article points out that a relation can mathematically capture properties of objects and groups of objects *"but the relational theory gives no indication about the way in which the world is to be represented by a collection of relations"* (Schmid & Swenson, 1975). This distinction is very important to M&S models since they are by definition purposeful abstractions of reality and therefore even when abstracted need to be kept as close as possible to reality. The article proposes a basic formal semantic data model that represents the real world by a set of objects and a set of instances of relationships to capture how objects are related and how they interact. By doing so, a clear distinction is made between classes of objects or object types and types of relations. Further, the article distinguishes between relations that are inherent or describe objects (*characteristic relation*) and relations that associate instances of objects (*association relation*). A depending relation is defined when the existence of an object (*depending part*) depends on that of another (*ruling part*). Unique depending objects are called *characteristic objects*. Objects that are not in depending relations are considered independent and relations between them are called *associations*. As noted in Schmid and Swenson (1975) "in some cases, the same relationships may be regarded as either a characteristic or an association. That means that the same part of the world can be modeled from different points of view." For M&S systems, the point of view mentioned here reflects the multiple worldview assumption and the recognition that each model needs to be captured so that its worldview can be identified precisely.

Chen (1976) offer a unified view of data modeling approaches called the Entity-Relationship (ER) model. The ER model in its conceptual view distinguishes between entities grouped in an entity set and relationships grouped in a relationship set. Formally, a relationship is an n-ary relation between elements of the entity set i.e. a subset of the entity set. Entities in a set share common characteristics which corresponds to the object type as defined in Schmid and Swenson (1975). Beyond entities and relationships, the ER model separates the notions of attribute, value and value set. A value set is a grouping of values and an attribute is a function that maps an entity set to a value set or the subset of all value sets obtained by taking their Cartesian product. The ER model formally allows relationships to have attributes thus extending the idea of the projection in the RM and making relations entities in their own right. The ER model distinguishes between *weak relationships relations* where the relationship identifies the entities it relates and *regular relationship relations* in which the relation is just a link between entities. The *weak relationship relation* implies a dependency between the entities it relates. This dependence is similar to the notion of functional dependency established in Codd (1979) and Schmid and Swenson (1975) though functional dependencies are defined for attributes of a given entity.

Codd (1979) recognized the need to capture more meaning of data but warns about calling this activity semantic data modeling since *"the task of capturing the meaning of* *data is a never-ending one. So the label "semantic" must not be interpreted in any absolute sense.*" The article describes an extension of the Relational Model to formally capture units of meaning that are as small as possible (Atomic semantics) and large units of meaning that are bigger than the conventional n-ary relation (molecular semantics). As a result, an extended formal relational model called the Relational Model Tasmania (RM/T) is specified. In RM/T a new domain called the E-domain contains surrogates for entities implemented in separate independent databases. The E-domain has E-attributes and acts as the common language for the participating databases. It is important to note that two surrogates are said to be equal if they refer to the same entity. In RM/T each entity has at least one type and a relation (unary) called E-Relation is defined to capture *"all the surrogates of entities that have that type and are currently recorded in the database"* Codd (1979).

Entities are classified as *characteristic* when they describe other entities, *associative* when they relate entities and *kernel* otherwise. Entities can also be subtypes of entities recursively. A subtype of an entity inherits its classification (characteristic, associative, kernel). Inner kernel entities are kernel entities that do not have a subtype. RM/T defines objects that associate entities but are not entities as *nonentity associations*. In addition, the article formulates rules governing the insertion and deletion of entities and discusses the notion of Cartesian aggregation which considers a relationship between objects as a single higher level object based on the ideas discussed in Smith and Smith (1977). Other notions introduced in Smith and Smith (1977) such as generalizations (the formation of a generic object) are further formalized.

Codd (1979) present a catalogue of operations on relations in RM/T and proposes an extension of the two-predicate logic (TRUE, FALSE) of RM to three-predicate logic (TRUE, FALSE, NULL) in order to support the notion of "value at present unknown" and "property inapplicable." The issue of incomplete information in a model with respect to another be it the model's referent (the real or imaginary object it abstracts) or another model of the same referent has been observed in the Body of Knowledge. Speaking of database queries that originate from different viewpoints, Lipski (1979) note:

"there are two essentially different ways of interpreting a query-the external interpretation and the internal interpretation. The external interpretation refers the queries directly to the real world modeled (in an incomplete way) by the system, whereas under the internal interpretation the queries refer to the system's information about this world, rather than to the world itself."

Lipski (1979) proposes a formal system to theoretically study issues related to incomplete information in a system. An information system is formally defined as a triple associating a finite set of objects, a finite set of attributes and a function that maps every attribute to a nonempty set of attribute domains. The article rejects the idea of the three-valued logic because it is not nuanced enough to handle the issues relative to information incompleteness and reverts to a two-valued logic by introducing the attribute value domain.

This section reviewed the relevant formal definitions of a data model. Codd (1970) introduce the relational model as the first formal specification of data using Set Theory. In this model the idea of functional dependency between data is defined between attributes. Schmid and Swenson (1975) argue that the relational model does not sufficiently capture semantics and propose a formal semantic model that separates objects, their relationships and types of dependance between objects. Chen (1976) present a unified view of data modeling with the ER model in which entities, relations, attributes, values and value sets are formally defined. The ER model extends the idea of functional dependancy to entities as well as relationships. Codd (1979) propose an extension of the relational model the RM/T that includes the ideas presented in Schmid and Swenson (1975) and Chen (1976). The RM/T offers a three-valued logic (TRUE, FALSE, NULL) to deal with the representation of incomplete data in a data model. Lipski (1979) formally defines an information system and refutes the three-valued logic approach as unable to handle the problem of information incompleteness and proposes a two-valued logic coupled with a set of values that an attribute can take i.e. its value domain.

The next section discusses the implications of the formal specifications reviewed in this section for M&S and justifies the need for a generalized view of data that takes into account the multiple worldview assumption and avoids the infinite recursion problem.Another paradox and its implications are presented along with one of the key assumptions driving interoperability. The section motivates the need to generalize the ideas interoduced by Data Modeling Theory.

3.1.2 Implications for M&S Interoperability Challenges

Data Modeling Theory has focused on capturing and communicating more semantics. This endeavor has taken historically two forms. One is the introduction of hierarchical structures to represent how things are related and the second is the reliance on terms to carry the meaning of what is represented. While the reliance on structure is adequate for M&S based on the review presented in section 2, the reliance on terms to carry meaning presents a central problem. Aside from the issue that computers cannot understand the meaning of terms, in M&S a term is a model because just like any model it is an abstraction of reality. This observation leads to another paradox that is stated as follows:

Paradox 2*: Given a term that has some meaning, it takes a least one term to describe it. The describing term needs at least one term describe it, which leads to an infinite recursion.*

This paradox is similar to the "symbol-grounding problem" found in Artificial Intelligence (Turing, 1950). In order to avoid this paradox, a starting point must be provided in the form of either an initial set of terms that all agree upon or a description of terms that all can refer to. This description of terms can be a dictionary, a taxonomy or an ontology. The agreement on the intital set of terms violates the multiple worldview assumtipon as it imposes a set of models (a term is a model) to capture the referent. The description of terms if captured as a set guarantees that each term is unique, and it is uniquely referencable. Paradox 2 is avoided not by agreing on a common description but by agreing that each term has a unique name withing a model and has one and only one meaning, namely the meaning assigned to it by the modeler. The description is essential in interoperability because it allows modelers to establish an equivalence relation between two sets of terms. This notion is formally captured in the next section as a domain.

A Data Model in Data Modeling Theory is usually a means to store data that is relevant to a domain. While it is a model in the M&S sense of the term, it is not developed to answer a modeling question. Therefore, the model is an abstraction of reality but not a purposeful one. (Lipski, 1979) argued the need to differentiate between the real world and the model of the real world and showed that if a question that is relevant to the real world is posed to a model of this world that did not consider the question in the first place, it leads to the problem of incomplete information. This problem simply does not exist in M&S because of the purposeful nature of the model. A question that the model did not consider is simply not within the purpose of the model. The notion of incomplete information, however, is rooted in the representation of the model, i.e. what is captured and what is ignored. In the general case, the idea that objects have attributes which have value domains does not hold true for M&S as shown in section 2. It is rather a particular case in which a modeler decides what is an object, what is an attribute, what are the value domains and relates them in a way that is suitable to their needs. Further, different modelers might model the same thing differently, one as an object the other as an attibute. This observation ties to the discussion in section 2, when the need to separate the referent from the model was first presented. Objects, attributes and value domains are possible representations of a referent. The selection of objects, attibutes and value domains depends on the purpose of the model or the modeling question one is trying to answer. The next section will formally capture this generalization and show how it maps to the formal models reviewed in this section.

The structure of models as reviewed in this section are assumed to be unique for a given referent. For M&S systems and interoperability, there exist multiple possibly equivalent structures. Further the structure addresses how things are related and not how they change with respect to one another. This issue is discussed in more detail in section 3.2.3. Data Modeling Theory needs the strucure for the purpose of data integrity meaning data that belongs together is always provided as a whole or not at all. In M&S interoperability however, the idea of integrity is not explicitly supported due to the independence assumption which is described as follows:

***Independence assumption**: For a given model, each element within the model exists independently from any other element.*

The independence assumption is the driving force behind the definition of interoperability and the interoperability approaches reviewed in section 2. However, this assumption leads to multiple versions of the truth in a federation of models, that is to say that if two identical elements participate in different structures they might change differently for the same input. As a simple example, let's consider model one where a tank has a crew (tank and crew are objects that are related) and model two where a crew is part of a tank (crew is an attribute of the tank). If the models interoperate over tank or exchange information about the tank, the destruction of a tank results in the destruction of the crew in model two while the crew might still be alive in model one if they were not in the tank at the time it was destroyed. In this federation, it is possible for the crew to be both dead and alive at the same time. The formalism will generalize the idea of dependencies in section 3.2.3 and show how to avoid the independence assumption.

These two assumptions motivate the need to separate the structure of models from the meaning of terms. The structure of models in Data Modeling Theory is applicable to M&S and brings the additional element of data integrity to avoid the independence assumption at work in the approaches to interoperability. The reliance on terms to carry meaning leads to an infinite recursion, which can be avoided by creating a domain of validity for the terms used in a model in the form of a set and allowing equivalence relations to be established between sets. The section shows that the definition of a data model in M&S should take into account the modeling question and allow all possible representations of the referent to be captured. The separation between the referent and the model of the referent are shown to be essential in M&S. The next section introduces a formal specification of data in M&S systems based on the review presented in this section. This formal specification takes into account the aspects of data from Data Modeling Theory and conceptual modeling as well as additional aspects essential to M&S.

3.2 A Formal Specification of Data in M&S Systems

This section presents a formal specification of data in M&S systems based on Set Theory and shows how it can be used to capture the formal descriptions of data presented in the previous section. The formalism presented here is an extension of existing ones but it captures additional requirements for M&S systems most notably the need to relate the referent to the set of objects that models it. The section will then examine the effect of set operations on the formalism and present a set of theorems and proofs about data in M&S. The goal of the formalism is to provide a general description of a data model in M&S that avoids the one worldview assumption and does not fall into the paradox identified in section 2.

3.2.2 A Relational Model of Data in M&S

A formal definition of a model has to take into account not only the model but also its relation to the referent on one hand and its relation to the simulation on the other hand. In order to reason about models in M&S, it is important to decouple the description of the model from its representation and its representation from its implementation. Based on the formal specifications of data reviewed earlier, there is an agreement that a model is constituted by entities that might or might not be related, and they exhibit properties that can be observed under certain conditions. In addition, the ER model introduces the notion of value and value set of attributes or properties.

These three terms-entities, properties, value domains- constitute the core of a data model in Data Modeling Theory. For M&S, this dissertation adds the notion of a domain which is similar to the traditional view of the value domain (a collection of values that an attribute can take) but is generalized to encompass the domain of discourse. Finally, the term element is used to mean anything real or imagined that can be described and/or observed. The terms are defined as follows:

- *Elements* are real or imaginary things.

- *Entities* are abstractions of elements. It is worth noting that by this definition any abstraction including processes and relationships are considered entities.
- *Properties* are the characteristics of an element.
- *Symbols* are the representations of elements. Symbols can be numbers, strings, images, text or a combination of symbols.
- A *domain* is a collection of unique symbols. The domain is the set of elements that belong to a context. Every element is uniquely identifiable within a given domain.

A *referent* differs from an element in that an element is the part of a referent that the modeler is interested in capturing. Consequently, a *referent* is formally the set of elements because capturing the referent reduces to capturing the elements that are of interest to the modeler and abstracting elements that are deemed irrelevant to answer the modeling question. The referent as captured in this dissertation is the result of a conceptualization process. In that sense, the referent is not the observable thing in the real world, rather it is the result of the observation. The observation process is a modeling process that need not be captured but its results (elements) are the things the modeler intends to study. Elements are the only things accessible to a computer as machines do not have access to reality.

It is important to note that elements are used to separate the notions of entities and properties and how they are related. Elements can be modeled as entities or properties and then related to form the entity-property relationship. This separation of elements and how they are represented reflects the general case and therefore subsumes the entity-property-value triple introduced by the ER and RM.

Given these definitions, let us first formally capture a conceptualization of the referent:

Definition 1.1: *Let S be the set of elements, Ω the set of entities, Π the set of properties, V the set of symbols. A conceptualization S is formally captured by the categorization of* *elements into entities, properties or symbols. Formally, a conceptualization is defined as a partial function F such that:*

$$F(S) = \begin{cases} \Omega\ if\ S\ is\ an\ entity \\ \Pi\ if\ S\ is\ a\ property \\ V\ if\ S\ is\ a\ symbol \end{cases}$$

A function is a binary relation between sets in which every ordered pair has a different first member. A partial function is a function for which not every member participates in the relation (Hein, 2002). The definition of a conceptualization as a partial function accounts for the complexity of elements and the fact that it is impossible to capture them completely for non-trivial cases. Staying with functions, a bijection, surjection and injection are defined as follows (Hein, 2002):

- *Injection*: For every element in S, $F(s1) = F(s2)\ implies\ s1 = s2$;
- *Surjection*: For every entity, property, or symbol $x,\ x = F(S)$;
- *Bijection*: A function is bijective if it is injective and surjective.

There is no requirement that reality be injective (distinct elements of S map to entities, properties or symbols) or surjective (every entity, property and symbol in their respective set must refer to an element) during the conceptualization process. However, once a commitment is made, a representation (capture) of *F(S)* must be bijective and consequently always has an inverse. An inverse is a function G such that for an element s and an entity, property, or symbol $x,\ G(x) = s\ \ if\ \ F(s) = x$ (Hein, 2002). Definition 1.2 captures the need to capture the conceptualization process as a bijective function.

Definition 1.2: *An element within S is an entity, a property or a symbol; otherwise stated the three sets are mutually disjoint.*

The existence of an inverse and the bijective nature of *F(S)* work well under the multiple worldview assumption but it does not avoid paradox 2 which states that there is more than one possible equivalent way to represent a referent. The introduction of the domain as a collection of symbol provides one way to avoid the paradox. To illustrate, let's consider the following problem absent the notion of a domain:

Given two conceptualizations, is it possible to determine whether they are equal?

This question is central in interoperability as it determines whether the information exchange is meaningful. Since conceptualizations are functions and two functions F and G are said to be equal if $F(s) = G(s)$ for every element *s*; it is simply a matter of comparing the input/output pairs of the two functions to come to a conclusion. However, if there is a reliance on terms to carry the meaning of elements, the comparison of the two functions will never terminate due to the paradox as the algorithm has to map conceptualizations to conceptualizations ad infinitum. As a simple proof, let's consider a single element s captured by these two functions. If *s* can be described by an element s^1 conceptualized using a function $F^1(s^1)$, it requires at least one more comparison to determine equivalence. In general:

$$F(s) = G(s)\ if\ \ F'(s') = \ G'(s') = \cdots if\ \ F^n(s^n) = \ G^n(s^n)$$

This process does not terminate as the set of terms that can be used to describe an element is uncountable because there is no surjective mapping to the set of natural number due to the existence of homonyms and synonyms within the set. This problem is also known as the *Equivalence Problem* and has been shown to be unsolvable in the general case (Hein, 2002).

In order to tackle this issue, let's consider the domain and the following definition:

Definition 1.3: *Given a set S of elements and a non-empty set of domains Δ, every element in S is associated with a domain.*

Mathematically, we define the tuples:

- *α is a subset of* Ω XΔ, *the Cartesian product of entities and domains;*
- *β is a subset of* Π X Δ, *the Cartesian product of properties and domains;*
- *γ is a subset of* V X Δ, *the Cartesian product of symbols and domains.*

For every element s belonging to S, s belongs to α, β or γ. In addition α, β and γ are disjoint as a consequence of definition 1.2. The domain reduces the term to its symbol. It has the meaning assigned to it its domain which might or might not be the same as other meanings it has in other domains. Assigning meaning to a term is a modeling decision captured by definition 1.3. Determining equality between conceptualizations is reduced to determining equality between domains by definition 1.3.
Formally:

$$F(s) = G(s) \; if \;\; \Delta(\mathrm{F(s)}) = \Delta(\mathrm{G(s)}).$$

For a given model, the domain being a list of unique symbols is mappable to the set of natural numbers and is therefore countable. The fact that the domain is a countable set removes the infinite recursion, but the comparison process is still computationally complex for a reasonable size domain. The complexity of this comparison is discussed in detail in section 4. Equality of domains as discussed in section 3 can be established either through a common set of symbols or a set of functions that map one group of symbols to another. A direct consequence of the introduction of the domain is the shift in the role of a referent which is now undistinguishable from the conceptualization at least in terms of its description. The distinction, while existing in reality, disappears once conceptualizations are captured.

This leads to an important observation directly related to how to avoid paradox 2. The one-worldview assumption must hold true for interoperability which means that in order to avoid paradox 2, all models within a federation have to share a unique set of domains. If models share a common set of terms within a domain, a comparison algorithm will halt if the set is finite and countable. In practice this sharing of terms is done either through a common data model, a common protocol or both as the literature review shows. Another common practice is to mediate between domains to align the terms and thus make them synonymous. All of these practices are equivalent in that they result in the use of a common set of domains.

Having captured a model description, the discussion in section 3 on the implication of data modeling in M&S also mentions the need to capture the structure of a model as a means to carry semantics. Since the domain is the representation of the conceptualization, it is now possible to represent a model of the conceptualization while still following the multiple worldview assumption. A model of a conceptualization is one of the many possibly equivalent representations of the conceptualization which itself is one of the many possibly equivalent conceptualizations of a referent. Conceptualizations might or might not be related through a relationship relation as shown in the RM and ER models. The following definition is a generalization of all possible relationships including relationships between relationships that are expressible within this formalism:

***Definition 1.4**: Given Δ the set of domains, we define the relation ρ as the subset of Δ X Δ the Cartesian product of domains.*

The relation ρ captures the relationship between entities and entities, entities and properties, entities and symbols, properties, properties and symbols and symbols and symbols. In addition, ρ captures relationships between relationships if one considers that all the relationships in definition 1.3 are elements that have as domain a subset of Δ X Δ and therefore abide by the previous definitions. The relation ρ is a graph with vertices Δ. Having defined the conceptualization, let us now define a model of the conceptualization:

***Definition 2.1:** A model M of a conceptualization S denoted M_S is the relation $(\alpha, \beta, \gamma, \rho)$.*

By definition 2.1 a model is also a representation of a conceptualization. If M is countable, M is computable. Further if M is finite and countable it can be implemented on a digital computer. However, results derived from these definitions are not limited to computable functions but should apply in general.

***Definition 2.2:** If the cardinality of Δ =1, M_S is the relation (Ω, Π, V, ρ).*

The model defined in these definitions supports the multiple worldview assumption of which the one worldview assumption is now a particular case. The model avoids the paradoxes by separating the referent, a conceptualization of the referent and a model of the conceptualization. The semantics are captured by a collection of groupings of symbols and how they are related. Several additional observations should be pointed out with these definitions. The first observation is that the referent and a model of the referent are not required to be finite and/or countable and therefore are not required to be computable. The second observation is that the definitions do not make any assertions about inherent semantic relationship between the sets. In this sense, any model of the referent is captured under these definitions.

Definition 1.1 does not take a position on which set to capture first and in fact does not require any set to be non-empty except for the domain. It is perfectly acceptable to view the world in terms of properties and symbols or entities and domains. Consequently, the definition does not espouse any predetermined description of the referent. Most importantly, the semantics of the referent and the model are explicitly captured by the quadruple. The introduction of the domain as part of the specification in definition 1.3 plays the same role as the use of labels to carry meaning but in a formal way. Definition 1.4 accounts for the semantic relationships between entities. The existence of a relationship between entities implies a relationship between the domains of these entities. Definition 1.4 implies that a new domain is created by relating entities and the domain thus created is the context of the relationship. Definition 2.1 is a generalization of the traditional view of a model in which entities have properties which have values and those values can in turn be grouped into a value domain. The traditional view does not cover a model in which entities interact and are affected by their environment. In this case, entities might or might not have properties explicitly modeled; nonetheless, there are properties of the environment that are affecting their interactions and properties of their behavior (the exchange of Protocol Data Unit between entities is a simple example). This is the case, for example, in multi-agent models. Another simple example is a grouping of properties of several entities to form a context or the occurrence of events in event-based simulations or the modeling of structures in System Dynamics. Definition 2.1 covers all these models within its specification and additionally allows relationships to be related within its specification.

Definition 2.1 includes the formalisms reviewed in previous sections. For instance the ER model is represented in the triple (Ω, Π, ρ) with Π the set of functions Π: $\Omega \rightarrow V$. The ER model has been shown to be equivalent to the third normal form of the RM (Chen, 1976). Definition 2.2 introduces a special case in which there is only one domain. This corresponds to the one worldview assumption. This definition is usually the one assumed by M&S practitioners but it is hardly the general case. In fact, in terms of interoperation, the assumption that definition 2.2 is always true breaks down as systems start to exchange information.

Given the definitions provided in this section, it is now possible to examine the notion on functional dependency first introduced in the RM model and subsequently addressed in all the formal models reviewed in this dissertation. The next section will examine data models in M&S systems and formally define dependency between elements, properties, symbols and domains.

3.2.3 Data Dependencies in M&S Systems

Normalization is a procedure to ensure the integrity of data and reduce manipulation (insertion, deletion, update) anomalies that could lead to ambiguities due to functional dependency between properties. Given an entity, a property B of that entity is functionally dependent on a property A of the same entity denoted $A \rightarrow B$ if each instance of A corresponds to exactly one instance of B. Functional dependency as introduced in Codd (1970) led to the specification of the first normal form (1NF) and the specification of keys (primary, foreign, etc.). Subsequent work presented in Codd (1971), Fagin (1977) and Zaniolo (1982) has led to the development of higher forms of normalization, but the consensus is that the Third Normal form (3NF or Boyce-Codd normal form) is sufficient in relational models. Functional dependency can be generalized to multi valued dependencies where a property is functionally dependent on multiple properties (Fagin, 1977). The ER model distinguishes between functional dependencies between entities and properties and functional dependencies between related entities focused mainly on the cardinality of the relationship (1:1, 1: n, n: m). The goal of normalization is to have a data model that is consistent over time and accurately captures the meaning of the referent.

However, for M&S systems functional dependency as described above is entirely based on the modeler's intent and the subsequent decisions. In general, while there are guidelines and best practices for modeling, a normalization process as practiced in Data Modeling Theory might result in a model that does not fit the scope and resolution originally intended. At this juncture, it is more important for the model to be separated from the business rules that dictate the interactions with it. This separation is also very useful in augmenting the ability of models to interoperate as it distinguishes between the semantics of the data model and the semantics of the interactions with the data model. As a reminder from the previous section, a model of a referent M_S is the relation (Ω, Π, V, ρ) which is a composition of the subset of the Cartesian product of entities and domains, properties and domains, symbols and domains, and domains with themselves. We distinguish between existential dependency and transformational dependency.

Definition 3.1: *Let X, Y be sets of entities, properties or symbols with respective domains Δ_X and Δ_Y, Y is existential dependant on X denoted XΦY or Φ(X, Y) if the existence of Y implies the existence of X.*

Every element is existentially dependent on itself, and it is worth noting that the set thus defined is a subset of ρ, the Cartesian product of domains which is nonempty implying that Φ is also nonempty. By this definition, multi valued dependency is the particular case where Φ is a function and X, Y are sets of properties where Y has cardinality one. Existential dependency is a generalization of traditional conceptual modeling relationships (is-a, part-of, has-a, etc.) that is able to capture the idea that a designated grouping of elements (entities, properties, symbols) has some meaning (Sowa, 2001). Traditionally the meaning of these groupings is carried by a semantic label assigned to Φ (is-a, parent-of, child-of), but in this case the meaning is carried by the grouping of the two domains or namespaces. In practical terms, semantic labels are meaningful to human consumers, but for computers it translates into an association between elements in one namespace with others in the same or a different namespace. The label is then another term used to capture existential dependency. As a simple example, let's take the statement "*son-of (parents, child)*" to mean "a child is the son of its parents." It can be easily verified that the existence of a child depends on the existence of its parents. This statement allows the identification of the entities that are members of son-of, but it falls short of capturing all the characteristics of the relation, i.e. what does it mean to be the son-of an entity and how to automatically identify those entities. Let's assume that this is an inheritance relationship as the label suggests, then *son-of* means that a child has at least all of the properties of the parent in addition to its own. This is another existential dependency but this time between the properties of the entities. We write son-of $\rightarrow(\Pi_Y \subseteq \Pi_X)$ to capture this relationship. Let us now assume that son-of also means that a son has to have certain properties assigned a constant value. The easiest example is to require that all sons be male. We write son-of $\rightarrow$ $(\Pi_Y = \text{Sex} \rightarrow \text{Sex} \equiv \text{Male})$ to capture that relationship. It is worth noting that because this is an existential dependency between symbols, we use equivalent instead of equal to capture the requirement that a function translating the symbol of Y into "Male" must exist for the property "Sex." This example illustrates the need to express the meaning of a label in terms of dependencies between entities, properties and symbols. It also shows that dependencies exist between domains for humans but must be expressed between terms to have meaning for machines.

Existential dependencies capture the fact that a set of elements (entities, properties, symbols) cannot exist without another. Transformational dependencies exist when in the process of interacting with the M&S model; an update to an element (entity, property, symbol) implies an update to another which often will be the case when users or other systems are interacting with the M&S system.

Definition 3.2: *Let X, Y be two disjoint sets of elements with domains Δ_X and Δ_Y respectively, Y is transformational dependant on X denoted $X\Theta Y$ or $\Theta(X, Y)$ if a change to Y implies a change to X.*

It is trivial to show that every element is transformational dependent on itself and Θ is a subset of ρ which means that Θ is nonempty. Change could be the creation, deletion or update of an element (Dori, 2002). The nature of the change can be captured similarly to existential dependency. As a simple example, a symbol y of Y is transformational dependant on a symbol x of X, thusly, $\Theta(X, Y) \rightarrow y = x + 3$. As is the case with existential dependency, a transformational dependency between domains can translate into dependencies at the entity, property and symbol level. Continuing with the previous example, in *son-of*, child is not transformational dependant (transformational independent) of parents because a change in Parents does not imply a change in child. It is important to note that contrary to intuition, existential dependency does not imply transformational dependency.

The denormalization process in data modeling theory however does not distinguish between these two types of dependency. However, this separation is important in M&S especially in interoperation because it separates the formulation of the model from the interactions with the model. Generally only dependencies at the symbol level are captured, while in M&S, the failure to capture dependencies at the entity and property level is a major cause of problems in interoperation because while there is alignment at the physical level of systems, there is a mismatch between the logical representations of data.

This section formally defined existential and transformational dependency of data in M&S and distinguished between dependencies that involve statements about the existence of elements and those that involve interactions with elements. A review of formal approaches in data modeling theory shows that in general it is important to distinguish between the structure of data and its meaning. Elements can be classified in terms of entities, properties and symbols. For M&S systems, which are model based by definition, this classification does not capture the notion of a referent thus the introduction of a fourth classification (domains) that captures the context of terms used to identify entities, properties and symbols. The next section will use the definitions formulated here to define and contrast interoperation and interoperability.

4 INTEROPERATION AND INTEROPERABILITY

Interoperability and current approaches to interoperability have been reviewed in section 2. While the focus is on interoperability, it is obvious that interoperation is subsumed within that concept. Interoperation is perceived to be necessary but not sufficient for interoperability. The definitions and models of interoperability as well as industry standards for interoperability are evidence of this common understanding with an accent on semantic interoperability. The review also shows that semantic interoperability can be enhanced by agreeing on the meaning of labels and models either through standardization or a CRM. Interactions with the model are subsumed within this agreement whatever form it takes. Based on the formalism developed in this dissertation, it has been shown that instead of interoperability, current approaches address interoperation. Without using any particular definition, the review done in section 2 shows that key characteristics of interoperability are the exchange of information and the use of the information thus exchanged. The exchange of information is interoperation and the evaluation of its usefulness determines the degree of interoperability between the systems. Using the formalism defined previously we can formally examine *interoperability* and *interoperation.*

4.1 Interoperation

Interoperation informally captures information exchange between systems. Interoperation is formally defined along with its characteristics and requirements as follows:

Proposition 1.1: *Let M_S be an arbitrary model of a referent S, Φ the set of existential dependencies within M and Θ the set of transformational dependencies within M. A model A is said to interoperate with M if there is a subset of Φ in A or A and M interoperate, denoted A Φ M if $\Phi(A) \cap \Phi(M) \neq \emptyset$.*

A and M are said to interoperate over the subset of Φ which represents the intersection of the sets of existence dependencies between the models. The subset of Φ over which A and M interoperate is the set of elements that A can produce and M can process. By definition this subset is the CRM of A and M. The degree of interoperation between A and M is the cardinality of the CRM. Starting with the definition formulated so far, it is possible to derive some interesting theorems of interoperation:

Theorem 1.1(Existence Theorem)*: Given two models A and M, A and M interoperate implies the existence of a CRM.*

This theorem will be referred to as the *existence theorem* in the remainder of this work. **Proof**: Let us assume that Φ (A) and Φ(M) are disjoint but still interoperate. If A exchanges data with M there is at least one element e, in A that goes to M, but since A and M are disjoint, e is not in M or e does not exist in M which contradicts the fact that A and M are exchanging e.

Using the existence theorem and the properties of the intersection, it is possible to explore the properties of interoperation:

Property 1.1*: Interoperation is commutative i.e.* $A\Phi M \rightarrow M\Phi A$

Lemma 1.1*: There is only one CRM when A and M interoperate*

Property 1.2*: Interoperation is associative i.e.* $(A\Phi B)\Phi M = A\Phi(B\Phi M)$

Theorem 1.2 (Uniqueness Theorem)*: There is only one CRM when N models interoperate.*

Property 1.1 is a direct application of the commutative property of intersection. Lemma 1.1 is easily proved by assuming that there is more than one CRM and using the commutative property of interoperation to show that they are all equal. Property 1.2 is also a direct application of the commutative property of intersection and Theorem 1.2 is a generalization of lemma 1.1. It is proved by using lemma 1.1 and property 1.2 to inductively show that there is only one CRM. Theorem 1.2 will be referred to as the *uniqueness theorem* in the balance of this dissertation. These two theorems point to the fact that it is possible to verify whether models can interoperate by verifying whether a unique CRM exists between them. Consequently, interoperation exists only with respect to a CRM and we say that the models are interoperating under a CRM. In practice, the CRM is either mandated or emerges from a federation agreement and in both cases it is a model of the requirements that need to be supported by participating models. This model can be specified using this formalism and therefore for any given model A interoperating with the model, the resulting CRM must exist and be unique. In practice, it means that given a set of requirements and N models, there is one and only one CRM under which the models can interoperate. Consequently, it does not matter whether the CRM is mandated or results from a federation agreement, it ends up being the same.

The existence theorem and the uniqueness theorem state the existence of a unique CRM when systems interoperate, so it is equally important to investigate whether the converse is true. The question can be formulated as follows:

Does the existence of a CRM imply interoperation?

In order to answer this question, let's consider the following two statements and their logical implication:

P: *Models interoperate*

Q: *There exist a CRM*

The truth table associated with the logical implication is presented in Table 1 and shows that the statement $P \rightarrow Q$ is False only when Q is False. Conversely, let's now look at the truth table of $Q \rightarrow P$.

Table 1: Logical Implication of P implies Q

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

Table 2: Logical Implication of Q implies P

P	Q	$Q \rightarrow P$
T	T	T
F	T	F
T	F	T
F	F	T

In this case, the logical implication is False only in the case where P is False. That is to say the statement "*There exists a CRM when models do not interoperate*" is False. The logical implication of these statements yields logical equality which has the truth table shown in Table 3.

Table 3: Logical equality of P and Q

P	Q	$P = Q$
T	T	T
F	T	F
T	F	F
F	F	T

The existence theorem can be reformulated as follows:

Theorem 1.1a*: Given two models A and M, A and M interoperate if and only if there exists a CRM.*

The theorem as formulated above is also a logical consequence of proposition 1.1 because of the properties of the intersection. There is an additional aspect of interoperation that is revealed by this theorem. Staying with the two statements P and Q as formulated above, the converse of the existence theorem can also be formulated:

Theorem 1.1b*: Given two models, A and M, A and M interoperate if and only if there is a consistent CRM.*

Consistent means statements are either True or False in the CRM but not both at the same time.

Intuitively, theorem 1.1b states that two models interoperate not only over the statements that are True in both models but also over the statements that are False in both models. Theorem 1.1b provides a way to verify that a given model is the actual CRM between models. The verification algorithm looks at the truth value of every statement in the CRM and makes sure that there are no contradictions i.e. a statement is not true and false in the CRM. This is an aspect of interoperation that is largely ignored in the Body of Knowledge but impacts the determination of whether two models interoperate. The current approaches to interoperability do not provide a mean to make this verification.

Furthermore for *N* models, the following theorem can be derived:

Theorem 1.3: *The CRM of N models of a referent contains at least k elements and at most the same number of elements k' as the number of elements in the smallest intersection between two models.*

Proof: As a proof, let's assume that for a given referent S there are M_k models with k=0, 1,..., n. This is a safe assumption because each model can be mapped one-to-one and onto the set of natural numbers. Using induction and the cardinality of the intersection, we can easily show that:

$$|M_1 \cap M_2 \cap \ldots \cap M_k| \leq |M_k|.$$

Theorem 1.3 shows that interoperation is bounded, remains constant or decreases as more models interoperate. This observation is explored in more details in the next section. The determination of the existence and value of k and k' will also be investigated in more details in section 4.

In practice, the existence theorem reflects the approaches to interoperability as discussed in section 2. One approach (common language) is to enforce a CRM and thus make interoperation possible by allowing models to derive an intersection with the CRM. The other approach (common framework) is to connect models and select a subset that satisfies a desired common model then use that model as a de facto CRM. However, neither approach takes into account the existential dependencies in deriving the CRM. Consequently, this independence assumption leads to trivial interoperation at best if the elements modeled are indeed independent. At worst, it leads to a CRM that contains contradictions that cannot be evaluated because contradictory statements are not captured.

The existence theorem and the uniqueness theorem show that when models interoperate there exist a unique and bounded CRM. This unique CRM represents what a collection of models is capable of exchanging. In practice, it is hard to discover the CRM and most often the CRM does not match exactly the requirements of the modeling question for which the collection of models is being created. The main reason is that a model is captured as a collection of statements that are true about the referent. The statements that are false about the referent are left out and thus make it difficult to determine the true intersection between models. The modeling question might be answerable by some subset of the CRM, but the complicating factor is the nature of existential dependencies between elements which means that the selected subset has to capture not only elements relevant to the modeling question but also their dependencies. Current approaches do not take these dependencies into account. This dissertation will present an engineering approach that tackles this problem.

In terms of interoperation, the existence theorem and uniqueness theorems are necessary in order to determine the elements that can be exchanged by a collection of models but the theorems do not guarantee that the models will act in concert to provide an answer to the modeling question. This latter part is in interoperability as presented in the next section.

4.2 Interoperability

Interoperability informally captures the notion of the use of information once it is exchanged. The fact that A and M interoperate does not mean that they are interoperable. Interoperability requires the ability of M to use what it receives from A or conversely the ability of A to interact with M following the rules of interaction of M. To illustrate, the following proposition is stated:

Proposition1.2: *Let A and M_S be arbitrary models of a referent S, Φ the set of existential dependencies, and Θ the set of transformational dependencies. A and M are interoperable denoted $A\Theta M$, if A and M can interoperate and* $\Theta(A) \cap \Theta(M) \neq \emptyset$.

The degree of interoperability between A and M is at most equal to the degree of interoperation (cardinality of the CRM). The notion of degree of interoperability is investigated in more details in the next section. As a note, it is impossible to ascribe meaning to a machine like a human or to claim that there is understanding in machine to machine interoperation which would be required to determine whether information exchanged through interoperation is meaningful. This dissertation does not claim that such a thing is possible; rather it asserts that information exchanged is usable if it respects all transformational dependencies of the receiving model. Interoperability will then require the sending model to know of the transformational dependencies relevant to the information it sends. As otherwise stated, those dependencies have to be shared by the models.

Similar to interoperation, interoperability is commutative and associative due to the inherent properties of the intersection of sets. Two theorems of interoperability can be immediately formulated, similarly to theorems of interoperation. While the theorems mention two models they can be extended to apply to N models inductively.

Theorem 2.1: *Given two models, A and M, A and M are interoperable if and only if they share transformational dependencies.*

Theorem 2.2: *Given two models, A and M, A and M are interoperable over a unique set of transformational dependencies.*

Theorem 2.3: *The set of transformational dependencies shared by N models of a referent contains at least k elements and at most the same number of elements k' as the number of elements in the smallest intersection between two models.*

These are the same theorems formulated for interoperation but applied to transformational dependencies. The proofs for these theorems are similar to those presented for interoperation. It is important to note that similar to interoperation, interoperability is over the transformational dependencies that are true or false in all models. This insight is critical because for those dependencies that are only TRUE (or FALSE) in one model, interoperability is not possible which means that models can appear to be interoperable without the possibility to verify whether it is actually the case or a consequence of interoperation. In short, the intersection of the transformational dependencies must also be consistent. It is also important to note that interoperability is bounded, which leads to the following theorem:

***Theorem.2.4**: The number of modeling questions a federation of models can answer is bounded.*

Theorem 2.4 points to the need for a modeling question. Given a list of modeling questions, it is very difficult to determine whether a federation of models can answer it (Page, 2007). The next section investigates the complexity in more details. However, this theorem shows that there are a finite and countable number of unique questions that the federation can answer which leads to the necessity of evaluating a CRM with respect to those questions. In fact each modeling question that a federation can answer is a subset of the CRM. Consequently, interoperability and interoperation are not inherent properties of a model and can only be measured with respect to other models and a research question. Current approaches to interoperability only deal with interoperation because they focus on trivial existential dependencies through the exchange of properties and events in the publish/subscribe paradigm that are assumed to be independent. For interoperability to be supported, transformational dependencies have to be specified and evaluated with respect to the modeling question that the federation is trying to answer. It is important to note that adherence to transformational dependencies does not imply validation. The two notions are different in that interoperability merely guarantees a coherent behavior of the federation while validity is a measure of how close the behavior is to that of the referent modeled.

In summary, the formalism presented in the previous section is used to distinguish between interoperation and interoperability and specify their respective properties. The existence of a unique CRM is proven and formulated in the form of theorems. The impact of these theorems on the state of the art is presented, and the need for additional descriptions or annotations of models to support interoperability is discussed. The next section examines additional properties of interoperation and interoperability using Graph Theory.

4.3 Interoperation, Interoperability and Graph Theory

This section examines the computational complexity of finding a CRM and formally shows that current approaches to interoperability are equivalent. This finding motivates the need for a heuristic that enables interoperation and interoperability. To this end, this dissertation formulates related questions within graph theory, as this allows transfer of a large body of knowledge regarding computational complexity of problems into the domain of interoperation and interoperability.

Graph theory is an area of mathematics focused on the study of connections between pairs of objects or collections of objects (Hein, 2002). This area of mathematics is relevant to the theory of interoperability because, as discussed in the previous section, a model is a composition of relations between elements which results in a structure. The representation of a model as graph is equivalent to its representation as a relation. The representation of models as graphs allows us to inherit all the findings of Graph Theory and apply them to interoperation and interoperability. The determination of a CRM is similar to finding the similarity between two or more graphs. This section will examine the complexity of finding a CRM and formally show that current approaches to interoperability are equivalent. This finding motivates the need for a heuristic that enables interoperation and interoperability.

There are different types of graphs, but in this dissertation the term graph is used to mean multigraph. A multigraph is formally defined as the triple $G = (V, E, \mathrm{A})$ where

- V is the finite set of vertices or nodes;
- E is the finite set of edges;
- $A: E \rightarrow E$, the identity map.

From the definitions of a model provided earlier, many graphs can be specified. It is important to note that we assume that all the sets defining a model are fine and countable and therefore graphs can be constructed. The most general case is to define a finite set of elements as the set V, $E = (\alpha \cup \beta \cup \gamma)$ the union of all possible relationships between elements and $A: E \rightarrow E$ the function that relates entities and their relationships. This dissertation has shown that in order to formulate a model for interoperability, every element must participate in a relation with the set of domains. Consequently, the model can be reduced to the relation ρ, the subset of the Cartesian product of domains, since all other relations are subsets of ρ. The set of vertices can then be the set of domains Δ, E the set of edges between the members of Δ, and ρ the identity map.

Simply stated, a model is simply the graph $G = (\Delta, E, \rho)$ where

- Δ is the finite set of vertices or nodes;
- E is the finite set of edges;
- $\rho: E \rightarrow E$, the identity map.

The formulation of G subsumes the existence theorem. That is to say that a formulation of a model as a graph contains the formulation of all of its existential dependencies. The graph G also captures the relations $\alpha, \beta \ and \ \gamma$ in the form of labels. Having defined a model as a graph, it is now possible to use properties of graphs to further examine the characteristics of interoperability. One such property is the idea of morphism, which is the study of the similarity between graphs. The idea of similarity is closely related to that of intersection in that similar things can be considered to share similar characteristics which constitute their intersection. As shown earlier, interoperability is the existence of a CRM, which is the intersection of the existential and transformational dependencies. Consequently, a study of the similarity between graphs is equivalent to a study of interoperability. This dissertation has shown that similarity has to be established between terms as of their meaning and between structures or forms. Similarity of terms is only possible if models share domains; therefore, this kind of similarity can be assumed. Interoperability in terms of a graph can then be reduced to the similarity between structures.

In graph theory similarity in structure is isomorphism. Two graphs G and H are isomorphic if there is an edge preserving morphism between G and H. Formally two graphs are isomorphic if there is a function:

$$f: G \rightarrow H \ and \ f \ is \ bijective$$

The definition of isomorphism leads to the following definition:

Definition 4.1: *Two models G and H interoperate if and only if they are isomorphic.*

This definition is consistent with previous theorems of interoperation as the existence of an isomorphism between the two models is equivalent to stating that they intersect. Interoperation between *N* models is defined as follows:

Definition 4.2: *A collection of models G_1, G_2,...,G_n interoperate is and only if they are isomorphic.*

Definition 4.2 is also consistent with the notion that interoperation is bounded. Simply stated, the degree of interoperation is the cardinality of the isomorphic class which is finite and bounded by the smallest and largest isomorphic set. Previously in this dissertation, interoperability has been defined as the intersection of the transformation dependencies between models. In term of a graph, if the set V of vertices is defined as the set of elements and the set *E* of edges as the set of transformation dependencies, definition 4.2 can be reformulated as follows:

Definition 4.3: *A collection of models G_1, G_2,...,G_n are interoperable if and only if they are isomorphic.*

Interoperability and interoperation are similar with the difference that interoperation is a necessary condition for interoperability. Interoperation and interoperability are simply the specification of a CRM and a set of rules governing interactions with the CRM. Based on these findings, the fundamental question of interest in studying interoperation and interoperability is:

The fundamental question of interoperability*: Given a modeling question and a set of models can a CRM be identified?*

This question is important because of the implications carried by the answer. If the answer is yes it means there is potentially an algorithm that could take the modeling question and models as inputs and provide the corresponding CRM. If this algorithm is efficient is terms of time and/or memory space, it would mean that interoperability can be solved in general at a reasonable cost. Conversely, if the answer is no and a CRM cannot be identified it would mean that the best that can be done is to engineer a solution that would be the closest to answering the modeling question. The degree of closeness is determined by the modeler by inspection or by a metric such as time, space or correctness. Current approaches to interoperability (common framework and common standard) assume that the fundamental question of interoperability is decidable and solutions can be constructed.

The questions as posed can be mapped to *decision problems* that are well known and have been studied in *Computational complexity theory.* Complexity theory is an area of mathematics and computer science that is focused on studying and classifying computational problems based on criteria such as time and resources required to provide a solution (Hein, 2002). A decision problem is a type of computational problem in which a yes-no answer is provided based on a given input (Hein, 2002). In terms of classifications a problem is said to be:

- *Polynomial(P)* if the answer to the question can be provided in polynomial time by a deterministic Turing Machine (computer).
- *Non deterministic polynomial (NP)* if the answer to the question can be verified in polynomial time.
- *NP-Complete* if the problem is in NP but there is no known efficient algorithm to solve it.
- *NP-Hard* if the problem is at least as hard as NP-Complete problems

The question of determining whether a CRM exist can be formulated as a decision problem in which the inputs are two or more models and the answer is *yes* they are isomorphic or *no* they are not. The determination of the existence of a CRM can be formulated as follows:

The fundamental question of interoperability*: Are two or more graphs isomorphic?*

This problem is known as the *graph isomorphism problem* for which it is not known whether it is in P or NP. A generalization of this problem known as the *subgraph* *isomorphism problem* in which the input is two graphs and the question is whether a subgraph in one is isomorphic to a subgraph in the other. The *subgraph isomorphism problem* is NP-Complete (Eppstein, 1999). As a result, for the general case, there is no known efficient algorithm to find the answer to the first question. That is to say, regardless of the approach taken to generate or identify a CRM, there is no known way to efficiently find the CRM. Consequently with respect to a computer, all current approaches are equivalent in that they are providing heuristics to obtain a CRM. In terms of interoperation and interoperability, it is worth noting that the decision problems addressed so far only focus on structure and as such are a subset of the general problem of interoperability. The question as to whether two or more elements are identical still needs to be answered. This question as discussed in section 3 is reduced to comparing two or more sets of strings as there is no algorithm that can determine whether two conceptualizations are the same. As a result, interoperation and interoperability are at least as complex as NP-Complex problems, which lead to the formulation of the following theorems:

Theorem 5.1: *The determination of whether models interoperate is at least NP-Complete.*

Theorem 5.2: *The determination of whether models are interoperable is at least NP-Complete.*

The overall complexity of interoperation and interoperability is possibly NP-Hard if one assumes an oracle that can identify equivalent elements instantaneously and an algorithm that resolves redundancies in polynomial time. This discussion is however out of the scope of the dissertation and will be revisited in future works.

In summary, this section formally defines interoperation and interoperability and shows that current approaches to both are equivalent but insufficient. Interoperation and interoperability are also shown to be at least as complex as NP-Complete problems. Consequently, a heuristic is necessary to identify a CRM that can help answer a given modeling question. In practice, the common language and the common standard approach are heuristics that provide a CRM but do not provide an algorithm to verify whether it is the actual CRM which is a requirement for NP-Completeness. The next section will provide an algorithm for CRM identification for a small number of models that is independent of the approaches reviewed in this dissertation and a way to quickly verify its correctness.

5 A HEURISTIC FOR CRM CONSTRUCTION: MODEL BASED DATA ENGINEERING

The results of the dissertation show that no general algorithm exists to generate a CRM for two models that shall interoperate. However, this does not exclude the engineering of solutions following common rules and guidelines. MBDE is the recommended solution. This section describes MBDE and discusses the benefits, shortcomings and assumptions as an engineering method. MBDE uses the same steps as Data Engineering, which was first introduced in (North Atlantic Treaty Organisation, 2002). While in Data Engineering the focus is on preparing a system for integration within a federation, MDBE shifts the focus to the CRM. As a result, in MBDE the system is replaced by the CRM which is considered as another system. MBDE, however, assumes the existence of a CRM whether it is selected or derived. Therefore, MBDE needs to be reformulated to describe how to capture a CRM rather than assume its existence. In order to do so, the notion of requirements, needs and capabilities are formally captured. The steps of MBDE are formally expressed as a series of algorithms that together form a heuristic that approximate a CRM for a relatively small number of models. The formalism described earlier in this dissertation is used along with the existence and uniqueness theorems.

5.1 Model Based Data Engineering

Model Based Data Engineering comprises four main steps as informally described in Data Engineering:

- *Data Administration*: The goal of data administration is to identify the format and physical location of data elements and their value domains when appropriate. For unstructured or semi-structured data a form of semantic enrichment is need resulting in structured data. Data Administration results in an unambiguous definition of data elements, a classification of data in terms of entities, properties and values or value domains, and documented data in the form of metadata. It is recommend both data and metadata be captured in a machine language such as XML Schema (XSD). The processes and artifacts of data administration are shown in Figure 4.

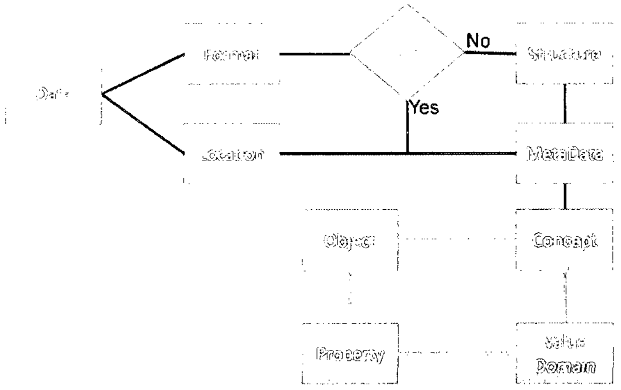


Figure 4: Data Administration Process and Artifacts

- *Data Management*: The goal of data management is to identify the logical relations between data elements and all relevant business rules including how to form meaningful grouping of elements otherwise known as composites (Tolk & Diallo, 2008). The application of data management results in the identification of all elements including their structure and context. Data management produces a logical data model that should also be captured in machine understandable language. The process and artifact(s) of data management are shown in Figure 5.

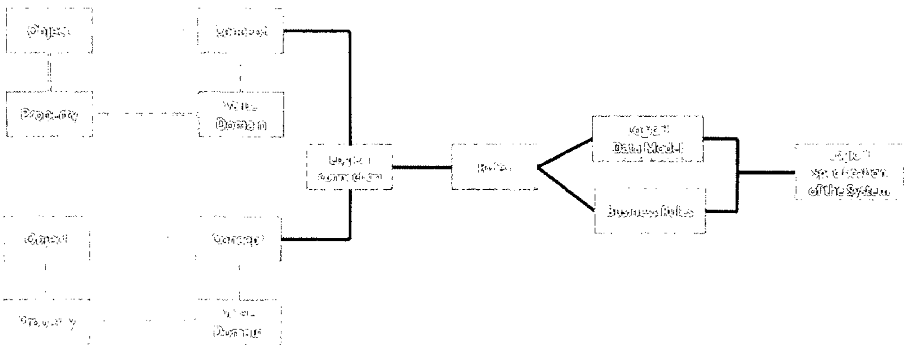


Figure 5: Data Management Process and Artifacts

- *Data Alignment*: The goal of data alignment is to identify the resolution and scope of data and identify gaps and variances between representations. Data alignment results in a list of resolution and scope issues that have to be resolved within the federation. The process overview of data alignment is shown in Figure 6.

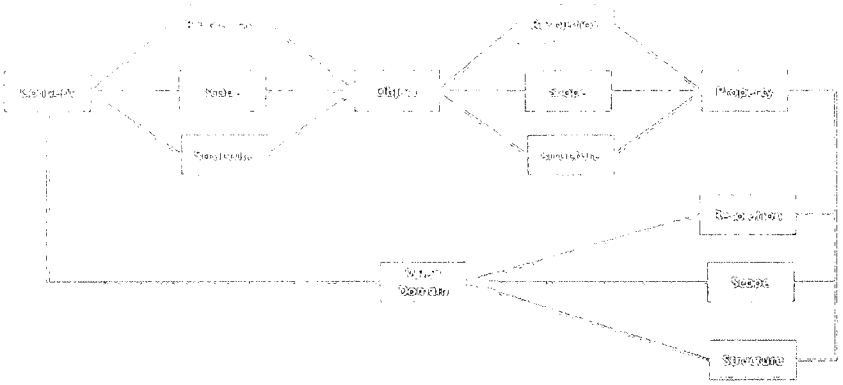


Figure 6: Data Alignment Process and Artifacts

- *Data Transformation*: The goal of data transformation is to identify and implement mapping functions that generate valid sentences of the system as specified in data management. Data transformation generates a computable model of the logical data model which is called the physical data model. Data transformation ensures consistency between the logical and physical views of data and serves as the basis for an implementation or a physical data model instance. The process and artifacts of data transformation are shown in Figure 7.

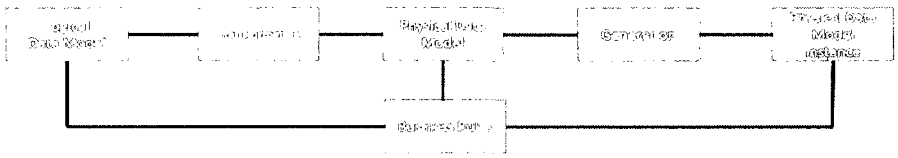


Figure 7: Data Transformation Process and Artifacts

While these steps are envisioned as linear, they are iterative in practice. The four processes as described here focus on preparing systems so that an evaluation of their ability to participate in a federation in support of a modeling question can be made.

In accordance with the findings summarized in the last section, this evaluation is only possible if the modeling question itself is formulated as a model. In this dissertation, this model is referred to as the *interoperability model.* The interoperability model is built using MBDE steps and represents a specification of the elements and business rules necessary to answer the modeling question. Based on MBDE, the interoperability model has a logical and physical representation and possibly a physical instance. Once the modeling question is specified as a model, the next task is to identify *candidate models* whose federation is equivalent to the interoperability model. It is important to note that another option is to implement the interoperability model and execute it directly which means to construct a new simulation systems representing the interoperability model. Since the focus of this dissertation is on interoperability, it is assumed that the federation option is selected. As demonstrated in the previous section, the fundamental activity of interoperability is the identification of a CRM. In order to formalize MBDE as a heuristic for defining a CRM, each step has to be formalized first.

5.2 Formal MBDE

Data Administration results in the identification of elements and their classification into entities, properties or values, as they were defined for the formal approach in section 3 of this work. Values can be grouped into value domains that represent the range of a property. In terms of the formalism presented in this dissertation:

Definition 5.1: *Data administration is the specification of the relations α, β, γ.*

A value domain is simply the set of symbols that belong to the same domain. With this definition however, a value domain cannot be assigned to an element as assignment is formally captured as a relation. This particular step has to be performed in data management. The format and location of data are captured by designating each format and location type as a domain and specifying α, β, γ relations where applicable. Formal data administration is presented in Figure 8.

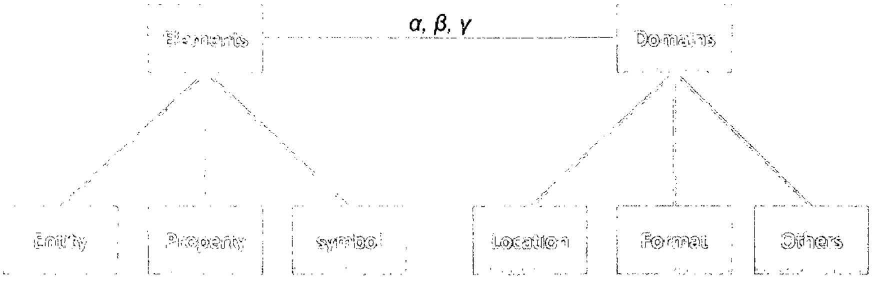


Figure 8: Formal Data Administration Process

Formal data administration adds the additional requirement of defining domains into traditional data administration. This definition ensures that elements are unambiguous because every element is uniquely defined in its domain, rather than relying on terms which are ultimately strings. However, in order to guarantee that domains themselves are uniquely named, the following rule is formulated:

Rule 1: *Every domain must be assigned a unique symbol that is not part of its domain.*

The fact that domains are uniquely named does not mean that they are considered unique in terms of their meaning. This excludes the existence of homonyms but not synonyms within the domain even though in MBDE every domain is considered to be unique. However, by the definition of a set, each member in the set of domains is unique. In practice, it means that a domain that is labeled twice following rule 1 results in two domains. Consequently, while there is no rule requiring domains to be unique in terms of their relation or meaning, the rule requires that each domain be addressable uniquely.

Data management is the specification of a logical data model and a set of business rules governing the construction of composites within the model. The following formal definition is proposed:

Definition 5.2: *Data management is the specification of* ρ *and* Φ*.*

The set ρ is the Cartesian product of domains and captures all meaningful composites. The set Φ of existential dependencies represents all valid statements within a model. Data management does indeed result in a logical model from the composition (α, β, γ, ρ) and the specification of its structure in Φ. The relations α, β, γ were already derived in the Data Administration process. Data management is described in Figure 9.

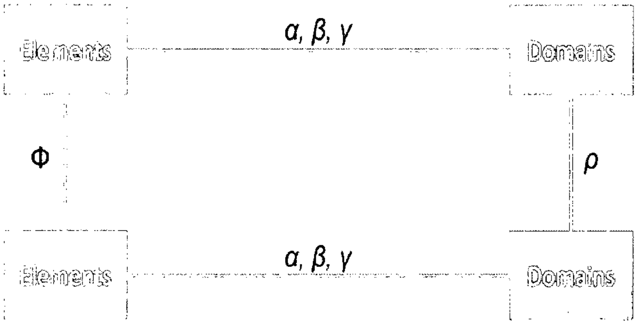


Figure 9: Formal Data Management

In data management some groupings can be made by partitioning the triple (α, β, γ) into ((α, β), (β, γ), (α, γ)) to denote the grouping of properties and entities *(hasProperty)*, the grouping of value domains and properties *(hasValue)* and the grouping of symbols and entities *(hasSymbol)*. For the sake of simplicity these relations will be referred to by the name in parentheses in the balance of this work. These groupings lead to the formulation of data alignment.

Data alignment identifies the resolution and scope of elements. Formally data alignment is defined as follows:

***Definition 5.3**: Data alignment is the specification of all groupings.*

The following simplification algorithm is proposed to simplify the groupings:

```
1. Begin
      a. For every  grouping set
           i.  For every element
                  1. Identify all related elements
                  2. Group properties into a new set
                  3. Delete   all   related   elements   from
                     grouping
                  4. Create new relation associating element
                     with new set
                  5. Delete element from grouping
                  6. Add new relation to grouping
2. End
```

This algorithm is guaranteed to terminate. For each iteration, the number of elements decreases non-monotonically by the number of tuples containing the current element. This algorithm yields the smallest number of groupings and is used to formally define scope and resolution:

***Definition 5.3.1**: The scope of an element is the number of unique groupings in which it appears.*

***Definition 5.3.2**: The resolution of an element is the number of elements within its scope.*

The scope of a model is by extension the union of the scope of its elements and the resolution of a model is the union of the resolution of its elements. The notions of scope and resolution are used later as a way to evaluate membership into the CRM. Figure 10 shows data alignment as described above with the domain added to show that by definition every element belongs to a domain. Data Alignment and data management are closely related but serve distinct purposes in MBDE.

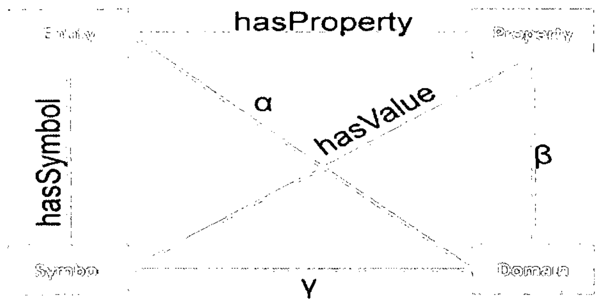


Figure 10: Formal Data Alignment

Data Management is centered on the model at hand whether it is the interoperability model or the candidate model. Data alignment shifts the focus to the federation. It is not a standalone process, unlike data management, in that it has to be performed with respect to another model. Consequently, data alignment constitutes the first step of the MBDE heuristic. That is to say, in order to answer the modeling question and specify a CRM, the initial step is data alignment. The full algorithm is presented after the formal definition of data transformation. Data transformation is the process by which the logical data model specified in data management is transformed into a computable or physical model. The following definition is proposed for data transformation:

Definition 5.4: *Data transformation is the specification of a set of functions and Θ the set of transformational dependencies.*

Data transformation implies the specification of a set of inputs, outputs and respective functions. This definition implies the transformation of the relations from data management into functions. This process is called normalization in Data Modeling Theory (Codd, 1971). The set of transformation dependencies represents the business rules by definition (definition 3.2). Transformational dependencies that are not computable are usually captured in documents but are limited to human consumption only. Figure 11 shows the process of data transformation from model to model as an aggregate and an individual process.

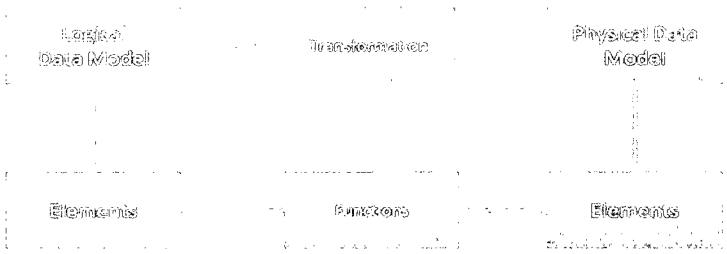


Figure 11: Formal Data Transformation

It is important to note that data transformation should be structure preserving and should not result in any additional elements and dependencies. In order to be more specific, additional rules are introduced:

***Rule 2**: Every relation is maintained during data transformation.*

***Rule 3**: An element preserves its set membership during data transformation.*

***Rule 4**: Every dependency (existential and transformational) is maintained during data transformation.*

***Rule 5**: Additional dependencies (functional, transformational) shall not be introduced during data transformation.*

These rules serve as guidance in determining how to resolve conflicts when there is more than one candidate function. If more than one function fulfills the rules, they are considered equivalent for a given transformation and either can be chosen. The rules also point to a practical issue that was raised in discussing mandatory and optional elements. Optional elements by these rules cannot be specified unless the conditions under which they are valid options are also specified which would make them mandatory under those options. As a result, if an element appears in the model it is mandatory by definition as it must belong to a set. Having formally defined the processes of MBDE, a heuristic for CRM construction is proposed.

5.3 A Heuristic to Generate a CRM

The heuristic proposed in this dissertation relies on formal MBDE as specified in the previous section. Parts of the heuristic can be automated while the formal MBDE and the specification of an interoperability model that captures the modeling question is mostly a human endeavor. The heuristic takes as input the interoperability model and a candidate model that is being evaluated for merging into a federation. The output is an approximate CRM or nothing if there is no CRM. If more than one model is required to answer a given part of the modeling question the heuristic can be applied by replacing the interoperability model with the CRM previously defined and another candidate model. If more than one model is needed to answer the modeling question as a whole, the heuristic can be applied by replacing the interoperability model with the subset for which the candidate model is being evaluated. The complete heuristic along with remarks is presented in the appendix of this dissertation and can serve as a best practice guideline in support of projects. Within this section, the steps and their motivation are described in form of an executive summary of the appendix.

The main approach in this heuristic is to first apply MBDE to create an interoperability model. Based on this model, a number of candidate models are selected to form a federation. The idea is that the federation of candidate models should replicate the interoperability model. There are two main possibilities to explore once these models are obtained. First, can a single model replicate the interoperability model? If the answer is yes, that model is selected and there is no need to federate. The next question is how many models are required to replicate the interoperability model. In order to answer that question, let's assume that the interoperability model can be partitioned into distinct but related parts. The question then becomes how many models are needed to replicate a given partition. The answer to that question in general is NP-Complete as shown in this dissertation. If it is possible to approximate the answer for every partition provided they are small and manageable, it is possible to take the union of those models capable of replicating those partitions as the ones most capable of approximating the interoperability model.

Once the interoperability model is specified, the heuristic assumes a small partition is selected even though it is formulated for the general case (the CRM as the input). The first goal is to determine whether the candidate model has some common domains with the interoperability model. If that is the case the domain along with its existential dependency set is selected to be a domain in the CRM. The reason for selecting the existential dependency set is to make sure that all possible relationships with elements are captured or to maintain the structural integrity of the candidate model. The domain might be independent in which case it is a trivial dependency. If no domains are common the candidate model is eliminated as it is impossible to ascertain in its current form whether it is capable of replicating the interoperability model. As a caveat, it is always possible to reengineer the candidate model or apply some series of transformation to arrive at the interoperability model, but the reader should think of the reengineered model as the candidate model, not the original one. Once the domains are identified, for each domain, the heuristic looks for elements defined similarly across equivalent domains. The heuristic eliminates elements that do not belong to the same groupings. Elements within the same groupings and same domain along with their existential dependencies sets constitute the initial elements of the CRM. After groupings are identified, elements that differ in scope and resolution are eliminated as a modeling decision is required to resolve the issue. The heuristic purposefully does not provide a resolution as this should be done on a case by case basis. With this step interoperation is now possible. In order to have interoperability, the transformational dependencies for each element have to be verified. If an element is independent or depends on an independent element in the CRM, it stays in the CRM. If the element depends on a dependant element or it depends on an element outside the CRM it is removed. That is not to say that this element cannot be exchanged;

it just means that an engineering process that ensures that the exchange does not lead to inconsistencies is required. As the candidate model needs additional information in order to replicate the parts or whole of the interoperability model those elements cannot be part of the CRM as-is.

The CRM thus constructed is still not the intersection of the models; rather, it is the parts of the interoperability model that the candidate model can satisfactorily fulfill. Through engineering approaches such as aggregation, transformation and filtering more elements can be introduced in the CRM but those processes are outside of the heuristic.

In summary, this section introduced MBDE as an engineering approach to interoperability. MBDE is formalized into formal MBDE through the formalism described earlier. The formal method is used to formulate a heuristic that allows the construction of an approximate CRM that is consistent and can be verified.

6 SUMMARY AND FUTURE WORK

The goal of this dissertation is to contribute to a general theory of interoperability for M&S. It is recognized that interoperability involves both data and processes. It was also shown that for M&S, the model based aspect is paramount. The dissertation focuses on data interoperability even though the formalism developed for data takes the effects of processes on data in the form of dependencies into account. In order to study interoperability, it is important to first establish how the term is defined and how it is understood in the state of the art. Informally, interoperability involves the exchange of information and the ability to use the information thereafter. This two part understanding of interoperability reflects two schools of thought: one defines interoperability as inherent to a system (a system can be built to be interoperable) and the other states that interoperability can only be evaluated with respect to other systems (systems are interoperable when they interoperate).

The notion of interoperability happening in levels or stages supports both schools and several models classifying levels of interoperability are reviewed. The consensus in the state of the art is that interoperability involves an exchange of information, the ability to use the information during or after the exchange and an agreement that interoperability happens in levels. Having established what interoperability is in the state of the art, this dissertation focused on how it is practiced. Again in this case two schools emerge: one consists of establishing standards and frameworks that enable interoperability and the other imposes a common model by which systems shall abide. The two schools in terms of enabling interoperability mirror the two schools in the definitional aspects of interoperability. The common framework approach is similar to the "interoperability as inherent to a system" school. A system can be built to support a given framework or standard (XML, HLA, DIS) and be called interoperable. The common model approach is similar to the "interoperability with respect to other systems" approach in that interoperability is defined as the ability of the system to fulfill the requirements of the common model. Both approaches seem to work; however, there is no way to verify that they are in general interoperating meaning that they are exchanging the information and using it.

In order to verify a candidate solution, it is necessary to first formally define interoperability and clarify what information exchange means and what usefulness is. Only then it is possible to answer other important questions such as how to measure interoperability, how to find an algorithm that solves it once and for all. Since this work is focused on M&S, this dissertation reviewed definitions of model and simulation based on seminal works in this area. This review showed that those two terms are also overloaded, and in terms of data at least, it is essential to formally define a model before attempting to define interoperability. As the focus is on data, the dissertation reviewed formal definitions of a data model found in data modeling theory. The two formal theories in data modeling theory are the relational model and the entity relationship model. Both models are shown to be equivalent and represent the basis for the state of the art in data modeling. The consensus from the ER model and the RM is that a model is a collection of relations. Both models capture the model as a collection of entities which have properties to which a value domain is assigned. In addition, entities can be related and relationships can be viewed as entities in that they can have properties and respective value domains. In terms of M&S, this dissertation abides by this view but shows that this is only one of many possible views. The decision to assign properties to entities for instance is a modeling decision meaning it is part of the abstraction process. In order to support all possible views a more general definition of a model is required. Furthermore, both models rely on terms to carry meaning which also reflects the one world view. In order to generalize, all possible meanings of a term must be supported in the definition of a term.

Essentially, this is recognition that a term in M&S is a model meaning it is an abstraction itself. The dissertation redefines the terms entity, property, and value domain as what results from the abstraction of real or imagined things and introduces the notion of elements, symbols and domains. A model is then defined as a composition of relations between elements, symbols, domains and domains with themselves. This definition captures the general case including the ER model and the RM. After defining a model, the dissertation introduces two types of dependencies between elements. Existential dependencies capture the idea that the existence of an element or group of elements depends on the existence of another element or group of elements. Transformational dependencies capture the notion that a change (creation, deletion, update) to an element or group of elements results in a change in an element or group of elements. Using these definitions, information exchange and use of information are formally specified. Information exchange or interoperation is the intersection of the existential dependencies between two or more models. Based on this definition, interoperation implies the existence of an intersection and vice versa. This theorem, known as the existence theorem, also formally describes the CRM as the collection of statement that are true or false in all models. Interoperation is necessary but not sufficient for interoperability. Interoperability is interoperation and the existence of an intersection between the transformational dependencies of two or more models. Based on these theorems, this dissertation shows that interoperation and interoperability have no direction and they are both bounded. Interoperation and interoperability are formally the specification of a CRM between two or more models. These definitions are consistent with the state of the state of the art and reflect both schools of thought in the definitional aspect as well as in the approach to interoperability.

The fundamental question of interoperability is formulated: given a set of models what is the CRM? From a definitional standpoint, one can argue that given a CRM a modeler can always develop a model that fulfills it; this is assuming a CRM is provided and the model has not been developed yet. The common model approach takes this viewpoint by providing a CRM and asking modelers to engineer interfaces that will allow their model to conform to it. The common framework approach on the other hand take the fundamental question head on by providing an environment that enables interoperation then verifying to what degree interoperability can be supported. In this dissertation, the answer to the fundamental question of interoperability is shown to be at least NP-Complete, meaning there is no known algorithm that can answer the question in a reasonable amount of time. This assertion is true for both interoperation and interoperability. Consequently all approaches to interoperability are equivalent in that they are heuristics. Based on this observation, a formal heuristic that is better than trial and error and can be quickly verified is required. MBDE is formalized using the definitions and theorems developed in this dissertation. This formal MBDE is then used to develop a heuristic that can eliminate candidate models and approximate the CRM given a modeling question. The notion of a modeling question is very important in that the state of the art considers it as an informal specification. In this dissertation, the modeling question is at the center of the heuristic and drives what models can and cannot participate in a given federation. The heuristic is a collection of algorithms that are shown to terminate, and the resulting CRM is consistent and quickly verifiable. Overall, this dissertation contributes a critical review of the state of the art resulting in a classification of interoperability and approaches to interoperability. It provides a formal definition of data in M&S systems based on theories developed in data modeling theory, formally defines interoperation and interoperability as two separate but dependent concepts and shows that all approaches to interoperation and interoperability are equivalent in general.

In terms of contributions to the body of knowledge, this dissertation provides a method for theory building and testing. The theory is built from the body of knowledge and tested through its ability to explain existing theories, methods and approaches and also though its ability to generate new findings. This dissertation contributes a formal model of data that generalizes existing formal models and used it to formally define interoperability and introduce the notion of interoperation. This dissertation also shows that current approaches are insufficient because they do not take into account dependencies, which have an impact on how interoperability projects are conducted and verified and how future standards might evolve. This contribution ties directly into the fact that interoperability is at least NP-complete which means that current approaches are equivalent in that they are heuristics. This dissertation contributes a heuristic to enable interoperability. The heuristic takes into account dependencies and provides a solution that can easily be verified as opposed to existing heuristics. This heuristic is based on MBDE which was an informal engineering approach and is now formalized. The focus of this work was to study interoperability from a computational standpoint, but the results can be generalized beyond this area.

As for future work, a rigorous algebra of interoperability is needed. There are several operations (insert, delete) that can be safely performed on a model as it stands alone but lead to deadlock situations in interoperability. The MBDE heuristic prevents this from happening initially; however, it does not guarantee that cascading effects might not lead to a deadlock situation where elements are mutually dependent in a circular way (A depends on B which depends on C which depends on .... which depends on A) at the N^{th} level when a new model is introduced. Such an algebra will move the community a step closer to semi-automated dynamic interoperability as defined in the LCIM.

The complexity of the algorithm presented here is polynomial. However, there are several optimization algorithms that have been proposed in the related fields of Mathematics and Computer Science. As future work, those algorithms could be studied to identify their applicability to M&S and how they affect the CRM. If a reasonably good optimization is applied, how does the CRM evolve and how far away does it move from the modeling question? The satisfaction of the functional and transformational dependencies of candidate models might, in fact, overwhelm the modeling question by introducing a growing list of requirements that need to be satisfied before the candidate model can participate. The optimization algorithm might find a model that fits the modeling question completely but requires N models itself in order to answer the modeling question. This observation introduces the need for a study of the requirements for a best fit. Possible candidate might be the model with the minimum list of dependencies, adaptation time, closeness to the modeling question etc. This dissertation does not delve into these important facets of the interoperability problem; however, the answer to this question is in general NP-Complete.

An implementation of the heuristic using the semantic web should be studied. This study should focus on what additional metadata are required to compose and orchestrate web services so that they represent the CRM. In addition to the identification of metadata, a description of model functionally is required to aid in determining equivalent abstractions. The important question in this case is to determine whether two or more elements described in the same manner are the same. The answer to this question is in general NP-Complete; however, it is possible to devise a heuristic that makes it possible to answer the question.

REFERENCES

Ackoff, R. (1989). From Data to Wisdom. *Journal of Applies Systems Analysis , 16*, 3-9.

Benjamin, P., Akella, K., & Verna, A. (2007). Using ontologies for simulation integration.

Blais, C., Brutzman, D., Drake, D., Moen, D., Morse, K., & Tolk, A. (2004). *Extensible Modeling and Simulation Framework (XMSF).* Monterey, CA.

Brutzman, D., Zyda, M., Pullen, J. M., & Morse, K. L. (2002). *Extensible Modeling and Simulation Framework (XMSF).* Monterey, CA.

Chen, P. P.-S. (1976). The Entity-Relationship Model-Toward a Unified Data View. *ACM Transacstions on database Systems , 10* (3), 9-36.

Clark, D., Numrich, S., Howard, R., & Purser, G. (2001). Meaningful Interoperability and the Synthetic Natural Environment. *Proceedeings of the 2001 Simulation Interoperability Workshop.* London Middlesex: IEEE CS Press.

Codd, E. F. (1970). A relational Model of Data for Large Shared Data Banks. (P. Baxendale, Ed.) *Communications of the ACM , 13* (6), 377-387.

Codd, E. F. (1970). A Relational Model of Data for Large Shared Data Banks. (P. Baxendale, Ed.) *Communications of the ACM , 13* (6), pp. 377-387.

Codd, E. F. (1979). Extending the Database Relational Model to Capture More Meaning. *4* (4), 397-434.

Codd, E. F. (1971). Further Normalisation of the Data Base Relational Model. In R. J. Justin (Ed.), *Courant Computer Science Symposia Series. 6.* New Yorf City: Prentice-Hall.

COI Resources. (2008, October 29). Retrieved October 29, 2008, from The United States Department of Defense: http://www.defenselink.mil/cio-nii/coi/links.shtml

Connors, C., & Malloy, M. A. (2007). *Practical Challenges Facing Communities of Interest in the Net-Centric Department of Defense.* The MITRE Corporation.

Dahmann, J., Salisbury, M., Barry, P., Turrell, C., & Blemberg, P. (1999). HLA and Beyond: Interoperability challenges. *Information and Security* , 25-42.

Darwen, H., & Date, C. J. (1995). The third manifesto. *ACM SIGMOD Record , 24* (1), 39-49.

Date, C. J. (1984). A critique of the SQL database language. *ACM SIGMOD Record*, *14* (3), 8-54.

Davis, P. K., & Anderson, R. H. (2004). Improving the Composability of DoD Models and Simulations. *1* (1), 5-17.

Davis, P. K., & Bigelow, J. (1998). *Multi resolution modeling.* Santa Monica: RAND.

Davis, P., & Anderson, R. H. (2004). *Improving the Composability of Department of Defense Model and Simulations.* Santa Monica : RAND National Defense Research Institute.

Department of Defense. (2008). Retrieved 10 2008, October, from http://www.dtic.mil/doctrine/jel/doddict/data/i/02802.html

Diallo, S. Y., Tolk, A., & Weisel, E. (2007). Simulation Formalisms:Review and Comparison of Existing Definitions of Key Terms. *Fall Simulation Interoperability Workshop.* Orlando: IEEE CS Press.

DoD Metadata Registry. (2008). Retrieved 10 2008, October, from http://metadata.dod.mil/

Dori, D. (2002). *Object-Process Methodology - A Holistic Systems Paradigm.* New York: Springer.

Eppstein, D. (1999). Subgraph Isomorphism in Planar Graphs and Related Problems. *Journal of Graph Algorithms and Applications* , 1-27.

Fagin, R. (1977). Multivalued dependencies and a new normal form for relational databases. *ACM Transactions on Database Systems* , *2* (3).

Fisher, M. C. (1994). Aggregate level simulation protocol (ALSP) managing confederation development. *Winter Simulation Conference* (pp. 775-780). Orlando: Society for Computer Simulation International.

Fortnow, L., & Homer, S. (2003). A Short History of Computational Complexity. *Bulletin of the EATCS 80* , 95-133.

Gridwise Interoperability Workshop. (2007). Dallas, TX.

Hein, J. L. (2002). *Discrete Structures, Logic, and Computability* (2nd Edition ed.). Jones and Bartlett Publishers, Inc. , USA.

Hieb, M. R., Pullen, M. J., Sudnikovich, W. P., & Tolk, A. (2004). Extensible Battle Management Language (XBML): A Methodology for Web Enabling Command

and Control for Network Centric Warfare. *Command and Control Research and Technology Symposium, CCRTS* .

Hofmann, M. A. (2004). Challenges of Model Interoperation in Military Simulations. *SIMULATION* , 659-667.

IEEE. (1990). *A Compilation of IEEE Standard Computer Glossaries.* New York: IEEE Press.

IEEE. (2002). *Distributive Interactive Simulation.* IEEE Press.

IEEE. (2000). *High Level Architecture.* IEEE CS Press.

ISO/IEC. (2003). *International Technology for Learning, Education, and Training.* Geneva: ISO.

(2007). *Joint Consultation, Command and Control Information Exchange Data Model.* Greding: NATO MIP.

Lasschuyt, E., van Hekken, M., Treurniet, W., & Visser, M. (2004). *How to Make an Effective Information Exchange Data Model or The Good and Bad Aspects of the NATO JC3IEDM.* Symposium, The Hague.

Lipski, J. W. (1979). On semantic issues connected with incomplete information databases. *ACM Transactions on Database Systems (TODS* , *4* (3), 262-296.

North Atlantic Treaty Organisation. (2002). *NATO Code of Best Practice.*

Online, W. (n.d.). Retrieved from http://www.merriam-webster.com/.

Overstreet, C., Nance, R., & Balci, O. (2002). Issues in Enhancing Model Reuse. San Antonio.

Page, E. H. (2007). Theory and Practice for Simulation Interconnection: Interoperability and Composability in Defense Simulation. In P. A. Fishwick, *CRC Handbook of Dynamic System Modeling.* CRC Press.

Page, E. H., Briggs, R., & Tufarolo, J. A. (2004). Toward a Family of Maturity Models for the Simulation Interconnection Problem. *Spring Simulation Interoperability Workshop.* IEEE CS Press.

Peckham, J., & Maryanski, F. J. (1988). Semantic Data Models. *ACM Computing Surveys* , *20* (3), 153-189.

Perme, D., Tran, H., Tedesco, B., Pandolfo, C., Diallo, S., & Tolk, A. (2007). Joint Event Data Initialization Services (JEDIS)- Implementing a Service Oriented

Architecture for Initialization. *Spring Simulation Interoperability Conference.* Norfolk: IEEE Press.

Petty, M. D., & Weisel, E. W. (2003). A composability Lexicon.

Renner, S. A. (2001). A Community of Interest Approach to Data Interoperability. *Renner, 2001) Scott A. Renner: A Community of Interest Approach to Data Interoperability. Proceedings Federal Database Colloquium '01.* San Diego.

Rowley, J. (2007). The wisdom hierarchy: representations of the DIKW hierarchy. *Journal of Information Science* , 163-180.

Schmid, H. A., & Swenson, R. (1975). On the semantics of the relational data model. *Proceedings of the 1975 ACM SIGMOD international conference on Management of data* (pp. 211-223). San Jose: ACM New York, NY, USA .

SISO Base Object Model Product Development Group. (2006, March 31). *Base Object Model Standards.* Retrieved February 4, 2010, from Bse Object Models: http://www.boms.info/spec/BOM_Specification_v2006_FINAL.pdf

Smith, J. M., & Smith, D. C. (1977). Database abstractions: Aggregation. *ACM Transactions on Database Systems , 2* (2), 105-133.

Sowa, J. (2001, July 28). *Knowledge Representation.* Retrieved February 8, 2010, from http://www.jfsowa.com/ontology: http://www.jfsowa.com/ontology/toplevel.htm

Spaccapietra, S., Parent, C., & Dupont, Y. (1992). Model Independent Assertions for Integration of Heterogeneous Schemas. *Very Large Database (VLDB)* , 81-126.

Tolk, A. (2005). An Agent-based Decision Support System Architecture for the Military Domain. In G. E. Phillips-Wren, & L. C. Jain, *Intelligent Decision Support Systems in Agent-Mediated Environments* (pp. 187-205). IOS Press.

Tolk, A. (2006). What comes after the semantic web, PADS Implications for the Dynamic Web. *PADS'06* (pp. 52-62). Singapore: ACM/IEEE/SCS.

Tolk, A., & Diallo, S. (2008). Strong and Weak Composites for Adaptable Web Languages. *Spring Simulation Interoperability Workshop.* Newport.

Tolk, A., & Muguira, J. (2003). The Levels of Conceptual Interoperability Model (LCIM). *IEEE Fall Simulation Interoperability Workshop.* IEEE CS Press.

Tolk, A., Diallo, S. Y., & Turnitsa, C. D. (2007). Applying the Levels of Conceptual Interoperability Model in Support of Integratability, Interoperability, and

Composability for System-of-Systems Engineering. *Journal of Systemics, Cybernetics and Informatics* , 65-74.

Tolk, A., Diallo, S. Y., King, R. D., & Turnitsa, C. D. (2009). A Layered Approach to Composition and Interoperation in Complex Systems. In A. Tolk, & L. C. Jain, *Complex Systems in Knowledge based Environments: Theory, Models and Applications* (pp. 41-74). Berlin: Springer.

Tolk, A., Diallo, S., Turnitsa, C., & Winters, L. (2006). Composable M&S Web Services for Net-centric Applications. *Journal of Defense Modeling & Simulation* , 27-44.

Tolk, A., Turnitsa, C., & Diallo, S. (2007). Model-based Alignment and Orchestration of Heterogeneous Homeland Security Applications Enabling Composition of System of Systems. *Winter Simulation Conference* , 842-850.

Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind* , 433-460.

Turnitsa, C. (2005). Extending the Levels of Conceptual Interoperability Model. *IEEE Summer Computer Simulation Conference.* IEEE CS Press.

United States department of Defense. (2002, November 4). *TENA Interoduction Documentation.* Retrieved February 4, 2010, from Test and Training Enabling Architecture: https://www.tena-sda.org/display/intro/Documentation

Webster Online. (2008, January 10). Retrieved October 10, 2008, from Webster Online: http://www.webster-dictionary.org/definition/Interoperability

Webster Online. (2008, January 10). Retrieved October 10, 2008, from Webster Online: http://www.webster-dictionary.org/definition/Interoperability

Weisel, E. (2004). *Models, Composability, And Validity.* Old Dominion University.

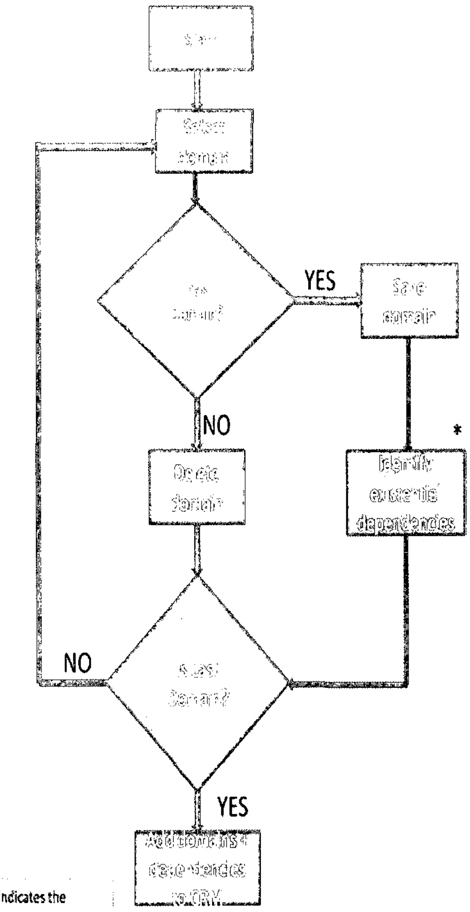
World Wide Web Consortium. (2004). Retrieved October 10, 2008, from http://www.w3.org/2004/OWL/

World Wide Web Consortium. (2008). Retrieved 10 2008, October, from www.w3c.org

Zaniolo, C. (1982). A New Normal Form for the Design of Relational Database Schemata. *ACM Transactions on Database Systems* , *7* (3), 488-489.

Zeigler, B. P., Praehofer, H., & Kim, T. G. (2004). *Theory of Modeling and Simulation. Integrating Discrete Event and Continuous Complex Dynamic Systems.* Academic Press.

APPENDIX: MBDE HEURISTIC

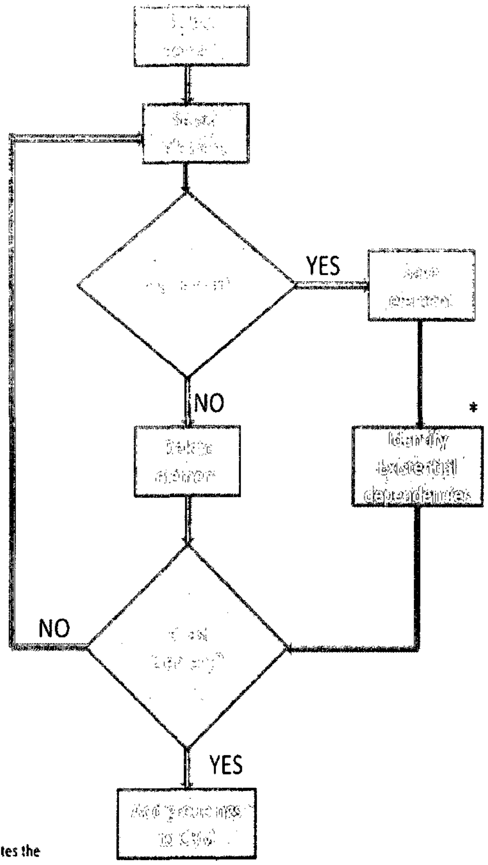


The "*" symbol indicates the algorithm continues from here

Remark 1: The input is the set of domains from the interoperability model and the set of domains from the candidate model. Both models are assumed to have gone through the MBDE process.

Remark 2: Since domains are uniquely named, the assumption is that elements are named using a common language such as English. The common language can be provided as the initial domain identifying the valid strings of the language. Synonym and homonym sets are relations of the domain unto itself (English to English).

Remark 3: This process captures the domains that are common to the models meaning elements that can be exchanged. The process leads to trivial interoperation if one assumes trivial existential dependencies (elements depend only on themselves). This is why it is crucial to capture the existential dependencies.



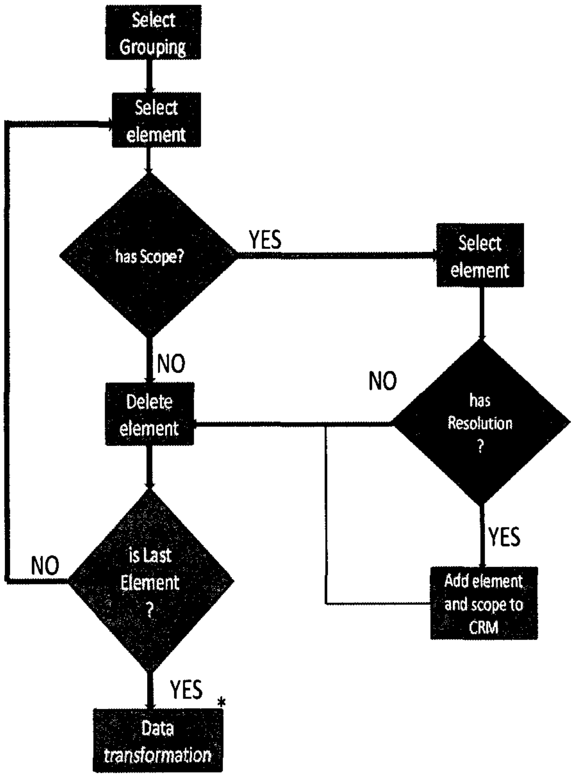
The "*" symbol indicates the algorithm continues from here

Remark 4: The input is the set of domains that are common between the interoperability model and the candidate model. The process is to compare domains across the models

Remark 5: This process is iterated for every element of every domain until all common elements are found. The process can start then with finding the hasProperty groupings, then the hasValue and hasSymbol groupings but any combination leads to the same outcome.

Remark 6: Elements that have no existential dependencies are the immediate candidates for the CRM as they fulfill the requirement for interoperation. Interoperability is brought in through transformational dependencies.

Remark 6: Elements that have existential dependencies outside of the domains selected require additional engineering if they are to make it to the CRM.

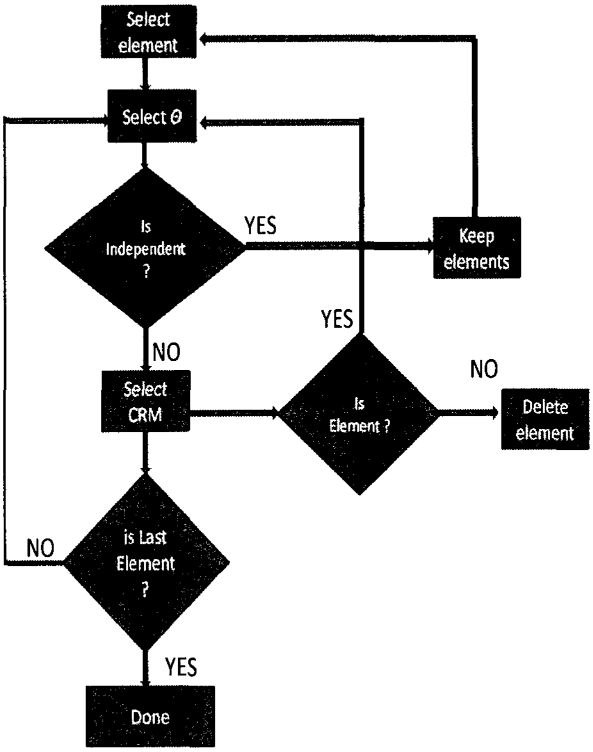


The "*" symbol indicates the algorithm continues from here

Remark 7: The inputs are the set of *groupings from the candidate model* and the set of groupings from the interoperability model.

Remark 8: Elements that have the same resolution and scope are the members of the CRM. The heuristic excludes elements that have a different scope as additional engineering is required to make a *decision on their inclusion. This* additional engineering process is a modeling decision. Elements that have the same scope but different resolution are also excluded for the same reason.

Remark 9: This process guarantees interoperation not interoperability. The CRM defined here is larger than the intersection of the two models for non trivial interoperation but it provides a model for which interoperation can be verified.



Remark 10: The inputs are the CRM *and the set of transformational* dependencies. The goal is to ensure that the candidate model can satisfy all transformation dependencies and also avoid circular dependencies that can be introduced by the interoperability model.

Remark 11: Elements that are independent are kept. Elements that that depend on other elements within the CRM have to independent in order to be kept. All circular dependencies are removed from the CRM.

Remark 12: This process guarantees interoperability. The CRM defined here is not necessarily the best. However it is the most consistent and it is verifiable. Additional engineering is still required to resolve multi resolution and scope issues. As mentioned before those are modeling decisions.

VITA

SAIKOU Y. DIALLO is a Senior Project Scientist 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 research focuses on the domain of Model Based Data Engineering and Web Services for M&S applications.