Research Proposal: Theory of Algebraic Foundation for Manipulation of DEMO Models

Towards Analysis and Synthesis of DEMO models

Tetsuya Suga

Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan
suga.t.ac@m.titech.ac.jp

Summary. The spread of business process management has induced the growing size of business process models and distributed executions. Meanwhile, an enterprise is more likely to be considered as a human activity system and often redesigned by aggregating several systems into one. These contexts underscore the importance of establishing techniques and theoretical supports for processing large models; it is also in demand to analyze requirements for large enterprise models that are capable of being handled with ease; namely split and merged. This Ph.D. project with two years left aims at shedding light on mathematical structures observed in DEMO models and manipulations of DEMO models in the framework of Universal Algebra, namely in the form of Boolean algebra. Extensive investigations are focused on algebraic characteristics such as closedness, commutativity, and associativity. It is also planned to develop preliminary implementation towards computer-aided manipulations and instantiate them on specific real-life cases for evaluating validity. The author hopes to formulate theoretical foundations and preliminary implication which may actualize computer-aided design automation in enterprise engineering.

Key words: model manipulation, model synthesis, formal study, computer-aided design, design automation, business process management, DEMO, enterprise ontology, enterprise engineering

1 Introduction

Since humankind invented the division of labor, a task may have been divided into several activities and performed by different persons or groups of people. Although the division of labor is taken for granted, it is still a challenging subject for the management as coupled with environmental changes of our modern society. The practices of the division of labor, i.e. how labor is divided, have been captured typically in a form of a business process—defined as *a collection of inter-related events, activities and decision points that involve a number of actors and object, and that collectively lead to an outcome that is of value*

16th CIAO! Doctoral Consortium

to at least one customer [1]—which can fall within the broad context of enterprise engineering [2] as dwelt on afterwards. In recent years, business processes handled in the life-cycle of business process management have grown in size. It is observed in two different ways: intra-organization and inter-organization. In intra-organization perspective, as a survey [3, p. 23] shows, more organizations are along the line of business process orientation while more enterprise-wide process management is observed within each organization. These trends accelerate the growth of process models in size within each organization. At the same time, initiatives toward inter-organizational business process collaboration is another trend that hastens the growth to the next level.

1.1 Problem Statement

In the environment as mentioned above, several people who have their own specialized areas, hence have different scopes of interest, share one process model, which is referred to as a global process model—defined as a process model that covers the whole scope of modeling. Due to its size, the global model is likely to be too complex for one person to understand even if its parts are simple enough respectively. One very natural solution is to take an arbitrary part of the global process model and produce a partial process model—splitting the large global process model into meaningful conceivable-sized fragments with each separate diagram. The production of partial process models is performed not only by the size but also distributed execution environments. The literature takes up this phenomenon under the name of "process fragmentation", in which a process fragment is defined as *an arbitrary subset of the process elements comprised within a process model* and the fragmentation is performed for a variety of reasons [4].

For instance, let us introduce a business process model of a supply chain. The procurement department only needs a part of the global process model that is focused on parts of the process between suppliers and assembly factories. On the other hand, the sales department would like to concentrate on parts of the process from warehouses through to stores. Put differently, the execution of the process is distributed into two departments. Therefore, both departments may produce their own partial models by excerpting processes with a bearing on their scope of responsibility. Rationales behind the scene are often for the sake of simplicity and understandability by throwing away things out of interest, but sometimes for serious practical concerns such that the model is too large to be printed on a single sheet of paper. In the case of inter-organizational modeling, the global model is often created by merging several partial models.

Provision of partial models along with the global model leads to a usage of model manipulation. In the scope of this research, a "model manipulation", or more precisely "manipulation over model(s)", means making one or more new models from one or more existing models. For example, [5] presents a basic set of manipulations such as merge (two models are merged into one model) as split (one model is split into two models). These manipulations over models that may have a certain structure are strongly associated with mathematical operations.

Although methods and techniques for merging and splitting business processes have been studied to some extent, mathematical examination and justification behind those methods and techniques seem to be as yet unknown as elaborated in Section 2.2.

Although the object of the study has been business processes so far, business processes are indeed just one limited aspect of enterprises. To overwhelm inadequate strategy implementation, it is necessary to achieve *coherence and consistency among the various components of an enterprise, which precludes it operating as a unified and integrated whole* [6, p. 7]. The need for a new approach has been answered by the emerging discipline of enterprise engineering. Therefore, this research takes enterprise engineering as the boundary of this study.

Although some studies on model manipulation in business process engineering are found in the literature, there exist very few studies on model manipulations in enterprise engineering, and the most of them still remain in the scope of the real world or semantic models at most. Therefore, these insights lead to engendering following problems.

- Currently, model manipulation requires a full load of manual work for the manipulation itself and following model verification procedures. The validity of model manipulations is usually reserved by decent efforts and huge considerations of experienced professionals in a sense of both notational syntax and semantic correctness.
- The current state of the art cannot tell a clear separation of "what is true in general" (context-independent, or case-independent; true for Volley tennis club, Pizzeria, Library, EU-Rent, and virtually any DEMO models currently existing or to be made in the future) from "what is not" (context-dependent, or case-dependent; not true in general, but true only for a specific case). Whereas the notion of context-independent will be mentioned in the framework of model theory in Section 2, the author believes that there may exist statements (theorems) that are context-independent.
- If the model manipulations are performed by non-experts, the verification would be insufficient.
- Moreover, the problem stated so far prevents us from hiring computing technology for computer-aided design in model manipulations.

1.2 Research Question

In order to solve the problems identified in the previous section, this research will draw attention to the mathematical structure of DEMO, which is a methodology aligned with the concept of enterprise engineering. Although DEMO is not the only option among many frameworks and methodologies in enterprise engineering, the author believes DEMO is a good option for challenging problems in enterprises.

Question 1. What is the mathematical structure of DEMO models? Put differently, what kind of mathematical structure is appropriate to be defined for examining model manipulations?

In this research, a term “mathematical structure” means *a set (or sometimes several sets) with various associated mathematical objects such as subsets, sets of subsets, operations and relations, all of which must satisfy various requirements (axioms)* [7]. As elaborated in Section 2, the mathematical structure embedded in DEMO models has been neither analyzed nor defined in the past. Therefore, this research has to answer this question at the very outset by studying the mathematical structure that apparently exists in the current modeling standard. The answer to this question shall formulate the domain of interest.

Question 2. How and to what extent are manipulations in the real world defined in the structure?

After getting the answer from Question 1, the next question is how the manipulations in the substantial world such as merging and splitting can be expressed as operations on the domain. For instance, given the natural numbers $\mathbb{N}$ as the domain, the manipulation which adds two natural numbers is expressed as an operation $+$. In other words, the operation $+$ is defined in that way.

Question 3. Is DEMO model is well-formed? If not, what is the requirement for DEMO models to be well-formed?

When a pair of domain and operation(s) are determined, it is of special interest to investigate whether the pair is algebraically well-formed or not. In this article, the term *well-formedness* means a set of algebraic properties that provides positive benefits such as closedness, commutativity, and associativity.

This research question can be rephrased in practical contexts: A practical concern rooted in the closedness is that when one is about to merge two partials models of a DEMO model, is there any possibility for him or her to get a *broken* model, which does not fit the requirements of DEMO models. For commutativity, when one is about to merge two partial modes, does the order of manipulation matter the result?; in symbol $A + B \stackrel{?}{=} B + A$. A practical concern rooted in associativity can be more tangible in that when one is about to merge three partial models (namely A, B, and C), is there any possibility to get different results when he or she merges A and B first, and when he or she merges B and C first?; in symbol $(A + B) + C \stackrel{?}{=} A + (B + C)$. If this research may be able to answer this question in Yes, users of DEMO do not have to worry about these concerns anymore.

This question is of importance because it is too optimistic to take these characteristics for granted for DEMO models without any investigation. Although they hold in a simple setting such that natural numbers and the addition operator, it is indeed rather rare that those characteristics hold. If we consider a slightly complex mathematical structure and operation such as 2×2 matrices and the matrix product, the commutativity does not hold: $\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right)\left(\begin{smallmatrix}5&6\\7&8\end{smallmatrix}\right) \neq \left(\begin{smallmatrix}5&6\\7&8\end{smallmatrix}\right)\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right) \because \left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right)\left(\begin{smallmatrix}5&6\\7&8\end{smallmatrix}\right) = \left(\begin{smallmatrix}19&22\\43&40\end{smallmatrix}\right), \left(\begin{smallmatrix}5&6\\7&8\end{smallmatrix}\right)\left(\begin{smallmatrix}1&2\\3&4\end{smallmatrix}\right) = \left(\begin{smallmatrix}23&34\\31&46\end{smallmatrix}\right)$. In fact, these concerns are overlooked in prior studies as described in Section 2.2.

Question 4. To what extent does model manipulation in DEMO comply with the behavior of system boundary?

Since model manipulations are virtually manipulating the original systems through models, issues on the system boundary are germane to model manipulation. As introduced in [8, p. 59], DEMO has its root in the definition of a system by Bunge. He also introduced the concept, definition, and formulation of system boundary with a topological setting [9]. The definition of system boundary was criticized and an alternative definition is put forward in [10] in the light of the concept of boundary in lattice. However, the behavior of boundaries in the procedure of splitting and allying systems apparently has awaited further discussion. This study also includes an investigation at this point.

1.3 Structure of This Article

The remainder of this article is organized as follows: Section 2 investigates the literature to identify the current state of research on DEMO and in surrounding areas. Section 3 states the proposed solution by this research with statements on research methodology and contribution. Finally, Section 4 reports on the current state of the research with a few achievements that are already produced and the work plan in the future.

2 Literature Review

Prior to heading to reviewing of the literature, some concepts and terminologies are revisited: real-world, model, semantics, and syntax. The following descriptions are summarized in Figure 1 on page 6.

There exists the real world and the model world in systems engineering. “Models are simplified representations of real world systems. [11, p. 44]” “A model is an abstraction of a (real or language-based) system allowing predictions or inferences to be made. [12, p. 370]” The three characteristics of a model are described as mapping, reduction, and pragmatism by [13, p. 157][1]. Therefore, the real world and the model are two distinct objects. Models are often classified into physical, schematic, verbal, logical, and mathematical models.

Model[2] theory introduces the concept of semantics and syntax. Model theory is the mathematical study on classes of mathematical structures such as groups, graphs, and universes of set theory. The term *model* included in the name of *model* theory means “a *model* of a *theory*“ that is a structure which satisfies the all of the sentences of that *theory*. *Model* theory captures a corresponding parallel relationship between semantical statements (having meanings and true/false) and syntactical statements (formulas and provable/unprovable). The

[1] The original text in in German. Characteristic, mapping, reduction, and pragmatism are translated from German word *smerkmal*, *abbildung*, *verkürzung*, and *pragmatisches* respectively.

[2] The notion of *model* in *model* theory is of a different kind than the one in systems engineering and rather the two concepts have little to do with each other even though they are spelled in the same way.

point is that syntactical statements are evaluated only by the means of symbol-manipulation in logic; whether the statement is provable or not by symbol-manipulations. If the statement is provable only by symbol-manipulations, it is said to be context-independent. The semantics can be expressed in formal languages and natural or informal languages such as diagrams. Establishing a formal semantics from natural or informal languages is another research topic. Formal semantics is helpful for bridging the semantics to syntax. In order to clarify whether a mathematical statement is semantical one or syntactical one, this article marks semantics in the sans-serif font ($AaBbCc$) whereas syntax in the sans fonts (AaBbCc).

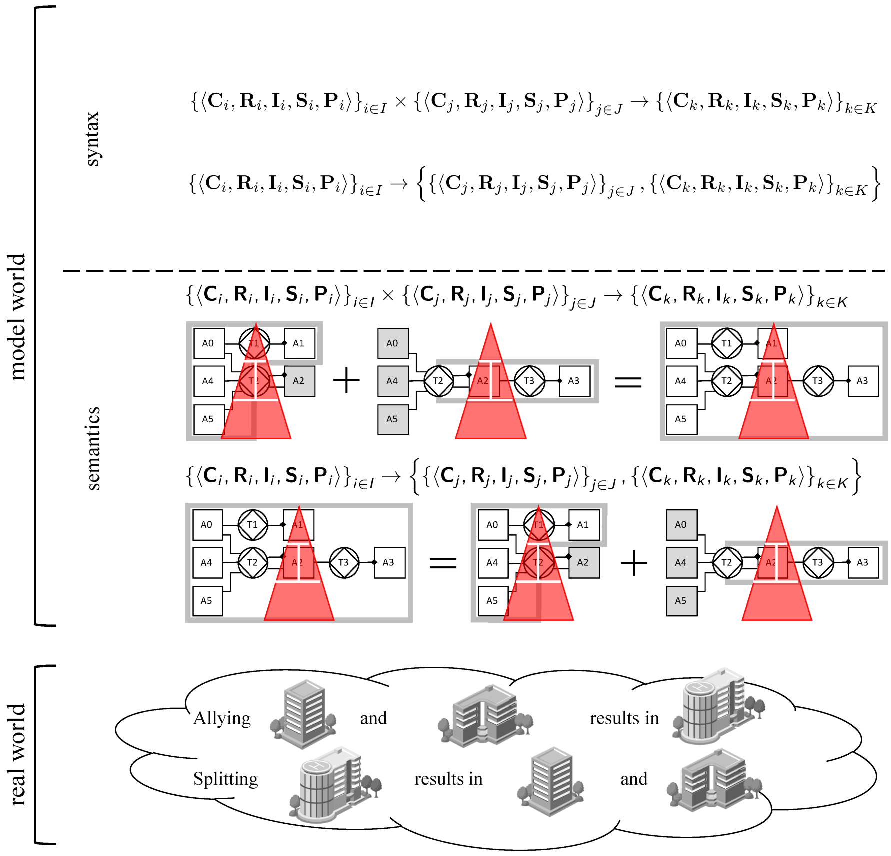

Fig. 1. Illustration of Real-World, Model, Semantics, and Syntax. [3]

[3] Image (buildings) courtesy of http://artshare.ru/

2.1 DEMO-related Studies

This section investigates how issues of model manipulations are mentioned in DEMO and DEMO-related studies.

DEMO & EE-theories Although DEMO exhibits a variety of implications and contributions, one of them highlighted in the context of this research is that DEMO, starting from the point that enterprises as a phenomena had existed in the real world but no models were available due to the lack of modeling artifacts including methods and notations, identified components of enterprises and modeled interactions, or a mechanism. Such a model can be regarded as a semantic model in the terminology of this paper.

CRISP model However, DEMO is equipped with a formalism, namely the CRISP model, which is described as *a meta model for modeling organizations* [8, p. 217]. A DEMO model, technically an elementary or composite actor role, is formally defined in [14] by a tuple $\langle \mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{S}, \mathbf{P} \rangle$, where $\mathbf{C}$ is a set of C-fact types (called the coordination base), $\mathbf{R}$ is a set of C-fact types and P-fact types (called the response base), $\mathbf{I}$ is a set of C-fact types (called intention base), $\mathbf{S}$ is a set of C-fact types and P-fact types (called the state base), $\mathbf{P}$ is a partial function (called the performance function), D is the set of time durations, $\wp X$ represents the power set of a set X, and then $\mathbf{P} \colon \wp\mathbf{C} * \wp\mathbf{I} * \wp\mathbf{S} \to \wp(\mathbf{R} * \mathbb{D})$ [4].

The model of an elementary organization according to the CRISP meta model is called a *crispie*, and then an organization is conceived as a network of collaboration of crispies, called crispienet [8, p. 133]. Therefore, aggregating crispies into one crispienet is considered as model manipulation in this research.

From the viewpoint of this study, the CRISP model may also work as the syntactical model. For clarification, it can be stated that a CRISP model as a syntactical model $\langle \mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{S}, \mathbf{P} \rangle$ becomes a semantic model $\langle \mathsf{C}, \mathsf{R}, \mathsf{I}, \mathsf{S}, \mathsf{P} \rangle$ if any interpretation is added such as traffic light or elevator. Also, how well the semantic model represents the real world is also another problem out of the scope of CRISP, which may be able to bridge a gap between semantic model and syntactical model, not between the real world and semantic model. The author believes that the CRISP model is a good candidate for the syntactical model of DEMO model. These formalizations are however too complex to define operation over them because an element of crispy, namely the performance function $\mathbf{P}$, is a complex mapping and it means that model manipulations should be operations over mappings. Moreover they are not adequately capable of symbol-manipulation in syntax.

[4] CRISP was previously defined by a tuple $\langle \mathbf{C}, \mathbf{R}, \mathbf{I}, \mathbf{S}, \mathbf{P} \rangle$, where $\mathbf{C}$ is a set of C-facta (called the coordination base), $\mathbf{R}$ is a set of action rules (called the rule base), $\mathbf{I}$ is a set of intentions (called intention base), $\mathbf{S}$ is a set of facta and stata (called the state base), $\mathbf{P}$ is a set of P-facta (called the production base), D is the set of time durations, $\wp X$ represents the power set of a set X , and then $\mathbf{R} \colon \mathbf{C} * \wp\mathbf{S} \to \wp(\mathbf{I} * \mathbf{P} * D * D)$ [8, p. 217].

Petri Net Another formalization of DEMO model is Petri Net [15]. However, it is generally limited to process-oriented models including Process Model of DEMO model. Since there exists a study that applied Petri Net to formalize non-process-oriented models such as UML Class Diagram [16] and Use Case Diagram [18], it is not impossible for Petri Net to cover all the four aspect models, but it is apparently not the best bet.

XML Schema It would be helpful to make a remark on data formats for positioning this study more clearly. As one of the model formalization initiatives, some studies propose a data format for DEMO. Although those data models authored along with in the data formats are replacing the information of informal semantic models into a digital format such as XML [19], it does not provide any insights in model manipulations and the syntactical models. At most, it provides the formal representation of the general model structure in a form of so-called meta-model as XML schema. As well as the CRISP model, those meta-models are not adequately capable of symbol-manipulation in syntax.

Studies using DEMO in Practice There exists several studies that used DEMO for solving problems in practice. Regarding model manipulation, the author spotlights a serious of research summarized in [20, 21]. This study investigates general guidelines for splitting and allying enterprises with real-life case studies using DEMO models. If positioned in the context of the present article, research of [20, 21] proposes and tests the validity of designing enterprises in the real world via DEMO models. However, the study did not enter a problem stated in this research and indeed model verification after model manipulations (e.g. checking the correctness of the model obtained through the manipulations in various perspectives including but not limited to notational syntax and semantics) were apparently performed by manual work of experts. This is that exactly what this research is addressing.

2.2 Other EE-related Studies

Business Process Mode and Notation (BPMN) is one of the most popular business process modeling notational standards and is frequently studied in many aspects. There exist studies on formal semantics of BPMN and its analysis by Petri Net [22], by process algebra Communicating Sequential Processes (CSP) [23], and by CSP+Time(CSP+T) [24]. These studies apparently propose the formal semantics for BPMN and do not reach mathematical discussion in syntax[5] and symbol-manipulation.

Regarding model manipulations, an approach proposed by [25], in which a directed graph named a business process graph is hired for formalism, covers

[5] It should be noted that the term *syntax* that appears in those literature above is used there for a different meaning i.e. standards of notations. This is not to be confused with the concept of syntax that this paper has used so fat and hereafter. Instead, this paper hires the term *notational syntax* for referring to *syntax* as standards of notations.

manipulations of match, merge, and digests (intersection) over BPMN models and any other specific notations as long as it can be represented as a business process graph. The merge of Event-Driven Process Chains (EPCs) is studied in [26], in which EPCs are substituted by function graphs, and then the function graphs are merged, and finally the result is obtained in the form of EPC restored from the merged function graph. The merge of workflows in the form of workflow net is also investigated in [27].

[25, 26, 27] and other similar studies define the domains of manipulations and provide concrete algorithms for the manipulation of merge. These studies are in line with this present research in terms of the research background and motivations. However, there exists a lack of mathematical examination and justification at all: whether the manipulations performed by the algorithms are closed, commutative, and associative or not. Moreover, even though [23] and [26] define *well-formed* BPMN and *well-formed* EPC as a subset of the general BPMN and EPC respectively, it seems that the concept of *well-formed* is used only for making the domain narrower for lifting the feasibility of the algorithms. Thus, it is not guaranteed that the results of the manipulations stay in the domain even if the manipulation is performed within the *well-formed* models. In addition, it should not be overlooked that these formalism and algorithms are usually suitable for flow-oriented modeling languages, not for non-flow models including DEMO Construction Model.

Another concept which may sound falsely similar to process fragmentation is "process decomposition", in which a process model is simplified by hiding *certain parts* within a collapsed sub-process for improving its readability. The *certain parts* can be determined typically by single-entry-single-exit regions [28]. Yet another such concept is "view-based separation of concerns", in which a process model is simplified into multiple views by hiding certain parts depending on different concerns of different stakeholders. Typical examples are control-flow view, collaboration view, information view, and so forth. Put differently, process fragmentation might be labeled as flat (or horizontal) decomposition, and process decomposition as hierarchical (or vertical) decomposition. View-based separation of concerns can be both, or hybrid of the flat and hierarchical decompositions. At this moment, the scope of this present research would accommodate the flat composition and decomposition.

2.3 Studies in Other Fields of Science and Engineering

The act of taking an extensive look at another field of science and engineering leads us to find existing research on similar issues in those fields.

Computer Science and Software Engineering The author found that computer science and following software engineering are one of the most advanced fields in the research of model manipulations. Especially, Unified Modeling Language (UML), which provides a standard way to visualize the design of a system, and Web Ontology Language (OWL), which composes ontologies, are apparently under intense investigation. It is partially because of the dramatic growth in size of models required in information systems and artificial intelligence, which are also being advanced rapidly.

UML Class Diagram is formalized in a variety of formal languages such as Z notation [29], category theory [30], algebraic set theory [31], Description Logic (DL) [17], temporal logic [32], and so forth. UML Use Case Diagram is also formalized by formal languages such as Z notation [33] and category theory [30].

The most of the studies mentioned above seem to be focusing on the establishment of formal semantics of the modeling languages and limited within the scope of semantics. Nevertheless, these early literature are very helpful in exploration of formal languages suitable for this research.

3 Proposed Solution

3.1 Preliminary Idea

The following high-school-level example may be helpful in drafting the concept of handleability in model manipulation. Let's say *one plus two equals three* ($1+2=3$). Suppose that the domain is the natural numbers $\mathbb{N}=\{1,2,3,...\}$. Then, plus $+$ is an operation on the domain $\mathbb{N}$ and the result 3 is a natural number. In general, if you apply the operation plus to two natural numbers, the result is a natural number and never become a number out of the domain: $\forall n,m \in \mathbb{N}, n+m \in \mathbb{N}$; this characteristic is called closedness. You can change the order of calculation like $1+2=2+1$ and $(1+2)+3=1+(2+3)$. These characteristics are named commutativeness and associativeness respectively. If you however say *one divided by two equals 0.5* ($1/2=0.5$), the situation has been changed. The result 0.5 is not a natural number. However, if you replace the domain with the real numbers excluding zero $\mathbb{R}\backslash\{0\}$, the result 0.5 is a real number and thus still remains within the domain. Although you solved the trouble in closedness, you still cannot change the order ($1/2 \neq 2/1$). Therefore, a clue to determine the property of operation such as closedness, commutativeness, and associativeness resides in how you define the domain ($\mathbb{N}$,$\mathbb{R}\backslash\{0\}$, etc.) and the operations (addition plus, subtraction minus, multiplication times, division divide, etc.). This observation implies that the same discussion might be the case for enterprise models.

These discussions with a set S and a collection of operations on S are formally studied as an algebraic structure in universal algebra: the field of mathematics that studies mathematical structures with a strong favor of algebra. Operations are described in the form of n-ary operation on S: taking n elements of S and returning a single element of S. One way to characterize an algebraic structure is by referring to it as an algebraic structure of a certain type Ω, where Ω is an ordered sequence of natural numbers that represent the arity of each operation of the algebraic structure. After a set and operations are prepared, the nature of the algebraic structure can be further specified by axioms, which are often given in the form of equational laws. Given the type of algebraic structure Ω, its basic constructions are defined in the following three concepts: homomorphic image, subalgebra, and product. Indeed HSP theorem by Garrett Birkhoff proved that, loosely speaking, two definitions of a class of algebraic structures are equivalent if and only if it is closed under homomorphic images, subalgebras, and arbitrary direct products.

3.2 Research Methodology

The basic approach of systems engineering is to define another type of structure Y for an object which has a structure X, and then to apply knowledge in Y to the object analogically. The definition of the new structure Y is given by composing a class of objects which have a similar structure and by defining operations on the class.

The purpose of this study is to figure out the entire structure of DEMO models by identifying any kinds of mathematical structure existing in various DEMO modes and by studying mutual relationships between those structures. Structures in mathematics are wide-ranging such as numbering structure for studying natural numbers, algebraic structure for operations, phase structure, geometry structure, complex structure, and so forth. Among these structures, algebraic structure seems to be the most promising for the focus of this research: model manipulation.

Although algebraic structure is a good choice for this research, it remains a problem how to represent the DEMO models in mathematics. As seen in Section 2, there exists a variety of options such as set theory, category theory, Description Logic, temporal logic, Z notation, CSP, etc. Since this study aims to investigate issues around model manipulation by utilizing symbol-manipulation in the syntax side, it should be determined which formalization to be hired in this study by not only the past usage in the semantics side but also their symbol-manipulation capability in the syntax side. In the light of this perspective, Z notation and CSP are in a sort of high-level languages but have a shortcoming in poor symbol-manipulation. Thus, this research should choose one of the options which are primitive in mathematics because primitive languages are very basic and studied very well. Among the final candidates of set theory and category theory, this research has chosen set-theoretic algebraic formalization as the first choice because set theory has a high affinity for model theory.

Among a variety of set-theoretic algebraic structures, this research tries to formalize the structure of DEMO models into Boolean algebraic structure. Boolean algebra is a branch of algebra in which the values of the variables are 1 (true) or 0 (false), initially introduced by George Boole [34]. Parallelly, *the Boolean algebra generated by a set A, denoted by $b(A)$, is the set of subsets of A that can be obtained by means of a finite number of the set operations: union , intersection , and complementation* [35, p. 185]. Indeed, only the complementation (^{C}) and either of the two other operations ($\cup$ and $\cap$) are the fundamentally basic operators [6]. For the sake of argument, this article uses $\cup$ and ^{C} as the basic

[6] Since the intersection $\cap$ can be obtained by means of the union $\cup$ and the complementation ^{C} such as $A \cap B = (A^C \cup B^C)^C$ (or, t $\cap$ can be obtained by $\cup$ and ^{C} such

operators. Then, combinations of only the two operators can create a variety of operators such as digest, reduce, slice, etc. Put differently, Boolean algebra lets us use the output of an operator for the input of another operator. This nature of Boolean algebra requires this study to firstly define the two basic operators, which should be corresponding to the union $\cup$ and the complement ^{C} in set theory.

Standing on the consideration above, this research is designed into the following steps.

Step 1: Formalization As reviewed in Section 2, the current state of research is missing formalization of DEMO models in set-theoretic approach. Therefore, the very first step should prepare the formalization.

Step 2: Algebra for Manipulation After setting up algebraic structure of DEMO models, the next step should construct the algebraic operations that are equivalent to model manipulation. Based on the consideration above brought by the nature of Boolean algebra, this step firstly should build the two basic operators, which should correspond to the union $\cup$ and the complement ^{C} in set theory. Analogically speaking, they would be the merge operator and the operator which takes a part of the model as the input and returns the rest of the model. After the two fundamental operators are established, this study will combine them for multiple times if necessary to construct other operations such as digest (providing the common part of more than one models; associated with the intersection $\cap$ in set theory), reduce (providing a partial model without a certain part from the original; associated with the difference $-$), and slice (providing an arbitrary part of the original models; associated with the concept of sub-algebra), and so forth. Although the author does not have any reason to insist on what is the sufficient set of operations, the point is that one can construct any new operation for his or her purpose *on demand* by combining the existing operations, which are originally made of the two basic operators.

Step 3: Analysis of Algebraic Structure The algebraic structure made in Step 2 is examined in this step to figure out any properties of the algebraic structure. Firstly, this step investigates whether those operations exhibit the closedness, associativity, and commutativity. If they do not always exhibit those characteristics, this step investigates further the requirement for defining well-formed sub-algebraic structure.

Step 4: Validation The validity of this research is evaluated by case studies. Currently, the author prepares two case studies.

The author is aware of the fact that if the formalization is made within the first-order logic, validation between semantics and syntax are already proven in two ways by model theory: soundness theorem—anything one can prove is

as $A \cup B = (A^C \cap B^C)^C$, vise versa), either of $\cup$ or $\cap$ is not of necessity. However, for the sake of simplicity, it is common to define these three operations (^{C}, $\cup$, and $\cap$) as the basic operations of Boolean algebra.

semantically valid—and completeness theorem—any argument that is semantically valid is derivable. Therefore, in this case, although whether this is the case or not is not unknown at this moment, validation is not needed. If so, the author is not sure that what should be validated in a higher dimension.

Regardless of whether the formalization is not within the first-order logic, this study evaluates the impact of the findings: how long or much effort does it take for professionals to verify models with separating of what is still in need of manual work and what is always guaranteed by the findings of this research and thus is not needed anymore.

3.3 Contribution

Firstly, the advantage of algebraic structure—one can extract objects that have a similar structure in a certain sense of mathematics and develop arguments on those objects *in a lump*—provides us, by describing similarities among different objects as axioms and deriving theorems only from the axioms, conclusions that are applicable to all objects without depending on each specific object. If putting in the practical scenario of model manipulations in enterprise engineering as mentioned in Section 1.2, the result of this research would tell in advance what can be concluded in general derived from the structure of the model, and what will be the remaining concerns that are dependent on each context and thus what should be examined by extra means, in most cases, by manual work of experts.

Therefore, the primary contribution of this research will be the reduction of manual work done by experts in model manipulations and following verification tasks. By clarifying the separation between context-dependent and context-independent concerns that are currently mixed up and fully performed by manual work, the experts would be able to work only on the part of context-dependent concerns. The reminder of the verification work, i.e. context-independent concerns, would not be needed anymore because such context-independent concerns are *pre-checked*. Moreover, it implies that a manipulation or (sub-steps of a manipulation, if not the whole) which is confined within the scope of the context-independent part can be processed by computers for automation without any human verification. Put differently, it may states what can be left to computers and what is still manually checked. Thus, the result of this research would be directed to the foundation of computer-aided design tools and programs for DEMO models.

Last but not least, although a sort of research including this one in which mathematics is applied for solving problems relatively practical from the perspectives of pure mathematics may disclose a lack of mathematical stringency, rather these research may exhibit a new problem back to pure mathematics. Indeed, a small piece of this contribution has been produced at this moment as mentioned in 3.1: what does the merge of two topologies or the boundaries of two topological structures mean?

3.4 Novelty

As compared to existing approaches to the problem, the solution is new in that this research extends the scope not only to the formal semantics but also to the syntactical models, to which little attention has been paid. The author believes that the solution is better in a sense that it can, if succeed, utilize the power of symbol-manipulation in syntactical models, which makes the argument independent from the original context.

In contrast to prior studies that developed algorithms as highlighted in Section 2.2, this research would be new in a sense that it challenges to provide mathematical examination and justification for model manipulations rather than limiting the scope research to algorithm developments.

4 Current State of the Research and Future Plan

4.1 Achievement

The author archived some initial results. Firstly, Step 1, 2, and 3 for DEMO Construction Model with limitations have been conducted and published in an international conference KEOD [36]. The limitation in that article is that it allows CM only with elementary actor roles, transaction kinds, single initiator and executor link for each transaction kind (i.e. excluding composite actor roles, aggregated transaction kinds, multiple initiator links for one transaction kinds, information links). The author is working on the removal of the limitation. The article described only one model manipulation, namely “merge” as an empirical result. The extended version including “intersection”, “difference”, and “complement” will be submitted to a journal in the very near future.

4.2 Work Plan

This research will be conducted by following the steps listed in 3.2 for CM, Process Model (PM), Fact Model (FM), and Action Model (AM) respectively. Due to the hierarchy of the four aspect models, this research will sequentially work on PM, FM, and AM in the rest of this project. The work plan is sketched in Figure 2 on page 15.

The author foresees a potential limitation of this research methodology for Action Model (AM). It is because AM is described in formatted text, so that it might be unsuitable to formalize them into set-theoretic notation. Thus, whether this study can accommodate AM in the scope or not has not been determined.

Acknowledgement. The author, a doctoral course student, is financially supported by Japan Student Services Organization (JASSO) and the program for Leading Graduate Schools “Academy for Co-creative Education of Environment and Energy Science (ACEEES)” funded by the Ministry of Education, Culture, Sports, Science & Technology in Japan at Tokyo Institute of Technology.

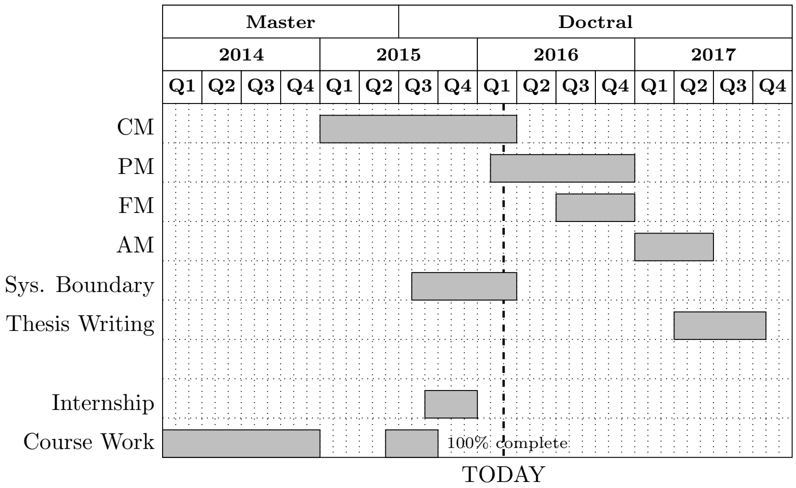

Fig. 2. Work Plan of a Four-Year Ph.D. Project Including Master

References

[1] Dumas M, La Rosa M, Mendling J, Reijers HA (2013) Fundamentals of Business Process Management. Springer Berlin Heidelberg, Berlin, Heidelberg

[2] Dietz JL, Hoogervorst JA, Albani A, Aveiro D, Babkin E, Barjis J, Caetano A, Huysmans P, Iijima J, van Kervel SJ, Mulder H, Op 't Land M, Proper HA, Sanz J, Terlouw L, Tribolet J, Verelst J, Winter R (2013) The Discipline of Enterprise Engineering. International Journal of Organisational Design and Engineering 3(1):86–114

[3] Harmon P (2016) A BPTrends Report: The State of Business Process Management 2016. Tech. rep.

[4] Mancioppi M, Danylevych O, Karastoyanova D, Leymann F (2012) Towards classification criteria for process fragmentation techniques. In: Daniel F, Barkaoui K, Dustdar S (eds) Business Process Management Workshops: BPM 2011 International Workshops, Clermont-Ferrand, France, August 29, 2011, Revised Selected Papers, Part I, Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, pp 1–12

[5] Brunet G, Chechik M, Easterbrook S, Nejati S, Niu N, Sabetzadeh M (2006) A manifesto for model merging. In: Proceedings of the 2006 international workshop on Global integrated model management - GaMMa '06, ACM New York, New York, pp 5–12

[6] Hoogervorst JAP (2009) Enterprise Governance and Enterprise Engineering. The Enterprise Engineering Series, Springer Berlin Heidelberg

[7] Wells C (2016) Mathematical structures. Revised 2015-12-16

[8] Dietz JLG (2006) Enterprise Ontology. Springer Berlin Heidelberg

[9] Bunge M (1992) System Boundary. International Journal of General Systems 20(3):215–219
[10] Marquis JP (1996) A Critical Note on Bunge's System Boundary' and a New Proposal. International Journal of General Systems 24(3):245–255
[11] Chapman WL, Bahill AT, Wymore AW (1992) Engineering Modeling and Design. Systems Engineering (Book 2), CRC Press
[12] Kühne T, S (2006) Matters of (meta-) modeling. Software and Systems Modeling 5(4):369–385
[13] Stachowiak H (1973) General Model Theory (in German: Allgemeine Modelltheorie). Springer-Verlag, Wien
[14] Dietz JLG (2005) The DELTA theory - understanding systems. Tech. rep., Czech Technical University in Prague
[15] Fatyani T, Iijima J, Park J (2014) Transformation of DEMO model into coloured petri net: Ontology based simulation. In: 6th International Conference on Knowledge Engineering and Ontology Development, KEOD 2014, SCITEPRESS (Science and Technology Publications, Lda.), Italy, pp 388–396
[16] Baresi L, Pezz M (2001) On Formalizing UML with High-Level Petri Nets. In: Agha GA, Cindio FD, Rozenberg G (eds) Concurrent Object-Oriented Programming and Petri Nets: Advances in Petri Nets, Lecture Notes in Computer Science, Springer-Verlag Berlin Heidelberg, pp 276–304
[17] Berardi D, Calvanese D, De Giacomo G (2001) Reasoning on UML Class Diagrams using Description Logic Based Systems. In: Proceedings of the KI-2001 Workshop on Applications of Description Logics (KIDLWS'01), Vienna, pp 1–12
[18] Zhao J, Duan Z (2009) Verification of Use Case with Petri Nets in Requirement Analysis. In: Gervasi O, Taniar D, Murgante B, Laganà A, Mun Y, Gavrilova ML (eds) Computational Science and Its Applications – ICCSA 2009: International Conference, Seoul, Korea, June 29-July 2, 2009, Proceedings, Part II, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 29–42
[19] Wang Y, Albani A, Barjis J (2011) Transformation of DEMO metamodel into XML schema. In: Albani A, Dietz JLG, Verelst J (eds) Advances in Enterprise Engineering V: First Enterprise Engineering Working Conference, EEWC 2011, Antwerp, Belgium, May 16-17, 2011. Proceedings, Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, pp 46–60
[20] Op 't Land M (2006) Applying Architecture and Ontology to the Splitting and Allying of Enterprises: Problem Definition and Research Approach. In: Meersman R, Tari Z, Herrero P (eds) On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops: OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeB, Lecture Notes in Computer Science, Springer Berlin Heidelberg, pp 1419–1428

[21] Op 't Land M (2008) Applying Architecture and Ontology to the Splitting and Allying of Enterprises. Doctor thesis, Delft University of Technology
[22] Dijkman RM, Dumas M, Ouyang C (2008) Semantics and analysis of business process models in BPMN. Information and Software Technology 50(12):1281–1294
[23] Wong PYH, Gibbons J (2011) Formalisations and applications of BPMN. Science of Computer Programming 76(8):633–650
[24] Capel MI, Mendoza LE (2012) Automating the transformation from BPMN models to CSP+T specifications. In: Proceedings of the 2012 IEEE 35th Software Engineering Workshop, SEW 2012, IEEE, Heraclion, Crete, pp 100–109
[25] La Rosa M, Dumas M, Uba R, Dijkman RM (2013) Business Process Model Merging: An Approach to Business Process Consolidation. ACM Transactions on Software Engineering and Methodology 22(2):1–42
[26] Gottschalk F, van der Aalst WMP, Jansen-Vullers MH (2008) Merging event-driven process chains. In: Meersman R, Tari Z (eds) On the Move to Meaningful Internet Systems: OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I, Lecture Notes in Computer Science, vol 5331 LNCS, Springer Berlin Heidelberg, pp 418–426
[27] Sun S, Kumar A, Yen J (2006) Merging workflows: A new perspective on connecting business processes. Decision Support Systems 42(2):844–858
[28] Vanhatalo J, Völzer H, Leymann F (2007) Faster and More Focused Control-Flow Analysis for Business Process Models Through SESE Decomposition. In: Krämer BJ, Lin KJ, Narasimhan P (eds) Service-Oriented Computing - ICSOC 2007: Fifth International Conference, Vienna, Austria, September 17-20, 2007. Proceedings, Lecture Notes in Computer Science, vol 1, Springer Berlin Heidelberg, pp 43–55
[29] Shroff M, France RB (1997) Towards a Formalization of UML Class Structures in Z. In: Proceedings Twenty-First Annual International Computer Software and Applications Conference (COMPSAC'97), Washington, DC, pp 646–651
[30] Meng S, Aichernig BK (2003) Towards a Coalgebraic Semantics of UML : Class Diagrams and Use Cases. Tech. rep., UNU/IIST Report
[31] Enjo H, Tanabu M, Iijima J (2009) A Step Toward Foundation of Class Diagram Algebra for Enterprise Service Systems. In: 6th International Conference on Service Systems and Service Management, IEEE, Xiamen, pp 456–460
[32] Klimek R, Szwed P (2010) Formal Analysis of Use Case Diagrams. Computer Science 11:115–131
[33] Sengupta S, Bhattacharya S (2006) Formalization of UML use case diagram - A Z notation based approach. In: International Conference on Computing & Informatics 2006, IEEE, Kuala Lumpur, pp 1–6

[34] Boole G (1847) The Mathematical Analysis of Logic: Being an Essay Towards a Calculus of Deductive Reasoning. Cambridge: Macmillan, Barclay, & Macmillan
[35] Comtet L (1974) Advanced Combinatorics: The Art of Finite and Infinite Expansions. Springer Netherlands
[36] Suga T, Iijima J (2015) Does 'Merging DEMO Models' Satisfy the Associative Law? - Validation of Partial Models and Merge Operation. In: Proceedings of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, SCITEPRESS - Science and and Technology Publications, Lisbon, vol 2, pp 467–478