A Formal Model Integration

Jiayang Wang, Mengchi Liu
School of Computer Science, Carleton University
Ottawa, Ontario, Canada K1S 5B6
wangjiayang@hotmail.com, mengchi@scs.carleton.ca

Abstract

Model integration is an important section of the model management research area, which can be divided into two parts: one is the integration based on model definition, the other is the model manipulation based integration. The primary foundation of the model integration based on model manipulation is the relationship of input and output between models. The paper puts forward a formalization representation of model, and presents some concepts, such as the combination model relation, the composite model and so on. Additionally, the existence of the model integration is also analyzed in detail and several sufficient conditions are proved.

1 Introduction

Model management research has been going on for many years[1,2,3,4]. A large number of research works mainly focus on the model representation and brings forward some classic model representations, such as structural modeling, logic modeling as well as the model representation based on graph, and so on. The research of model representation is the jumping-off point for model management research. But it is not sufficient to solve combination model computation in model integration just by means of studying model representation. For this purpose the model integration based on model manipulation has become an exciting direction.

There is no unified concept for model integration at present[5], and intuitive comprehension is to construct a composite model by the combination of many models in order to finish some tasks. The model integration can primarily be divided into two parts: one is the model integration based on model definition, the other is the model integration based on model manipulation. The model integration based on model definition means that the integration is accomplished when the model is defined, and in this case the integration has close relationship with model representation. The model integration based on model manipulation realizes the integration by transferring parameters between models. The literature[6] describes a comprehensive method based on structural modeling in model definition, and the literature[7] defines the model integration based on model manipulation, whereas the literature[8,9] realize the model integration based on object-oriented model management.
Because different model is of different representation, the model integration based on model definition is closely related with actual model representation, so it does not bear universality. The study of the model integration based on model manipulation does not yet come into being rigorous theory up to now. Model management method based on graph[10] analyzes the input and output between models, and gives the significant thought for model integration based on model manipulation, but it does not make elaborate theory analysis for whether model integration can be done and how to integrate.

The paper argues that input and output of general-purpose model are the key point for model combination. It makes three main contributions. First, it formalizes the model representation, and proposes the concepts about the combination model relation and the composite model. Second, the existence of the composite model is analyzed in details, and several sufficient conditions are proved. Under satisfying sufficient conditions the constructing method of the composite model is also be given. At the end of the paper the optimal strategy and the lowest-value strategy for the selection of the composite models are simply discussed.

2 Model Integration

Many problems encountered when building applications of database systems involve the manipulation of models. A model means a complex structure that represents a design artifact, such as a relational schema, object-oriented interface, UML model, semantic network, complex document, or software configuration. Many uses of models involve managing changes in models and transformations of data from one model into another. These uses require an explicit integration representation between models. This work endeavors to make database systems easier to use for these applications by making model and model integration as first-class citizens with special operations that simplify their use. We call this capability model integration.

A complete model management must assist in the selection, linking, and execution of models. That needs a general framework for formalization of models. By input and output standardization and model rules, it may be done to link different models together to solve a complicated problem. This makes heterogeneous model integration possible, and supports the advanced model integration.

We presented an outline of model, which has two main abstractions. One is the model, which captures the structure of engineered information artifacts, such as database schemas, interface definitions, semantic networks, complex documents, and software configurations. The other is
model integration, which captures relationships between models such as transformations and matchings.

2.1 Model

A model also can be thought as an entity, which can finish some tasks. When some data are inputted then it will output some results according to its interior function. The general-purpose model is a worthwhile and achievable goal. At an abstract level the following gives formal model definition.

Definition 1. A model \( m \) is defined as \( m=(IN, OUT) \)

1. IN denotes a set of input parameters, and represented as:
   \[
   m(IN) = \{ \text{in}_{1}, \ldots, \text{in}_{P} \}, \text{ or } m(IN) = \{ m(\text{in}_{1}), \ldots, m(\text{in}_{P}) \},
   \]
   \( P \) is the number of input parameters, \( P=|m(IN)| \).

2. OUT denotes a set of output parameters with only one output variable, and represented as:
   \[
   m(OUT) = \{ \text{out} \}, \text{ or } m(OUT) = \{ m(\text{out}) \}
   \]

For example, a model \( m=(IN, OUT) \), \( m(OUT) = \{ \text{out} \}, m(IN) = \{ \text{in}_{1}, \text{in}_{2}, \text{in}_{3} \} \). It means that the model \( m \) has one output parameter in output set and three input parameters in input set. Generally inputting data come from external source. Getting data from database can be represented as a special model, and the input set IN of all this kind of model is an empty set \( \Phi \).

2.2 Combination model relation

Definition 2. Let \( M = \{ m_{1}, \ldots, m_{n} \} \) be a set of models, for any two models \( m_{i} \in M, m_{j} \in M \), and \( i \neq j \), the \( <m_{i}(\text{out}), m_{j}(\text{in})> \) means that output parameter(\( \text{out} \)) of model \( m_{i} \) provides only one input parameter(\( \text{in} \)) for the input set of the model \( m_{j} \), the \( <m_{i}(\text{out}), m_{j}(\text{in})> \) is named as model ordered-pair; all of the possible model ordered-pairs compose a set \( R \), and \( MR=(M, R) \) is defined as Combination Model Relation about model set \( M \).

As above definition 2, for any model \( m \) in set \( M \), the number of input parameter of a model is denoted as \( |m(IN)| \), \( |m(IN)| \) input parameters must come from the output parameters of \( |m(IN)| \) different model. A model \( m \) provides only one parameter for another model, but the output parameter of a model can provide possible input for many different models.

Because of the uniqueness of the model output parameter, the ordered-pair \( <m_{i}(\text{out}), m_{j}(\text{in})> \) may be simplified as \( <m_{i}, m_{j}(\text{in})> \). When we do not care about which parameter of input set of model \( m_{j} \) corresponds to \( m_{j}(\text{in}) \), we can just simply use the expression \( <m_{i}, m_{j}> \).
Note that the combination model relation comprises all possible transforming relation of input and output parameters among model set M, and whether a ordered-pair $<m_i(out), m_j(in)>$ belongs to R depends on characteristics of the problem that we are discussing.

Definition 3. Let $M=\{m_1, \cdots, m_n\}$ be a set of models, $MR=(M, R)$ a combination model relation, and

$$\forall m_j \in M \land \forall in \in m_j(IN) \rightarrow \exists m_i [m_i \in M \land out \in m_i(OUT) \land <m_i, m_j(in)> \in R]$$

we say that MR is complete.

The completeness of MR means that for any input parameter $m_j(in)$ of model $m_j$ in model set $M$, there is at least one model $m_i \in M$, the output parameter $m_i(out)$ makes $<m_i(out), m_j(in)>$ come into existence. The completeness can be verified by judging whether every input of all models in M can be provided by the output of another model. Because of limitation of inputs of all models, therefore the judgment of completeness will terminate in finite steps.

2.3 Model integration

The model integration consists of formal structures for representing models and mappings between models. According to the relationship between models we have the following model integration definition.

Definition 4. Let $MM=(M, R)$ be a combination model relation and OUTPUT be an expected output set. The model integration is to constitute a set $MM=(MI, RI)$, which is defined as follows:

1. $MI \subseteq M$, $RI \subseteq R$

$MI$ is the final model set of model integration, and $RI$ is the model input and output relation among the set $MI$.

2. $\exists! mt \in MI \land [OUTPUT=mt(OUT)] \land [\exists m \in MI \rightarrow <mt, m> \in RI]$

The output of model $mt$ in $MI$ is the unique expected output after integration, and $mt$ is named as the terminal model.

3. $[\forall m \in MI \land m \neq mt] \rightarrow [\exists m_j \in MI \land <m_i, m_j> \in RI]$

The output of a non-terminal model in the set $MI$ must be the input of other models among $MI$.

4. $\forall m_i \in MI \rightarrow [\exists! m_i \in MI \land <m_i, m_j> \in RI]$

One of the input of any model in $MI$ is provided by only one model among $MI$.

5. Does not exist ordered-set $\{m_1, \cdots, m_L\} \subseteq MI$, satisfying:

$<m_1, m_2> \in RI$, $\cdots$, $<m_k, m_{k+1}> \in RI$, $\cdots$, $<m_{k-1}, m_k> \in RI$, $<m_k, m_1> \in RI$
This means that the ordered-set \( \{m_1, \ldots, m_L \} \) whose input and output parameters between models are transferred circularly does not exist.

\( \text{MM}=(\text{MI}, \text{RI}) \) that satisfies the conditions (1)-(4) is called Composite Model; moreover \( \text{MM}=(\text{MI}, \text{RI}) \) which satisfies the conditions (1)-(4) and (5) is called Loopless Composite Model.

Once \( \text{MR} \) and expected output set OUTPUT are given, then selecting all kinds of models to compose a composite model in the dynamic state is called model integration based on model manipulation. This means that only those models in given model set \( M \) can be selected for the model integration, and the integration result or the output of the composite model is the expected output.

Given a set of models \( M \), then \( \text{MR} \) may be counted on by means of the characteristics of \( M \). If \( M \) and expected output set OUTPUT are given, then it is possible to constitute \( \text{MI} \). As stated above, for any two models in \( \text{MI} \), according to transferring relationship of input and output parameters when integrating, input and output between two correlative models may form an ordered-pair, and all of the ordered-pairs consist of associated model set \( \text{RI} \), at this point model integration is finished. Whereas depending on \( \text{MI} \) and \( \text{RI} \), we can know all parameter transferring correlations between input and output of all models among the composite model. Hence it is possible to assemble different models to do model computation.

Under given \( \text{MR} \) and expected output set, model integration is to find \( \text{MI} \) and \( \text{RI} \). If \( \text{MM}=(\text{MI}, \text{RI}) \) that satisfies the conditions (1)-(5) in definition 4 can be found, then the composite model exists, otherwise it does not exist. Whether the composite model exists, how to construct and realize the composite model are basic problems in model integration.

### 3 Composite Model

Once given combination model relation and expected output set, the leading point of model integration is whether it may be finished. This is the existence of the composite model. The following theorems give out some necessary and sufficient conditions about the composite model.

**Theorem 1.** Let \( \text{MR}=(M, R) \) be a combination model relation and OUTPUT be an expected output set, terminal model \( m_t \in M \), and \( \text{OUTPUT}=m_t(\text{OUT}) \). The completeness of \( \text{MR} \) is the necessary and sufficient condition for model integration or constituting a composite model.

**Proof:** If the model integration can be finished, the completeness of \( \text{MR} \) follows immediately from definition 3, that is to say \( \text{MR} \) must be complete.
In other words, if MR is not complete, there must be true that at least one of the input of a model can not be provided when we are going to constitute a composite model, then it is impossible to work out a complete composite model.

Contrarily, if the MR is complete, the following proves that there must exist a composite model MM=(MI, RI). By constructing MM=(MI, RI) we can know MM satisfies the conditions (1), (2), (3), (4) of definition 4.

In the combination model relation MR we must find out a model whose output is expected output firstly. If the input set of this model is not empty, then continue to search other models in the combination model relation MR, and their outputs will provide parameters for necessary inputs. For every model which is called, if the input set is not empty, then it is necessary to call other models to provide input parameters further. This procedure will be going on until all input parameters are provided. At the end the expected output may be worked out by the composite model which is just now constituted.

Constructing steps are as follows:

Step A):
Let MI=Φ, RI=Φ
In order to be convenient for the expression, suppose the models in set MI will be ordered by joining order, and CurrentMI represents the model that is being processed at present in ordered-set MI.

Step B):
Finding out mt∈M, mt(OUT)=OUTPUT
Let MI={mt}, RI={<mt, NIL>}, CurrentMI=mt

Step C):
REPEAT
IF CurrentMI(IN)≠Φ
Because of the completeness of MR
Finding out m₁, m₂, ..., mₚ in model set M, P=|CurrentMI(IN)|
Satisfying <m₁, CurrentMI>∈R, ..., <mₚ, CurrentMI>∈R
LET RI=RI∪<m₁, CurrentMI>∪<m₂, CurrentMI>∪...∪<mₚ, CurrentMI>
FOR i=1 TO P
IF mᵢ∈MI THEN
MI=MI∪{mᵢ}
ENDIF
ENDFOR
ENDIF

CurrentMI moves to next model in ordered-set MI
UNTIL CurrentMI=NIL

Step D):

The final composite model MM=(MI, RI) is constituted up to now. ■

The completeness of MR ensures the rationality of step C). By \(|MI| \leq |M|\), and M is a finite set, then step C) will terminate in finite steps.

Obviously, for MM=(MI, RI), the condition (1) in definition 4 is correctness, that is \(MI \subseteq M\), \(RI \subseteq R\). The condition (2) among definition 4 can be verified by step B). According to constructing method, the conditions (3), (4) in definition 4 come into existence by step C).

By \(mt \in M\), OUTPUT=mt(OUT) and the completeness of MR, the composite model MM can be constructed. According to the constructing method, MM satisfies the conditions (1), (2), (3), (4) among definition 4. If the condition (5) can also be further satisfied, then it is proved for the existence of loopless composite model. Actually only loopless composite model can be realized in model computation.

Theorem 2. Let MR=(M, R) be a combination model relation and OUTPUT be an expected output set. and \(mt \in M \land OUTPUT=mt(OUT)\). Assume MR is complete, there does not exist ordered-set \(\{m_1, \ldots, m_L\} \subseteq M\), which satisfies:

\[
\langle m_k, m_{k+1} \rangle \in R, \text{ when } k=L, \text{ let } k+1 \text{ be } 1.
\]

Then the loopless composite model MM=(MI, RI) satisfying the conditions (1)-(5) among definition 4 must exist.

Proof: Because of completeness of MR, by the theorem 1 MM=(MI, RI) can be constituted such that satisfy the conditions (1)-(4) among definition 4.

By the characteristic of MR, and MM is constituted from MR, \(MI \subseteq M\), \(RI \subseteq R\), it is impossible to exist an ordered-set \(\{m_1, \ldots, m_L\} \subseteq MI\) which will satisfy:

\[
\langle m_1, m_2 \rangle \in RI, \ldots, \langle m_k, m_{k+1} \rangle \in RI, \ldots, \langle m_{L-1}, m_L \rangle \in RI, \langle m_L, m_1 \rangle \in RI
\]

Then MM=(MI, RI) satisfies the condition (5) in definition 4, so MM=(MI, RI) is a loopless composite model. ■

Theorem 2 means if the combination model relation MR=(M, R) is loopless, then the composite model which is constructed from it exists, and it must be a loopless composite model.
Theorem 3. Let \( MR=(M, R) \) be a combination model relation and \( OUTPUT \) be an expected output set, \( \exists m_0 \in M \), and \( OUTPUT = mt(OUT) \).

If \( R \) is complete, and \( \exists M_L = \{m_1, \ldots, m_L\} \subseteq M \), satisfying:

1. \( \forall m_i \exists m_j \left[ m_i \in M_L \land m_j \in M_L \land \text{in} \in m_i(IN) \land \text{out} \in m_j(OUT) \land \rightarrow m_j, m_i \in R \right] \)
2. \( \forall m_i \exists m_j \left[ m_i \in M_L \land m_j \in M_L \land \text{out} \in m_i(OUT) \land \text{in} \in m_j(IN) \land \rightarrow m_i, m_j \in R \right] \)

Then the loopless composite model which satisfies the conditions (1)-(5) among definition 4 will exist.

Proof: Supposes that the set \( M_L = \{m_1, \ldots, m_L\} \subseteq M \) exists.

and \( <m_1, m_2> \in R, \ldots, <m_k, m_{k+1}> \in R, \ldots, <m_{l-1}, m_l> \in R, <m_l, m_1> \in R \)
then, it will be:

1. \( \forall m_k \in M_L, \exists m_{k-1} \in M_L, \text{in} \in m_k(IN) \land \text{out} \in m_{k-1}(OUT), \)
   
   There will be \( <m_{k-1}, m_k> \in R \), when \( k = 1 \), let \( k-1 \) be \( L \)
2. \( \forall m_k \in M_L, \exists m_{k+1} \in M_L, \text{out} \in m_k(OUT) \land \text{in} \in m_{k+1}(IN), \)
   
   There will be \( <m_k, m_{k+1}> \in R \), when \( k = L \), let \( k+1 \) be \( 1 \)

It is inconsistency with what is given in the conditions of the theorem, then the hypothesis does not come to existence. Further more it can be proved by means of the theorem 2. 

Theorem 4. Let \( MR=(M, R) \) be a combination model relation and \( OUTPUT \) be an expected output set, \( mt \in M \land OUTPUT = mt(OUT) \). Assume \( MR \) is complete, and \( MM=(MI, RI) \) is constituted according to theorem 1.

If \( \exists! M_L = \{m_1, \ldots, m_L\} \subseteq M \),

\( <m_1, m_2> \in R, \ldots, <m_k, m_{k+1}> \in R, \ldots, <m_{l-1}, m_l> \in R, <m_l, m_1> \in R \), and \( MI \cap ML = \Phi \)

Then \( MM=(MI, RI) \) is a loopless composite model.

Proof: Because \( MM=(MI, RI) \) can be constructed from \( MR \), by the constructing method we know \( MM \) satisfies the conditions (1), (2), (3), (4) among definition 4.

If it does not satisfy the condition (5) among definition 4

then there is a set \( M_0 = \{m_1, \ldots, m_L\} \subseteq MI \), and

\( <m_1, m_2> \in RI, \ldots, <m_k, m_{k+1}> \in RI, \ldots, <m_{l-1}, m_l> \in RI, <m_l, m_1> \in RI \)

By hypothesis in the theorem \( M_L \) is the unique subset in \( M \) that possesses given properties, and that \( M_0 \subseteq MI \subseteq M, RI \subseteq R \), then it can be inferred \( M_0 = ML \), so it is inconsistency with what is given \( MI \cap ML = \Phi \) in the theorem. Therefore \( MM \) satisfies the condition (5) among definition 4. 

\( \blacksquare \)
Actually there may be a number of subsets which have the same characteristics with model ML. As long as MI does not intersect with all of these subsets, then the composite model MM=(MI, RI) which is constructed according to theorem 1 is a loopless composite model.

Theorem 4 means that if existing circular transformation among combination model relation MR, then takes out all loops among MR, and the composite model MM=(MI, RI) still exists, consequently MM must be a loopless composite model. This describes a special state.

4 Model Evaluation

According the combination model relation MR, for a given input m(in)∈m(IN) of some model m, maybe there are many model outputs m_1(out), m_2(out), ..., m_n(out), all of them satisfy <m_1, m(in)>∈R, <m_2, m(in)>∈R, ..., <m_n, m(in)>∈R. In the step C) of theorem 1, the strategy is to select any one of them which satisfies the given conditions. Therefore a number of composite models may be constructed just for one expected output set OUTPUT.

There are different application range, precision, overhead of time and space for different models, so it naturally leads to different characteristics for every composite model. In order to serve the actual need it is always necessary to select a suitable composite model. There are two selective strategies: the optimal strategy and the lowest-value strategy.

The two kinds of selective strategy all may be used to evaluate the composite model. The optimal strategy will evaluate all of the composite models that can be constituted, then the optimal evaluating value will be gotten, so the corresponding model is an optimal model. But the overhead of this method is too much to evaluate all composite models. The lowest-value strategy sets an accepted lowest value at the beginning. When evaluating some composite model, if the evaluation value satisfies the lowest value, then the evaluating procedure terminates. Otherwise the evaluating procedure will go on until the lowest value is satisfied.

5 Conclusions

This paper has investigated some issues for model integration. we presented an outline of a model for model integration. Given combination model relation MR=(M, R) and expected output set OUTPUT, Theorem 1 shows that by mt∈M, OUTPUT=mt(OUT) and the completeness of MR, the composite model MM may be constituted. From the constructing method it is clear that MM satisfies the conditions (1), (2), (3), (4) among definition 4. Theorem 2, 3, 4 indicates that in addition to the conditions which is provided among theorem 1, MM will sat-
isfy the condition (5) in definition 4 once some additional conditions are provided. In that case
MM is a loopless composite model.

Applications that manipulate models are complicated and hard to build. By implementing
generic model integration functionality presented in this paper, the database field stands a good
chance of improving programmer productivity for these applications by an order of magnitude.
It is an exciting prospect.

References

3. R. Kimball and K. Strehlo. Why decision support fails and how to fix it. ACM SIGMOD
Record 24(3), September 1995.
Twenty-First Hawaii International Conference on System Sciences III, IEEE Computer Soci-
9. Shi Ping Wang, Gee Kin Yeo, and Kim Leng PoH. An Object Oriented Modeling Inte-
gration Framework In Distribution Systems. In J. Gu (Editor), Proceedings of the Third In-
ternational Conference on Systems Science and Systems Engineering (ICSSSE'98), Beijing
China, pp 75--80, Scientific and Technical Documents Publishing House, Beijing, China,
1998.
10. Kaushal Chari, Tarun K. Sen. An implementation of a graph-based modeling system for