A Framework for Managing Optimization Models for Supply Chain Software Agents

Jae Kyu Lee
School of Information Systems,
Singapore Management University
80 Stamford Road, Singapore 178902
65-6868-0230
jklee@smu.edu.sg

Yong Sik Chang
Department of e-Business,
Hanshin University
411 Yangsan, Osan, Kyunggi 447-791 Korea
82-31-370-6421
yschang@hs.ac.kr

ABSTRACT

As third party logistic services become popular, the role of software agents increases in importance in terms of the logistics scheduling of buyers and sellers. To support many models in such a portal site focused on logistics, automatic formulation and modification of optimization models embedded in the multiple software agents is necessary. The stakeholders like manufacturers and third party deliverers have their objectives and constraints in terms of delivery requirements and resource limitations. Since a variety of situations require many combinations of models, it is not easy to prepare all the necessary models in advance. To resolve this issue, we propose the primitive model approach which identifies a base model first and then modifies it to meet the modeling requirements. A prototype architecture AGENT-OPT2 is designed with the capability of rule-based model modification. This framework is demonstrated with the cooperative delivery scheduling problems.

Categories and Subject Descriptors
H.4.2 [**Information Systems Applications**]: Types of Systems – *decision support, logistics.*

General Terms
Management, Design.

Keywords
supply chain management, optimization models, optimization agents.

1. INTRODUCTION
Many optimization models have been used to improve supply chain performance within and between enterprises [32]. Thus in an effective agent-based supply chain management (SCM) system, it is necessary to embed the optimization function in the distributed software agents. Since the combination of situations that involve many stakeholders along the supply chain is very diverse, and many corresponding models are necessary, it is too expensive to prepare all the combination of models in advance.
To overcome this problem, we need to develop a framework that can dynamically formulate and modify the optimization models specifically tailored to the supply chain domain.
In this line of research, Chang and Lee [7] contrasted three approaches of optimization model formulation and modification – the *primitive model approach, the most similar case approach,* and the *full model approach* depending upon how a ***base model*** is selected to start with the modifications.

In this study, we adopt the ***primitive model approach*** as a foundation of optimization agent design. The primitive model implies a *minimal scale base model whose modification requires only the addition of terms and constraints,* thus the modification operator for the target model is simple: ***INSERT***.

In the Web Services environment, a third party Supply Chain Optimization Service Provider can provide the dynamic formulation and solution to many users. To handle the dynamic formulation based on the users' requirements and resource constraints, we need to handle the following issues:

1) Designing the architecture of the SCM optimization modeling agent system,
2) Identifying a base model from the candidate models in SCM domain,
3) Designing SCM optimization modification operators and rules,
4) Designing the procedure of modifying the base models,
5) Designing a canonical and semantic representation for the target optimization model,
6) Transforming the canonical model to into a commercial solver.

A prototype AGENT-OPT2 is developed with this framework. To illustrate the validity of this approach, we applied the framework to the delivery scheduling problem which is a typical model for efficient logistics management in the supply chain. As an illustrational scenario, we adopted the Multi-depot Vehicle Routing Problem with Time Windows (M-VRPTW) as the base model for the delivery scheduling category. In this problem, manufacturers produce their products at factories, store them in multiple depots distributed over the regions, and deliver products to multiple buyers. Manufacturers request that the third party logistics company delivers products to buyers after picking them up from one of the depots. The modified models from the base model M-VRPTW are suitable for solving a variation of problems in this category.

When multiple base models exist, we need to select a suitable base model. For the selection of a base model, we adopt the rule-based backward chain reasoning approach. On the other hand, for the rule-based modification of the base model, we adopt the forward chain reasoning approach. For the canonical representation of the optimization model, we adopt the Document Type Definition (DTD) form that is used for the structural definition of XML statements. As a solver of integer programming (IP) models, we adopt the package LINGO.

The following sections describe the above issues one by one. Section 2 reviews the literature on the approaches to optimization model management. Section 3 describes the structure of optimization models in the supply chain. Section 4 describes the architecture of AGENT-OPT2 and the procedure of automatic formulation of optimization models. The study concludes with a summary of its contributions and limitations.

2. REVIEW OF MODEL MANAGEMENT FOR OPTIMIZATION AGENT

Model management research aims at investigating the establishment of tools and methodologies that can support effective model formulation, modification, and maintenance in the context of real world scale and multiple models in the dynamically changing business environment

Earlier research on the model management system (MMS) has focused on the representation of models and the reasoning methods in order to map specific problems with the model structure. In this line of research, a number of model management studies were conducted from the perspective of data and object management, network data modeling [16], relational data modeling [3], entity-relationship data modeling [4], and object-oriented data modeling [13,19,24,30]. Liang proposed a framework that included both relational and network concepts [26].

On the other hand, the researchers in the Artificial Intelligence community explored the knowledge representation techniques to represent the models and focused on automating the model formulation processes. These representations are based on predicate calculus [6], semantic information net [10], knowledge abstraction [8], first order logic [9,17], structured modeling [11,12,31,32], rule-base formulation [20,21], and frame based representation which is a precursor of objects in the object oriented program paradigm [2,22].
They recognized that it is very difficult to formulate a model from scratch. So they attempted to modify the existing models by analogy [3,14,15,27,28,34] and case-based reasoning approaches [7,33]. These approaches require complex knowledge for model modification. In case of modeling by analogy, it takes a huge computational effort because feature mapping is theoretically NP-complete [25]. Modeling by analogy attempted to compose a model using the model components without modifying its structure. Although the analogy approach was introduced for this purpose [1], it was not entirely satisfactory because it handles only the conceptual level.

Model management specifically for the Operations Research models are investigated by Lee and Kim [22], Yeom and Lee [35], and Chang and Lee [7]. Since optimization models are the primary solution tools for the agent's decision in supply chains, management of optimization models is the foundation of optimizing agents for the supply chain. The time-bounded negotiation framework is studied from the multiple agents context [23].

Since the previous studies do not provide a dynamic model formulation capability to cope with diverse situations in SCM, we need to develop a framework of automatically formulating optimization models specifically for supply chain domain. So this study aims at creating a methodology and tool AGENT-OPT2 that generates an automated modeling procedure within the software agents along the supply chain.

3. STRUCTURE OF OPTIMIZATION MODELS IN THE SUPPLY CHAIN

To organize the potential primitive models in the supply chain, we need to classify the optimization models for the supply chain. Typical models adopted in the supply chain domains are *transportation*, *order selection*, *production*, and *supplier selection* as depicted in Figure 1. Each model domain has several optimization model types. Figure 1 gives an example of SCM modeling hierarchy. It consists of four layers: ***SCM model domains, optimization model types, primitive models***, and ***target model components***. Each optimization model types can have a corresponding primitive model which consists of a primitive objective and primitive constraints. The upper three layers are generic, while the bottom layer of the target model components is very case specific.

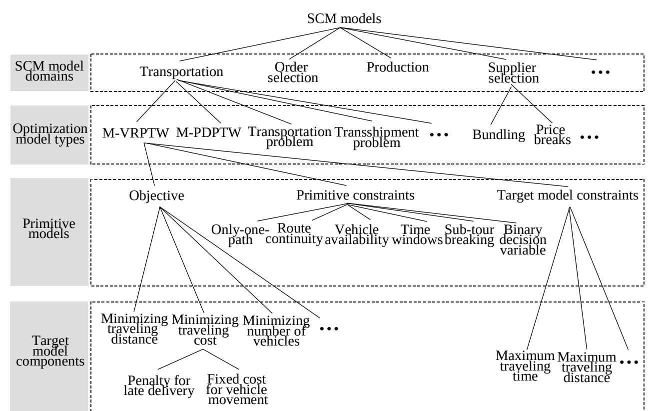

Figure 1. An illustration of SCM modeling hierarchy.

For instance, in the transportation domain, there are optimization model types like M-VRPTW, M-PDPTW (Multi-Depot Pickup and Delivery Problem with Time Windows), the transportation problem, and transshipment problem. In the supplier selection domain, there are models for bundling [31] or price break [7]. From the selected optimization model types, a primitive model is selected as a base model which it will be modified to synthesize a target model.

Let us illustrate the structure with the optimization model type M-VRPTW in the transportation domain.

3.1 Primitive Model of M-VRPTW

The primitive model of M-VRPTW is based on the vehicle routing problem [18]. The constraints of the primitive model are stated in Table 1 without objective function. The primitive constraints are the only-one-path, route continuity, vehicle capacity, vehicle availability, time window, sub-tour breaking, and binary decision variable constraints.

Table 1. Constraints for the M-VRPTW primitive model

Constraint name	Constraints in mathematical notation
Only-one-path	$\sum_{i=1}^{n+m} \sum_{k=1}^{v} X_{ijv} = 1$ for $j = 1,2, \ldots, n$ (1)
	$\sum_{j=1}^{n+m} \sum_{k=1}^{v} X_{ijv} = 1$ for $i = 1,2, \ldots, n$ (2)
Route continuity	$\sum_{i=1}^{n+m} X_{ihk} - \sum_{j=1}^{n+m} X_{hjk} = 1$ for $h = 1,2, \ldots, n+m$, $k = 1,2, \ldots, v$ (3)
Vehicle capacity	$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} q_i X_{ijk} \leq p_k$ for $k = 1,2, \ldots, v$ (4)
Vehicle availability	$\sum_{i=n+1}^{n+m} \sum_{j=1}^{n} X_{ijk} \leq 1$ for $k = 1,2, \ldots, v$ (5)
	$\sum_{j=n+1}^{n+m} \sum_{i=1}^{n} X_{ijk} \leq 1$ for $k = 1,2, \ldots, v$ (6)
Time windows	if $X_{ijk} \geq 1$ then $T_i + s_i + t_{ij} \leq T_j$ for i=1,2, …,n, j=1,2, …,n, k=1,2,…,v (7)
	$et_i \leq T_i \leq lt_i$ for i=1,2, …,n (8)
Sub-tour breaking	$Y_i - Y_j + (m + n) X_{ijk} \leq n + m - 1$ for $1 \leq i \neq j \leq n$, $1 \leq k \leq v$ (9)
Binary decision variable	X_{ijk} = binary for all i, j, k (10)

The notations used in this paper are as follows:

i, j, h: the indices i, j, and h imply the locations of buyer delivery points and depots $\{1, 2, \ldots, n+m\}$, where

n: the number n implies the number of delivery points,

m: the number m implies the number of depots.

k: the index k implies the vehicles $\{1, 2, \ldots, v\}$, where

v: the number of vehicles.

q_i: demand to delivery point i.

p_k: capacity of vehicle k.

s_i: service time at delivery point i.

t_{ij}: traveling time between delivery point i and j.

et_i: earliest delivery time at delivery point i.

lt_i: latest delivery time at delivery point i.

T_i: arrival time at delivery point i.

f_k: maximum traveling time for vehicle k.

g_k: maximum traveling distance for vehicle k.

X_{ijk}= 1 if a pair, starting node i and ending node j, is in the route of vehicle k,

X_{ijk} = 0 otherwise.

Y_i: the real number that breaks sub-tours.

The constraints (1) – (10) implies the following:

Constraints (1) and (2) ensure that each delivery point is served by one and only one vehicle.

Constraint (3) means route continuity.

Constraint (4) describes the vehicle capacity.

Constraints (5) and (6) verify vehicle availability.

Constraints (7) and (8) specify the time window constraints in which vehicles have to visit the delivery points.

Constraint (9) prohibits sub-tours**.**

Constraint (10) limits the decision variable to binary number of 0 or 1.

3.2 Options of the Objective Function

In the M-VRPTW case, the objectives of M-VRPTW are dependent on the deliverer's policy. For example, a deliver may require minimizing traveling distance, cost, or number of vehicles. The cost related objectives may be reclassified into three categories: minimizing travel cost, penalty cost for late delivery, and fixed cost for vehicle movement. The notational expression of these constraints is shown in Table 2. Multiple objectives may be combined with each other.

Table 2. Illustrative options of objectives for M-VRPTW

Type	Objective factors	Objectives
Distance	Minimizing traveling distance (*O1*)	$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \sum_{k=1}^{v} d_{ij} X_{ijk}$ (11)
Cost	Minimizing traveling cost (*O2*)	$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \sum_{k=1}^{v} a\, d_{ij} X_{ijk}$ (12)
	Penalty cost for late delivery (*O3*)	$\sum_{i=1}^{n+m} b\,(lt_i - T_i)\, Z_i$ (13)
	Fixed cost for vehicle movement (*O4*)	$\sum_{k=1}^{v} c_k W_k$ (14)
Vehicle	Minimizing number of vehicles (*O5*)	$\sum_{k=1}^{v} U_k$ (15)

Here, the three constants a, b, and c_k imply the following:

a: a proportional constant that transforms the distance unit into a cost unit.

b: a proportional constant that transforms the time unit into a cost unit.

c_k: a constant for fixed cost when a vehicle k moves.

U_k: 1 if vehicle k starts at a depot, 0 otherwise.

3.3 Objective-Driven Constraints

When the deliverer decides the objectives, some constraints should be generated to make the constraints consistent with the objective function. We illustrate two constraints of this kind as summarized in Table 3.

The objectives of minimizing traveling cost, penalty cost for late delivery, and fixed cost require searching for delivery routes minimizing the sum of their corresponding costs. For instance, the objective in (13) requires the constraints (16) and (17), and the objective (14) requires the constraints (18) and (19) to make the base model valid.

Table 3. Illustrative objective-driven constraints for M-VRPTW

Type	Objective factors	Constraints	
Cost	Penalty cost for late delivery (*O3*)	if $lt_i - T_i > 0$, then $Z_i = 1$, for $i = 1,2,\dots,n$	(16)
		Z_i = binary, for $i = 1,2,\dots,n$	(17)
	Fixed cost for vehicle movement (*O4*)	if $\sum_{i=n+1}^{n+m} \sum_{j=1}^{n} X_{ijk} = 1$ then $W_k = 1$ for $k = 1,2,\dots,v$	(18)
		W_k = binary, for $k = 1,2,\dots,v$	(19)

3.4 Optional Constraints for the Target Problem

To define a target model, additional constraints may be added to the base model. For example, the required restrictions in the M-VRPTW case may be the maximum traveling time and maximum traveling distance. Table 4 demonstrates such constraints.

The maximum traveling time in (20) means that each vehicle has to take time less than or equal to the required time limit to complete the work on a route. Similarly, the maximum traveling distance in (21) means that each vehicle has to travel less than or equal to the required distance. The notations of these constraints are shown in Table 4.

Table 4. Illustrative constraints for the target model of M-VRPTW

Type	Constraint factors	Constraints	
Time	Maximum traveling time (*C1*)	if $X_{hjk} = 1$ and $X_{ihk} = 1$ then $T_i + t_{ih} - (T_j - t_{hj}) \le f_k$ for $i=1,2,...,n$, $j=1,2,...n$, $h=n+1,n+2,...,n+m$, $k=1,2,\dots,v$	(20)
Distance	Maximum traveling distance (*C2*)	$\sum_{i=1}^{n+m} \sum_{j=1}^{n+m} d_{ij} X_{ijk} \le g_k$ for $k=1,2,\dots,v$	(21)

In this manner, knowledge of the primitive models, optional objective functions, objective-driven constraints, and optional constraints for the target problem needs to be specified to support the dynamic formulation of various target problems. So the basis of the optimization agent for any application domain is the analysis of this kind of knowledge for the particular domain. In this research, we demonstrate the analysis for the supply chain domain.

4. ARCHITECTURE OF OPTIMIZATION AGENTS

Based on the knowledge we have analyzed in section 3, we build the optimization agent AGENT-OPT2 tailored to supply chain modeling. Figure 2 illustrates the architecture of AGENT-OPT2 for the transportation domain. The optimization agent takes five steps to perform its mission.

Step 1: Identify a base model.

The optimization agent identifies a base model using the Base Model Identification Rules in the SCM model domains. Backward chain reasoning can be applied considering the delivery requirements received from the requesting manufacturers.

Step 2: Identify the requirements of the target model.

The optimization agent receives the message on resource constraints from the deliverers, and identifies a target model. The target is specified by the base model and objectives and constraints. The modification can be automatically executed by forward chain reasoning using the Target Model Identification Rules.

Step 3: Generate a canonical target model.

The target model formulated is generated in a canonical document. The canonical representation adopted in this study is the DTD form.

Step 4: Transform to a commercial solver's formulation.

The formulated model is transformed to the format of a commercial IP Solver. In this study, LINGO is adopted for this purpose.

Step 5: Solve and report.

The model is solved by the solver and the answer is reported to relevant stakeholders such as manufacturers and deliverers.

The steps are explained in the next sub-sections.

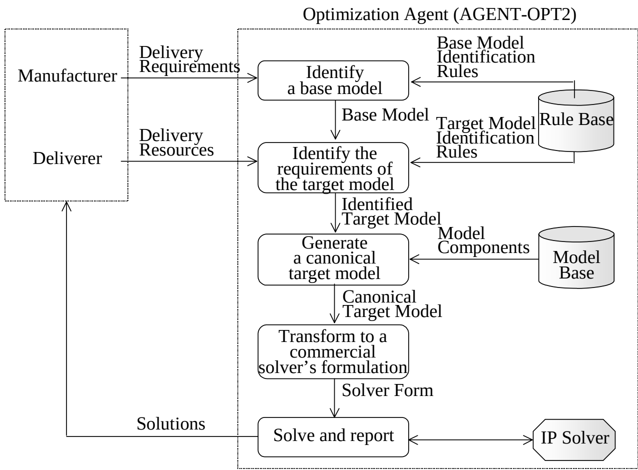

Figure 2. Architecture of AGNET-OPT2 for transportation problems.

4.1 Identification of a Base Model

In Step 1, the optimization agent recognizes the SCM model domain and identifies a base model after having received the delivery requirements from multiple manufacturers. Figure 3 illustrates an AND/OR graph for base model identification in the transportation domain. M-VRPTW will be selected when the *SCM model domain is transportation*, *items of each depot are delivered to multiple delivery locations*, and *a route is composed of a depot and several delivery locations*.

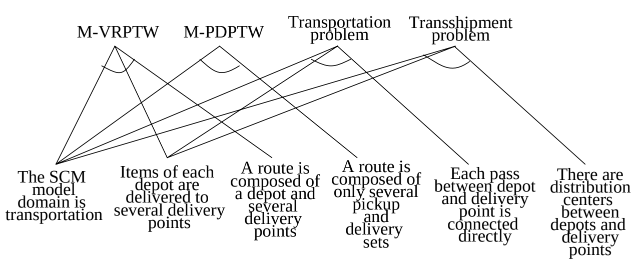

Figure 3. AND/OR Graph for base model identification.

Base models need data for their coefficients. Some of them are given from manufacturer delivery requirements and deliverer resources. The others are retrieved by the optimization agent from the common database. Table 5 shows three data groups.

In the M-VRPTW case, depots and delivery points, demands of delivery points, and time windows of delivery points are given by the manufacturer. Capacities of vehicles are given by the deliverer. The optimization agent may retrieve the service time at a delivery location and the traveling time between points based on historical data.

Table 5. Modeling data for the M-VRPTW base model

Data source	Coefficient data
Delivery requirements from manufacturer	Distance between depots and delivery points (i, j) Demand to the delivery points (q_i) Time windows at the delivery points (et_i, lt_i)
Delivery resources of deliverer	Capacity of vehicles (p_k)
Basic data in the optimization agent	Service time at delivery points (s_i) Traveling time between delivery points (t_{ij})

4.2 Identification of a Target Model

In Step 2, AGENT-OPT2 with a selected base model makes a request for information about the delivery resources. In this example, the M-VRPTW is selected as the base model in Step 1. To identify the target model, AGNET-OPT2 needs to specify the base model, additional objective terms, objective-driven constraints, and target model constraints.

The identification can be conducted by the rules described in the Target Model Identification Rules in Tables 6 to 8. Since we have adopted the primitive model approach, the necessary operators in model modification is simply the ***INSERT*** statement to insert terms to objectives and constraints, and to insert new constraints as shown in Figure 4. If we have adopted other approaches such as Most Similar Case Approach, we would need the *DELETE* operator as well.

INSERT <Expression> INTO OBJECTIVE

INSERT <Expression > INTO CONSTRAINT

Figure 4. Operators for optimization model modification in the primitive model approach.

The five rules that identify the additional objective terms are added as illustrated in Table 6. The additional term is selected by the definition of the identified objectives, and more than one rule may be fired.

Table 6. Objective identification rules for M-VRPTW

Rule name	Rule statements
RULE_M_VRPTW_ OBJ_TRAVELING_ DISTANCE	**IF OBJECTIVE IS** *minimizing_traveling_distance* **THEN** **INSERT** *Term (11)* **INTO OBJECTIVE**
RULE_M_VRPTW_ OBJ_TRAVELING_ COST	**IF OBJECTIVE IS** *minimizing_traveling_cost* **THEN** **INSERT** *Term (12)* **INTO OBJECTIVE**

The rule as illustrated in Table 7 derives constraints from the identified objectives. These constraints are necessary to make the added objective semantically meaningful. The rule identified by the target of traveling time can derive the constraint as illustrated in Table 8.

Table 7. Objective-driven constraints for M-VRPTW

Rule name	Rule statements
RULE_M_VRPTW_ CON_PENALTY_ COST	**IF OBJECTIVE IS** *minimizing_traveling_cost* **AND** *penalty_cost_for_late_delivery* **THEN** **INSERT** *Eq(16)* **INTO CONSTRAINT** **INSERT** *Eq(17)* **INTO CONSTRAINT**

Table 8. Target-driven constraints for M-VRPTW

Rule name	Rule statements
RULE_M_VRPTW_ CON_TOTAL_ TRAVELING_TIME	**IF CONSTRAINT IS** *maximum_total_traveling_time* **THEN** **INSERT** *Eq(20)* **INTO CONSTRAINT**

As such, the delivery requirements, delivery resources and policy identify the target model derived by these rules. The derived model is canonically represented in DTD form.

4.3 Canonical Representation of the Target Model

The identified target model is represented in a canonical DTD form to equip the semantics and ability of transforming to any commercial solver. In this study, we have adopted LINGO as the IP solver. To define the complete model, we need to identify the base model, added objective terms, and target constraints along with the relevant data set. This structure is depicted in Figure 5. In this model, the necessary data set includes the delivery points, vehicles, and constants. Figure 6 expresses the semantic structure in DTD form.

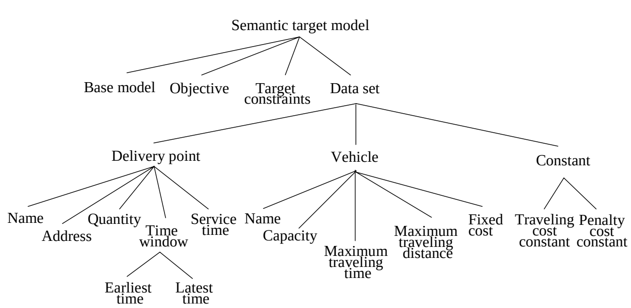

Figure 5. Semantic structure of the target optimization model in M-VRPTW.

```
<?xml version="1.0" encoding="euc-kr" ?>
<!ELEMENT canonical_target_model (base_model, objective,
          target_constraints, data_set)>
<!ELEMENT base_model (#PCDATA)>
<!ELEMENT objective (#PCDATA)>
<!ELEMENT target_constraints (constraint+)>
<!ELEMENT constraint (#PCDATA)>
<!ELEMENT data_set (delivery_points, vehicles, constants?)>
<!ELEMENT delivery_points (delivery_point+)>
<!ELEMENT delivery_point (l_name, l_addr, quantity?,
          unit_weight?, time_window?, service_time?)>
<!ATTLIST delivery_point type CDATA #REQUIRED
          seq CDATA #REQUIRED>
<!ELEMENT l_name (#PCDATA)>
…..
```

Figure 6. DTD form of the canonical optimization model for M-VRPTW.

4.4 Formulation of the Target Model for the IP Solver

The modeling factors described in Tables 1 to 4 corresponds to the terms in the objective function and equations in the constraints. This representation is effective if the optimization models can be managed as a unit of objective terms and constraint equations. Fortunately, a popular IP solver LINGO effectively handles the optimization models in this manner, and data can be managed separately. So we adopt LINGO as executing solver.

Table 9 illustrates the objective function of (11) and constraints of (21) in the form of LINGO [29]. The formulation in DTD form can be transformed to the LINGO form, and solved by the solver.

Table 9. Illustrative model components in the form of LINGO

Model component	LINGO formats
Minimizing traveling distance in (11)	*MIN = @SUM(LINKS(i,j,k): D(i,j) * X(i,j,k))*
Maximum traveling distance in (21)	*@FOR(VEHICLE(k): @SUM(LINK2(i, j): D(i,j) * X(i,j,k)) <= G(k))*

5. Conclusion: Toward Ontology for Supply Chain Model Warehouse Services

We have seen how the primitive model approach along with modified objectives and constraints can easily formulate a set of optimization models for SCM problems. This study proposed a rule-based model modification scheme based on backward chain and forward chain reasoning that can formulate optimization models automatically. We showed the viability of this approach through the prototype AGENT-OPT2, and illustrated the framework using delivery scheduling problems.

We have confirmed that the proposed framework can be used for the management of multiple models in this manner because the modeling components can be shared by multiple instance models. In this sense, this architecture can be adopted for use in public model warehouses [5] which are equipped with external solvers. In the Web Services environment, optimization model management service can be provided to multiple and diverse customers, such as the third party deliverers and virtual manufacturing schedulers. This implies that the framework can be effectively scaled up to diverse supply chain modeling services.

The application of AGENT-OPT2 developed in this spirit is not limited to SCM alone, but the rules of adding objectives and constraints need an intensive analysis in terms of the application domain. In this sense, the supply chain is the most typical application with such analyzable modeling knowledge. Let us call such knowledge ***Modeling Ontology***. In order to more comprehensively support SCM automatic modeling, the modeling ontology study needs detailed scrutiny. This implies huge research potential for supply chain modeling experts and information system designers.

If a customer repetitively uses a formulated model with minor modifications, the formulated model can be regarded as the base model of next step. Then the adoption of the *primitive model approach* can evolve into the *most similar case approach* reducing the effort of modification.

6. ACKNOWLEDGMENTS

A full version of this study is also printed in the book, *Multiagent-based Supply Chain Management* (edited by B. Chaib-Draa and J. P. Muller, Springer 2006), and reproduction is permitted by the editor.

7. REFERENCES

[1] Binbasioglu, M. Process-based Reconstructive Approach to Model Building, *Decision Support Systems*, 12, 2 (1994), 97-113.

[2] Binbasioglu, M., and Jarke, M. Domain Specific DSS Tools for Knowledge-based Model Building, *Decision Support Systems*, 2, 3 (1986), 213-223.

[3] Blanning, R.W. A Relational Framework for Model Management in Decision Support Systems, *DSS-82 Transactions*, (1982), 16-28.

[4] Blanning, R.W. An Entity-relationship Approach to Model Management, *Decision Support Systems*, 2, 1 (1986), 65-72.

[5] Bolloju, N., Khalifa, M., and Turban, E. Integrating Knowledge Management into Enterprise Environments for the Next Generation Decision Support, *Decision Support Systems*, 33, 2 (2002), 163-176.

[6] Bonczek, R., Holsapple, C., and Whinston, A. *Foundations of Decision Support Systems*. Academic Press, New York, 1981.

[7] Chang, Y.S., and Lee, J.K. Case-based Modification for Optimization Agents: AGENT-OPT, *Decision Support Systems*, 36 (2004), 355-370.

[8] Dolk, D.R., and Konsynski, B.R. Knowledge Representation for Model Management Systems, *IEEE Transactions on Software Engineering SE-10*, 6 (1984), 619-628.

[9] Dutta, A., and Basu, A. An Artificial Intelligence Approach to Model Management in Decision Support Systems, *IEEE Computer*, 17, 9 (1984), 89-97.

[10] Elam, J.J., Henderson, J.C., and Miller, L.W. Model Management Systems: an Approach to Decision Support in Complex Organizations. In *Proceedings of the first international conference on Information system* (1980), 98-110.

[11] Geoffrion, A.M. Introduction to Structured Modeling, *Management Science*, 33, 5 (1987), 547-588.

[12] Geoffrion, A.M. The Formal Aspects of Structured Modeling, *Operations Research*, 37, 1 (1989), 30-51.

[13] Huh, S.Y. Model-base Construction with Object-oriented Constructs, *Decision Sciences*, 24, 2 (1993), 409-434.

[14] Ishikawa, T., and Terano, T. Analogy by Abstraction: Case Retrieval and Adaptation for Inventive Design Expert Systems, *Expert Systems with Applications*, 10, 3/4 (1996), 351-356.

[15] Kedar-Cabelli, S.T. *Toward a Computational Model of Purpose-directed Analogy*, Kaufmann, Analogica, CA, 1988.

[16] Konsynski, B., and Dolk, D.R. Knowledge Abstractions in Model Management, *DSS-82 Transactions*, (1982), 187-202.

[17] Krishnan, R. A Logic Modeling Language for Automated Model Construction, *Decision Support Systems*, 6, 3 (1990), 123-152.

[18] Kulkarni, R.V., and Bhave, P.R. Integer Programming Formulations of Vehicle Routing Problems, *European Journal of Operational Research*, 20 (1985), 58-67.

[19] Le Claire, B., and Sharda, R. An Object-oriented Architecture for Decision Support Systems. In *Proceedings of ISDSS conference* (1990), 567-586.

[20] Lee, J.S. A Model base for Identifying Mathematical Programming Structures, *Decision Support Systems*, 7, 2 (1991), 99-105.

[21] Lee, J.S., Jones, C.V., and Guignard, M. MAPNOS: Mathematical Programming Formulation Normalization System, *Expert Systems with Applications*, 1 (1990), 367-381.

[22] Lee, J.K., and Kim, M.Y. Knowledge-assisted Optimization Model Formulation: UNIK-OPT, *Decision Support Systems*, 13, 2 (1995), 111-132.

[23] Lee, K.J., Chang, Y.S., and Lee, J.K. Time-bound Negotiation Framework for Electronic Commerce Agents, *Decision Support Systems*, 28, 4 (2000), 319-331.

[24] Lenard, M.L. An Object-oriented Approach to Model Management, *Decision Support Systems*, 9, 1 (1993), 67-73.

[25] Liang, T.P. Analogical Reasoning and Case-based Learning in Model Management Systems, *Decision Support Systems*, 10, 2 (1993), 137-160.

[26] Liang, T.P. Integrating Model Management with Data Management in Decision Support Systems, *Decision Support Systems*, 1, 1 (1985), 221-232.

[27] Liang, T.P. Modeling by Analogy: A Case-based Approach to Automated Linear Program Formulation, *IEEE*, (1991), 276-283.

[28] Liang, T.P., and Konsynski, B.R. Modeling by Analogy: Use of Analogical Reasoning in Model Management Systems, *Decision Support Systems*, 9, 1 (1993) 113-125.

[29] Lingo, User's Guide.Lindo Systems Inc., Chicago IL, 2003.

[30] Muhanna, W.A. An Object-oriented Framework for Model Management and DSS Environment. In *Proceedings of the ISDSS conference* (1990), 553-565.

[31] Park, S.J., and Kim, H.D. Constraint-based Metaview Approach for Modeling Environment Generation, *Decision Support Systems*, 9, 4 (1993), 325-348.

[32] Tsai, Y. Model Integration using SML, *Decision Support Systems*, 22, 4 (1998), 355-377.

[33] Vellore, R.C., Sen, A., and Vinze, A.S. A Case–based Planning Approach to Model Formulation. In *Proceedings of the ISDSS conference* (1990), 353-382.

[34] Winston, W.L. *Operations Research, 3rd ed.*, Duxbury Press, Belmont, CA, 1994.

[35] Yeom, K., and Lee, J.K. Logical Representation of Integer Programming Models, *Decision Support Systems*, 18, 3-4 (1996), 227-251.