An Intelligent Solution Support System For Spatial Modelling and Decision Support

Anthony Gar-On Yeh and Jiming Qiao
Centre of Urban Planning and Environmental Management
The University of Hong Kong, Pokfulam Road, Hong Kong
Email: hdxugoy@hkucc.hku.hk

Abstract

'Modelling inside GIS' has been widely researched over the years. This paper argues that such approach lacks appropriate model reusing and management functions because it is often domain-dependent. In this paper, we propose a 'modelling outside GIS' approach to design a model management module that can be incorporated with GIS and other systems to perform flexible model retrieval and development. As a first step, an intelligent solution support system (subsystem) that can assist users to select and construct appropriate models for their particular application domains is developed. Some essential mechanisms and procedures involved in the developing this system are proposed and demonstrated.

1. Demands in spatial decision making

Geographical information systems (GIS) has emerged as an important information processing and spatial technology. It has helped to regenerate the interests and applications of complex spatial and dynamic models in problem solving and decision making (Bertuglia et al, 1994) . There are two main approaches in current spatial modelling. One commonly adopted approach is 'modelling inside GIS' which attempts to incorporate different models into GIS with different coupling and integrating strategies (Batty and Xie, 1994). Another approach is the adoption of decision support systems (DSS) originally developed for operation research and management science (OR/MS) to spatial decision making domains. This is commonly described as spatial decision support systems (SDSS) which is particular useful for ill-structured and unstructured spatial problems (Densham and Amstrong, 1994). Recently some intelligent functions are built into such system to provide knowledge based support in information processing and decision making (Leung, 1997). However, since these approaches are generally developed within GIS or AI environments, and applications are often domain-dependent, they do not provide flexible mechanisms and capabilities for model sharing, reuse and communications among different applications and systems. Thus considerable modelling programming and model development process are often involved when the problem domains and application systems are changed.

Prototypes of object-oriented geographical modelling systems (GMS) have recently been proposed with some functions of modelbase management (Bennett, 1997; Raper and Livingstone, 1995). Their approaches support the development and use of dynamic geographical models, and can help users to construct domain specific geo-oriented simulation models by using icon-based user interface and autonomous modelling agents (Bennett, 1997). However, these systems do not provide appropriate mechanisms for reusing the established models, and do not offer intelligent assistance to users to retrieve and select appropriate models from a modelbase according to their needs. Another drawback is that they do not utilize existing systems such as GIS which has already developed some advanced features in spatial modelling.

Model management systems (MMS) have been proposed in OR/MS research to resolve the above problems. Some interesting model management frameworks have been designed such as database approach, graph-based approach, object-oriented approach and knowledge-based approach (see review by Bharadwaj et al, 1992; Blanning 1993). These approaches provide a challenge to the current development of SDSS although spatial modelling problems are generally much more complex than models in operation research and

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

management science. In this paper, we propose to develop an integrated solution support system (SSS) based on the extension of existing model management framework that can be applied to assist the end-users to select suitable modelling methods, and to construct more complex models for complex spatial problems.

2. Architecture of a solution support system for spatial decision making

As illustrated by Densham and Armstrong (1994), a powerful SDSS should facilitate a number of functions such as data analysis, model-based analysis and powerful visual presentation. In recent years, a variety of spatial information systems has been developed. These provide numerous resources upon which an integrated system infrastructure can be developed. With the support of object-oriented programming and system integration techniques, we proposed an integrated SSS environment in Figure 1 which incorporates essential functions of GIS, database systems, KBS and model management techniques to support overall modelling and decision making processes. The main components of the system are briefly described below.

2.1. GIS component

GIS is functioned as a display and communication system (subsystem) to store and manipulate locational, topological, thematic and non-spatial data to support cartographic display, spatial query and analytical modeling, and to present analysis results in a map form. Because the proposed system is based on MS Windows environment, we select ESRI's ArcView GIS and MapObjects as a tool for spatial data management and display. ArcView is one of the most commonly used desktop GIS software. MapObjects is built upon MS's Object Linking and Embedding (OLE) standards which consists of an ActiveX control (OCX) and programmable ActiveX automation objects. It provides a collection of

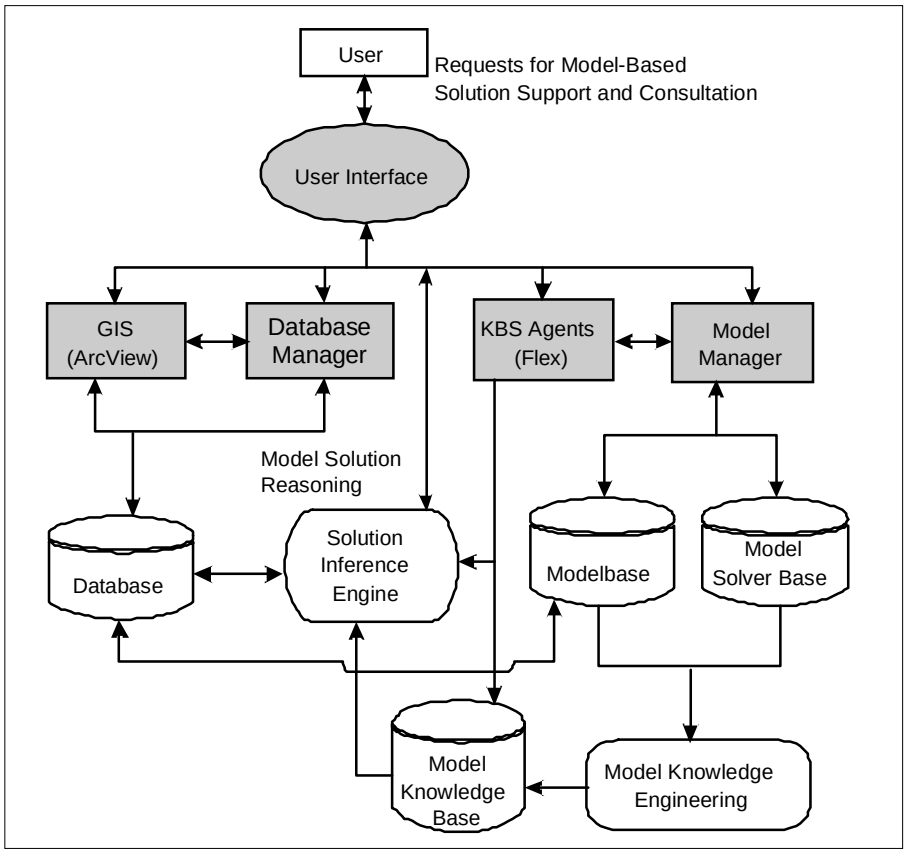

Figure 1. The architecture of an integrated solution support system

powerful mapping and GIS components for application developers. Based on the programming codes provided by MapObjects, we built a MapObjects-based map display tool which can be fully integrated with our overall system structure to enhance the functions of spatial data dislpay.

2.2. Database manager

The database manager we proposed for the intelligent SSS environment is based on MS Jet database engine, data access objects (DAO) programming interface and Open Database Connectivity (ODBC) techniques since our overall environment is built on desktop Microsoft Windows. The database manager provides a user-friendly and commonly adopted environment for database development and handling the modelled results as the defined database formats.

2.3. KBS agent

The concept of intelligent agents (multiple agents) has been used to develop a type of the distributed artificial intelligent systems that can handle various aspects of information management. Wooldridge and Jennings (1995) proposed that an intelligent agents is a system that has the following features: (1) *autonomy*: agents operate without the direct intervention of humans and others, and have some kind of control over their actions and internal states; (2) *social ability*: agents interact with other agents (and possibly humans) via a some kind of agent communication language; (3) *reactivity*: agent perceive their environment and respond in a timely fashion to changes that occur in it; (4) *pro-activeness*: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by taking the initiative. Thus a primary objective of a KBS agent is to help end uses to react to changes in the real world. A simple example is meeting scheduling. When an agent is notified in the event that a meeting is needed, it will get the information related to the event, and search for the available time on the user's schedule. The agent will then try to contact the agents of all the involved parties to decide on a mutually acceptable time for the meeting.

The KBS agents in our system do not fully fit with the above schema. We simply consider that a model is a type of agents when it is facilitated with knowledge. Thus it has certain intelligence in its actions and reactions to its changing modelling environment. It can also be viewed as a knowledge-based object in which object-oriented methods and programming can be applied.

Flex KBS shell, developed by Logic Programming Association (LPA), provides a comprehensive and versatile set of facilities for programmers to construct sophisticated expert systems (LPA, 1996). It supports frame-based reasoning, rule-based programming and data-driven procedures fully integrated with a logic programming environment. To make these construction accessible, the system shell also provides an Intelligent Server module which allows the user to communicate with Visual C++ codes and develop more powerful object-oriented knowledge representation schemes. The component of KBS agents is utilized to represent and handle model knowledge according to which the structure of model knowledge base and solution inference engine are developed.

2.4. Model manager

Model-based analysis is a very important approach in spatial decision making. In recent years, a number of spatial decision related model resources have been developed for different problem domains. The model manager component proposed in the system aims to handle some commonly accepted spatial models in building modelbase (or model library) which is a collection of a set of model components according to the defined model decomposition and processing procedures. Model solver is built with Visual C++ programming and directly linked with individual model components. It can be accessed according to the method calling sequences defined by the user and model solution inference engine developed by the intelligent agents component. The model execution sequence is particularly important for the implementation of integrated models which are constructed based on the existing and newly created model components. In order to achieve these objectives, model generalization and presentation schemes are developed and will be discussed in the following section.

3. Development of model manager

One of the most important techniques we proposed in the solution support system is to handle models and associated model knowledge to develop a functional model manager component. This requires the processing of existing models and the establishment of appropriate model decomposition and generalization mechanisms. Early research in knowledge-based model management framework in OR/MS primarily focused on knowledge base organization, model representation and model

retrieval. Model knowledge base structure especially for spatial or geographical models and communication with GIS has not been developed. Because of the complexity of developing a model management system, we only focus on some important aspects of the system development. In order to fully understand the concepts and framework, some explanatory examples are provided.

3.1. Generalization and decomposition of models

Model generalization aims to define and organize models into a hierarchical structure according to commonality or semantic relatedness. This provides a structural and organizational mechanism for creation of a set of model knowledge from which models and associated attributes and methods can be retrieved. Based on the frameworks of object-oriented Modelling Support Systems (Lazimy, 1993) and Structured Modelling (Geoffrion, 1987) in OR/MS, we extended their model management functions to develop an intelligent model management framework that can be utilized to handle spatial models and model components and support intelligent model solution reasoning.

Structured Modelling developed by Geoffrion (1987) provides one of the first well developed framework of model management systems. It is facilitated with an acyclic graph form of model structuring scheme (Modular Structure, Generic Structure, and Elemental Structure from the higher level to the lower level). In elemental level, five types of model elements are identified: Primitive Entity, Compound Entity, Attribute, Function and Test. Because of its structural form, the ‘class’ type of property inheritance have many similarities with object-oriented paradigm. This framework has also been extended to develop some prototype systems of the object-oriented structured modelling (Gagliardi and Spera, 1997; Huh, 1993). Lazimy (1993) also developed an integrated modelling system with considerable conceptual definitions of model components and linkages based on object-oriented philosophy. Because of the object-oriented features of their systems, we combine their structural frameworks and formulate an extended and general model generalization scheme which is shown in Figure 2. Four levels of model hierarchy are identified from the top to the root: Model Type, Model Instance, Model Component and Model Element.

Model Type defines a general pattern of model taxonomy and modelling capabilities. It can be divided into two abstract forms: general model and domain model. General model includes optimization model, simulation model, forecasting, heuristics analysis and other models which define general structural form of models and the nature of modelling activities. Domain model is specific to the application or problem, such as population forecast models. urban and regional planning models.

Model Instance corresponds to individual models. As we explained above, when model knowledge is developed and embedded into the model, it can be considered as a model agent from which retrieval and inference processes can be conducted at instance level. The example of spatial interaction model in Figure 2 includes four model cases based on different modelling conditions and constraints (Wilson, 1974). These individual cases can be viewed as model instances used to measure different interaction processes between different factors among regions (zones).

Model Component aims to decompose individual model instance into parts which are small model units with certain basic functions and procedures. Thus each component comprises some model variables and their associated actions. Distance factor $f(c_{ij})$, the attractiveness of area i and area j $W_i^{(1)}$, $W_j^{(2)}$ in spatial interaction model can be considered as model components which are associated with some 'basic unit of functions'. They can be calculated by other model instances such as analytical hierarchy process (AHP) method and factor coefficient method based on different modelling environment (Bowen, 1993; Wilson, 1974). The modelbase and solver base are constructed according to the organization of model components which can be considered as the ‘building blocks’ of the model library. Model integration and selection mechanisms are also processed at this level.

Model Elements include all the details of a specific model instance. They are basic objects in model construction, and can be divided into three subclasses in object-oriented analysis: Entities (Primary and Compound Entity), Attributes (Variable and Constant), and Functions. The Test function defined in the Structured Modelling can be considered as a subclass of Functions indicating that it inherits the general properties of the Function in addition to its special ones. Primary Entity subclass represents things or concepts postulated as primitives of the model which have no calling sequence in implementation. Compound Entity generally represents things or concepts that are defined in terms of other things or concepts (Geoffrion, 1987). It is associated with calling sequence, that is, a list of elements upon which it is dependent. In spatial interaction models, six model elements can be identified: K, O_i, D_j, A_i, B_j, T_{ij}. Attribute elements have values and generally represent properties of things and concepts. It includes three subclasses: variable attribute, constant and derived attribute. The derived

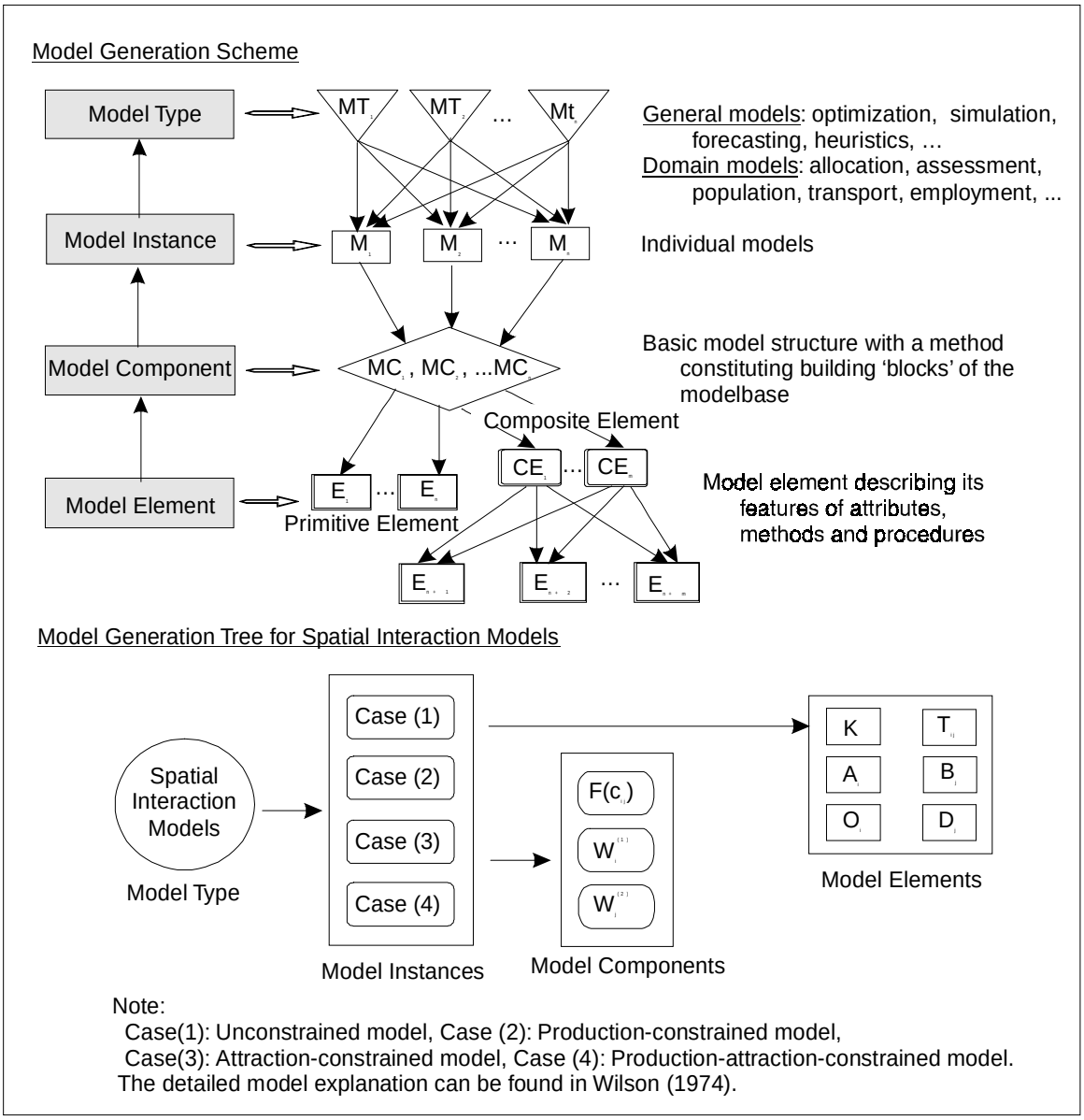

Figure 2. Model generalization scheme and applications

attribute is extracted from other functions or models, such as O_i and D_j (total interaction flow out of zone i and total interaction flows to destination zone j) can be calculated from the summation of individual factor flows and individual zones.

Class-subclass structures can also be adopted into our framework. A class may generate some subclasses which inherit attribute and functional abstractions from its parent classes, superclasses or related classes (multiple inheritance). A model class can be defined as a group of model entities and association sets. Lazimy (1993) pointed out that class-subclass specialization structure and inheritance relationships are particularly important for model building and modelling support which enable some models to share some common model components and model elements in the modelbase and inherit their attributes and procedures. This feature is very useful for model sharing and reusability of model components and

elements for different application domains (for example, factor scores computed can be shared and reused by a variety of other analysis procedures and methods). It also provides mechanism for the sharing of code components (for example, the codes for implementing certain model components or elements in solver base can be shared by other models with the same components). This mechanism can be utilized to integrate existing model components to construct more flexible new models. This process is defined as incremental spatial modelling capabilities of the system.

In object-oriented analysis, there are two types of objects: entity object and association object. An entity object can be distinctly identified. A set of entities with similar nature and functions can be grouped into entity class. For example, the total volume of interaction flows leaving individual zones in spatial interaction model can be grouped into an outflow entity class. An association object relates two or more independent entities, and represent links and interactions that exist between them. In retailing spatial interaction models, three object classes can be identified: Outflow Zone i, Inflow Zone j and Retailer, of which the Retailer is an association class representing links and interactions between instances of the entity class Outflow Zone i and instances of the entity class Iutflow Zone j. Some entity instances with value attributes can be derived from other models, such as Attractiveness variable that can be calculated from other models.

3.2. Representation of model knowledge

Because of the complexity of models and model knowledge, we build a hybrid knowledge representation scheme for intelligent agents component to handle and retrieve model knowledge. As we described above, the intelligent agents is developed based on Flex KBS shell. It provides support for frame-based and rule-based knowledge representation and it also provides linkages to Visual C++ through the Intelligence Server toolkit. Since of the object-oriented features of model generalization, we can extend Flex based knowledge representation scheme with object-oriented methods to develop a hybrid knowledge representation scheme. A model agent can be composed of many interrelated knowledge frames and each frame explains and indicates certain properties and methods (actions and interactions) for that agent. Slots in a frame are used to store an object nature, relations and attributes, whereas object methods and actions are structured in object-oriented form involving functions, implementation procedures and associated model calling sequences. Rules are used to store domain declarative knowledge, modelling procedures (methods of formulating, executing and interpreting models) and model calling sequences connected with modelbase and solver base.

A knowledge frame thus can be considered as a generalized framework describing all properties of a model and interactions among models. It also involves property inheritance from which each frame can inherit the characteristics of all related frames at higher levels. In term of individual frame structure, two types of knowledge frame can be defined: primitive frame and composite frame. A primitive frame refers to a frame that cannot be decomposed into simpler ones which can be considered as the lowest level of model frame. While composite frame is a frame that includes a set of sub-frames or primitive frames linked in a certain manner. The knowledge in a composite frame should include knowledge about all its associated primitive frames and their action sequence. Primitive and composite frames are to some extent in parallel with primitive and composite model elements in which models are generally formulated by many model elements. Hence a model knowledge base is composed of a bundle of knowledge frames formulating appropriate knowledge ‘blocks’ to describe model properties and methods. An agent based knowledge structure for spatial interaction model is shown in Figure 3.

3.3. Procedures of model solution reasoning

Solution support system is explicitly designed to support model-based solution formulation. It can also be considered as a modelling utilization system that applies all features of model and knowledge management to provide an intelligent advice to users for selecting and executing models. It tells users how to intelligently and efficiently apply the system and its internal resources to construct complex models for domain specific problems.

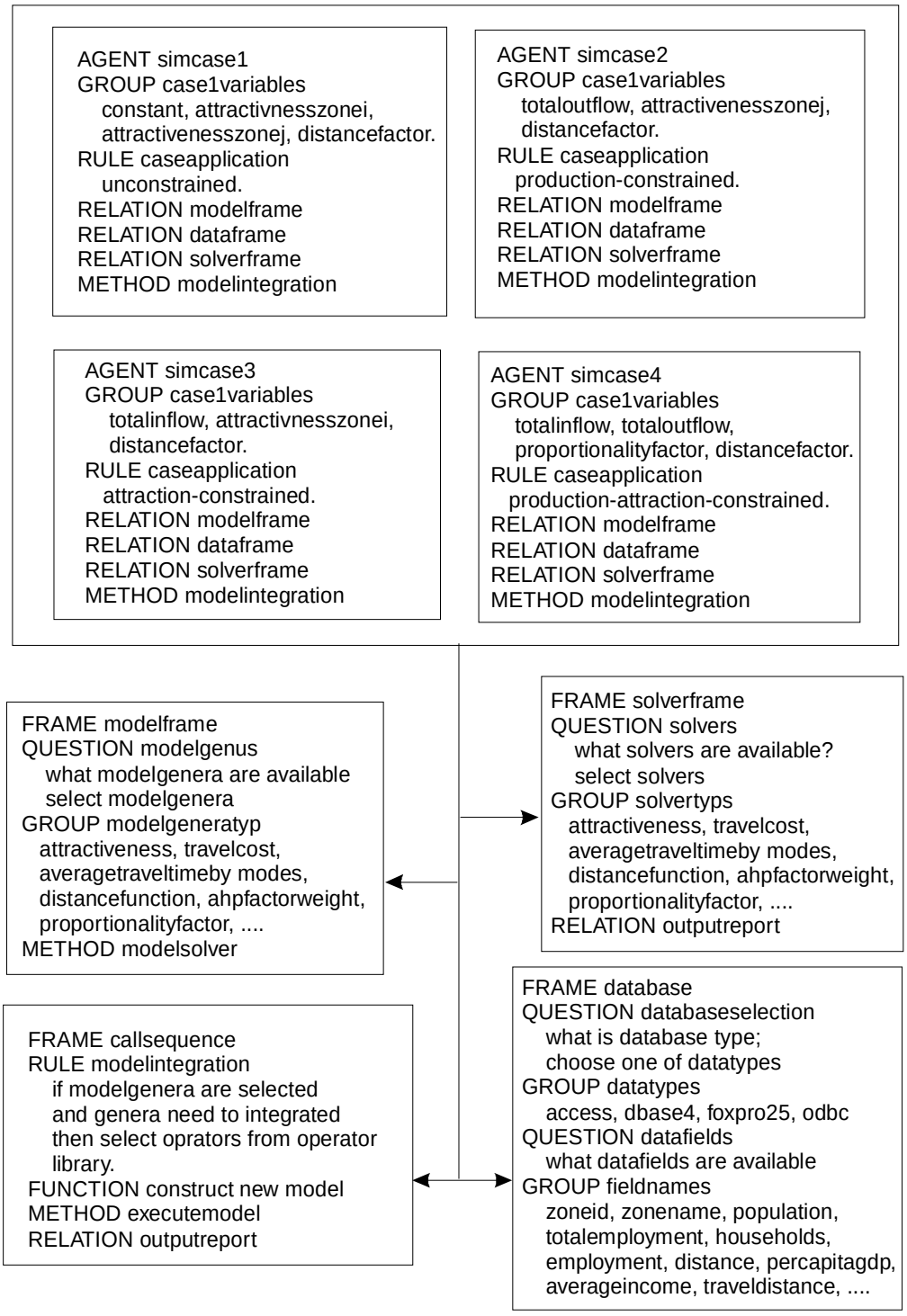

Figure 3. A simplified example of the knowledge representation for spatial interaction models

The essence of solution reasoning in solution support system is to develop an appropriate procedure for solution inference engine (facilitated by KBS agents) to retrieve models and model knowledge bases. General solution inference procedures are show in Figure 4. The system processes the user query through a set of internal procedures and messages inputted from the interface. The system then alerts that three types of analysis can be performed: rule-based analysis, model-based analysis and data-based analysis. This selection is dependent upon the nature of a spatial problem, availability of information sources and users preferences. Rule-based analysis is that the user access rule knowledge base through which the problem can be solved if the domain knowledge base is established. If the required rules are available, inference engine induces the solutions. If not, the user can assess knowledge engineering to create new knowledge base when necessary or access other types of analysis. Data-based analysis is performed by database manager through which all the functions of DBMS can be performed and integrated with GIS for visualization.

Model-based analysis is an important feature in the solution support system. It involves time-consuming and exhaustive model knowledge acquisition and knowledge base creation, and an extensive search in the model base and database in case of a model-originated query. After a user has placed a query to the system, the search procedures are as follows (see Figure 4):

1) The system first checks the availability of the required model components for the proposed analysis through inputting messages and accessing model knowledge base.
2) If the requested models are available, the system then defines which is the optimizing model and satisfactory model based on inputted messages. If models or model components are not available, then its stops searching and reports to 8) that no model is available in the model base. If new models need to be built, the system is directed to 8) for developing new model or model components.
3) If the optimizing models and satisfactory models are available, then the system select models or model components according to optimizing-first strategy and checks whether model integration is needed. If it does, it will implement the model integration procedures, then goes to 4); and if it does not, it will directly go to 4).
4) If the required data is in database, it then goes to 8) to execute models; if not, it will go through 5) and back to 1) to search for the possibility to using available models to produce the required data sets or goes to 6) to create new database.
5) If the models producing the required data are available, it will then loop 3) and 4) to select models and associated data, and further to 8) to execute model.
6) If the required database are not available, the system is prompted to the system environment to create new databases.
7) If model or model component is not available, the user is requested to create new model component and update modelbase and model knowledge base.
8) If the model and associated data are all selected, the system then formulates problem solutions in SMDS environment, then selects model solvers to execute the methods and goes to the next step.
9) Solutions and the new generated information can be displayed and further analyzed in GIS environment.

From this process, we can identify that four forms of the reasoning mechanisms are involved: (1) integration of models or model components, (2) integration of model knowledge base and model base, (3) integration of modelbase and database, and (4) controlling the execution of selected models and delivering the results to output environment. These tasks are conducted by inference engine in Flex shell with the support of a set of the predefined procedures, algorithms and calling sequences. It should be noted that the solution support system involves time-consuming knowledge engineering process and an extensive development of the model knowledge base to describe a large and growing number of models. This is particularly an exhaustive task when developing a multi-functional model knowledge base for multi-domains analysis. It should be considered as a long term and flexible information engineering and software development process.

On the basis of above reasoning procedures and model knowledge, we developed a model knowledge base for spatial interaction models. For example, a user wants to use the spatial interaction model to simulate interactions of shopping locations which are based on different planning zones with a region (or a city). The system first allows the user to input their requests to infer which model instance best fits the problem domain if he or she does not have general knowledge about the use of the models. Supposing that the production-restrained model instance is selected by the system, the system then responses to the request and access database to check the availability of data, i.e. the total amount of interaction flows leaving Zone i (O_i), the attractiveness of the

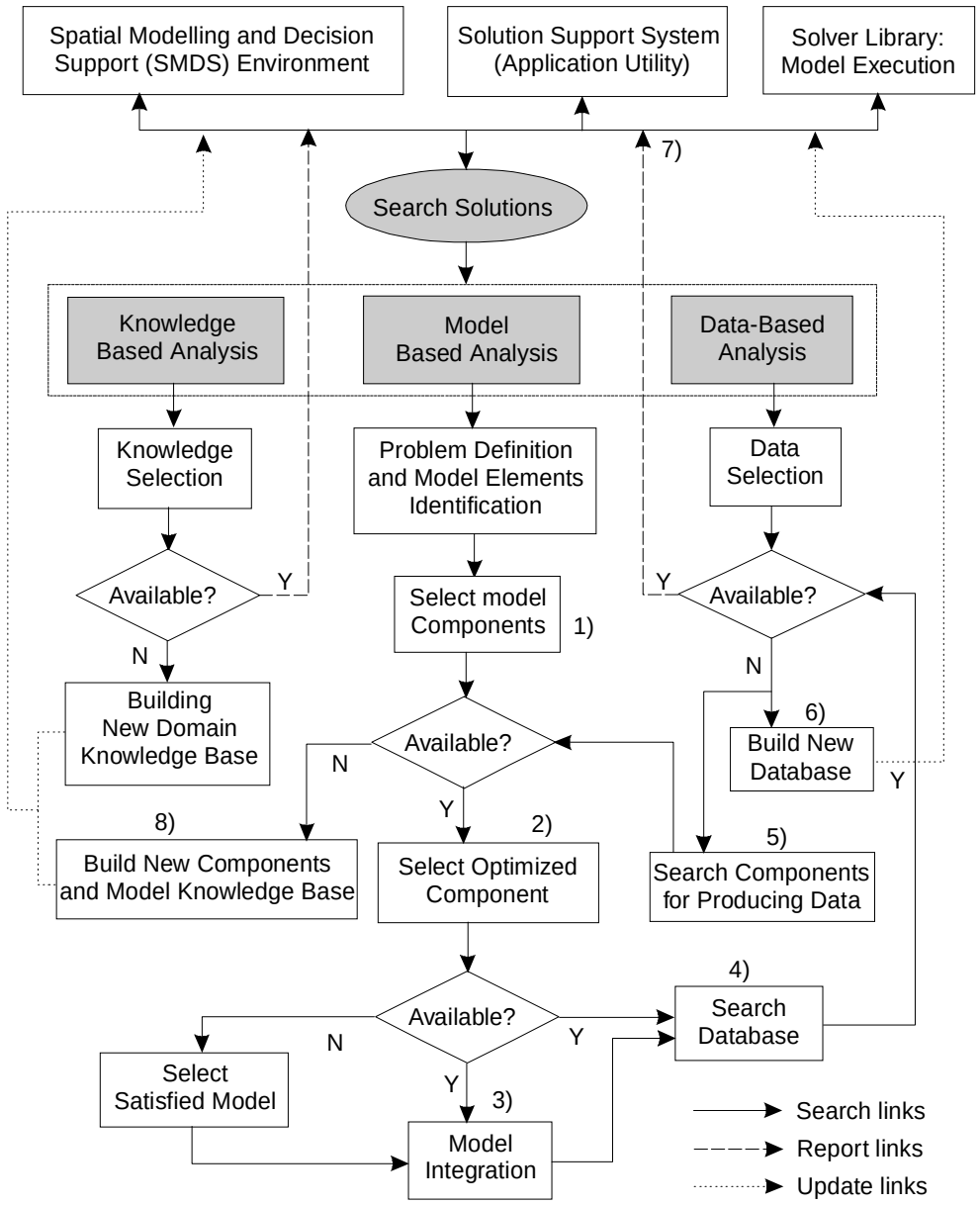

Figure 4. Model-based solution reasoning in an SDMS environment

destination Zone j (W_j) and distance factor between Zone i and Zone j ($f(C_{ij})$). If data for these variables are available, the system then directly execute the model and save the modelled results to a database for visual presentation or further analysis by GIS.

If data are not available for all or some variables, the modelbase of the system provide models (model components) to calculate the requested data. For example, the attractiveness of zones can be calculated by factor score model or factor weighting model, the amount of interaction flows can be calculated by summarizing flows to other individual zones, and the distance factor can be calculated by models such as average accessibility model, travel time or simply distances. After all the required data are available, user can execute the model based on the

procedures of model integration and model calling sequences described in the above sections.

4. Summary and further research

Model-based analysis is very important for the better understanding of spatial changes and formulation of scientific decision scenarios. Research in 'modelling inside GIS' and SDSS has not provided appropriate and flexible mechanisms for model reusing, sharing and model management. This paper proposes a 'modelling outside GIS' approach within which certain model management procedures are developed and can be extended to other systems to formulate an integrated spatial modelling and decision support environment.

One of the important issues in the proposed system infrastructure is to develop a model management framework to facilitate model reuse, intelligent model retrieval and integration, and incremental modelling capabilities. Essential mechanisms for model generalization and model knowledge representation and reasoning to achieve the above objectives are proposed and demonstrated with a reference to the spatial interaction models. An intelligent solution support (sub)system is proposed to utilize these procedures to help users to retrieve models and construct new complex models for their problem domains. This incremental modelling capability is very important for integrated spatial modelling within which a variety of models are often involved.

The intelligent solution support system presents a challenge and an opportunity for more advanced development of SDSS technology with model management and intelligence capabilities. This paper presents our current thinking and the development of the proposed systems. This can also help to advance the research on planning support systems (PSS) (Harris and Batty, 1993). Further research on the structure, communication, source building and practical evaluations of the overall system is needed.

References

[1] M. Batty, and Y. Xie, Modeling inside GIS: part 1. Model structures, exploratory spatial data analysis and aggregation, *International Journal of Geographical Information Systems*, 8 (3), 1994, pp.291-307.

[2] D. A. Bennett, A framework for the integration of geographical information systems and modelbase management, *International Journal of Geographical Information Science*, 11(4), 1997, pp.337-357.

[3] A. Bharadwaj, J. Choobineh, A. Lo, and B. Shetty, Model management systems: a survey, *Annals of Operations Research: Model Management in Operations Research*, 38, 1992, pp.17-67.

[4] R.W. Blanning, Model management systems: an overview, *Decision Support Systems*, 9, 1993, pp.9-18.

[5] C.S. Bertuglia, G.P. Clarke, and A.G. Wilson, Models and performance indicators in urban planning: the changing policy context, in Bertuglia, C.S., Clarke, G.P. and Wilson, A.G. eds. (1994), *Modeling the City: Performance, Policy and Planning*, Routledge, London, 1994, pp.20-36.

[6] W.M. Bowen, AHP: Multiple criteria evaliation, in R.E. Klosterman, R.K. Brail and E.G. Bossard (eds.), *Spreadsheet Models for Urban and Regional Analysis*, the State University of New jersey, Rutgers, 1993, pp.333-356.

[7] P.J. Densham, and M.P. Armstrong, A heterogeneous processing approach to spatial decision support systems, in T.C. Waugh and R.G. Healey (eds.) *Advances in GIS Research, Proceedings*, Sixth International Symposium on Spatial Data Handling, 1, 1994, pp.29-45.

[8] M. Gagliardi, and C. Spera, Blooms: a prototype modeling lanuage with object oriented features, *Decision support systems*, 19, 1997, pp.1-21.

[9] A.M. Geoffrion, An introduction to structured modeling, *management Science*, 33, 1987, pp.547-588.

[10] B. Harris and M. Batty, Locational models, geographical information and planning support systems, *Journal of Planning Education and Research*, 12, 1993, pp.184-198.

[11] S.Y. Huh, Modelbase construction with object-oriented constructs, *Decision Sciences*, 24(2), 1993, pp.409-434.

[12] R. Lazimy, Object-oriented modeling support system: model representation, and incremental modeling, in Nunamaker, J.F. and Sprague, R.H. (eds.), *proceedings of the Twenty-Sixth Hawaii International Conference on System Sciences (Vol.III)*, IEEE Computer Society Press, Los Alamitos, CA, 1993, pp.445-459.

[13] Y. Leung, *Intelligent Spatial Decision Support Systems*, Springer-Verlag, Berlin, 1997.

[14] LPA, *Flex Expert System Toolkit: Technical Reference*, Logic Programming Associates Ltd, London, 1996.

[15] J. Raper, and D. Linvingstone, Development of a geomorphological spatial model using object-oriented design, *International Journal of Geographical Information Systems*, 9(4), 1995, pp.359-383.

[16] A.G. Wilson, *Urban and Regional Models in Geography and Planning*, John Wiley & Sons, London, 1974.

[17] M. Wooldridge, and N.R. Jennings, Intelligent agents: theory and practice, *Knowledge Engineering Review*, 10(2), 1995, pp.115-152.