REPRESENTATION OF COMPLEX AGENTS BY FRAMES FOR SIMULATION OF INTERNAL RELATIONSHIPS IN STRUCTURAL MODELLING

Ieva Valkovska
Janis Grundspenkis
Department of Systems Theory and Design
Riga Technical University
1 Kalku str., LV-1658, Latvia
E-mail: {ieva, jgrun}@cs.rtu.lv

KEYWORDS

Agents, Agent-based Simulation, Structural Modelling, Morphological Model, Frame Model, Ontology

ABSTRACT

This paper presents a novel approach that uses an agent–based modelling to improve the dynamic relations between the objects of modelled system. The set of different frames as the representation of agent's internal structure is considered in the structural modelling domain. Main accent is paid on the usage of the created frame system skeleton that is viewed from two aspects: agent representing a system as a whole and representing its internal structure as multi-agent system.

INTRODUCTION

During decades of research done in Artificial Intelligence one of the main research objects is knowledge representation. Four categories of the knowledge representation schemes, namely, logical, procedural, network and structural representation schemes have been developed (Luger and Stubblefield 1998). Each of these representation schemes has it's own advantages and drawbacks but, in the aggregate, all of them lack dynamics, that is, extensive capabilities to represent dynamic relationships between objects and, as a consequence, they lack advanced techniques to support various kinds of reasoning. It is worth to point out that recently widely used techniques in another field – information systems development – based on various diagrams, like UML or other approaches implemented in CASE tools, as a rule, do not support any reasoning at all. They are only declarative knowledge representation schemes. In fact, the only knowledge representation scheme where the dynamic relationships are simulated is the object-oriented frame based systems. The dynamic relationships between objects are simulated by the message sending or by the use of facets (Durkin 1994).
Structural modelling approach emerges in early 70-ies and was targeted towards diagnosis problem solving in complex technical systems (Grundspenkis 1997). In structural modelling two causal models are used, e.g., morphological structure and functional structure. These models in the knowledge base are represented as a frame hierarchy. Up till now the captured knowledge helps to represent only a static view of the investigated system. Moreover, such static representation allows to support only one of the four kinds of reasoning, namely, structural reasoning (Grundspenkis 2004). Concepts of other three kinds of reasoning that are supported by structural modelling and called behavioural, diagnosis and predictive reasoning (Grundspenkis 2004) have not been implemented yet due to the lack of dynamism of the represented knowledge.
In this paper we present an ongoing research which final goal is to expand reasoning capabilities of the structural modelling. We suggest not only to use facets and message sending for simulation of dynamic relationships but, in addition, to use intelligent agent paradigm to provide dynamic relationships more effectively. Our research is based on the assumption that there should be similarities between knowledge representation in a human agent and that in an intelligent software agent. In other words, we are following Minsky`s description of a frame: "When one encounters a new situation (or makes a substantial change in one's view of the present problem) one selects from memory a structure called a *Frame*. This is a remembered framework to be adapted to fit reality by changing details as necessary" (Minsky 1975). So, we are using a set of frames as problem domain knowledge representation in the agent's knowledge base. Interpretation of the whole causal model or it's primitives as agents, from our point of view has at least two advantages. First, it provides collaboration between agents and their environment. Second, when internal structure of knowledge is considered, in agent communication a common knowledge base is used by all system's agents that is in turn, allows to implement different kinds of reasoning, using one knowledge representation scheme.

THE BASIC PRINCIPLES OF STRUCTURAL MODELING

An abstract causal domain model built within the framework of structural modelling consists from three models, namely, a model of morphological structure (MSM, in brief) and two kinds of models of functional

Proceedings 19th European Conference on Modelling and Simulation
Yuri Merkuryev, Richard Zobel, Eugène Kerchkoffs © ECMS, 2005
ISBN 1-84233-112-4 (Set) / ISBN 1-84233-113-2 (CD)

structures (FSM). Building of these models is essentially a method for encapsulating domain knowledge into small, independent, composable and decomposable units of knowledge (Grundspenkis 1997). Objects are basic units of the MSM. These primitives have input and output contacts. If interpreted in an application domain, abstract objects correspond to the components of the given system, and contacts represent their inputs and outputs. The connection of one object's output to another object's input is the only path by which the components may interact. Interactions are called flows. Flows are determined and connected to contacts by the **A**utomated **S**tructural **MO**delling **S**ystem (ASMOS). Each contact is characterised by its behaviour state. Behaviour states specify how flows "act" at corresponding inputs and outputs. A MSM is visualised as a diagram or a digraph. The MSM represents a physical structure of a given system, i.e., it represents structural relationships that can be reasoned in logic. The essence of this kind of reasoning (structural reasoning) is the exploration of paths and cycles between objects (Grundspenkis 2004). A frame hierarchy is used to represent all primitives of the MSM into knowledge base. Up to this moment structural modelling helps to create only a static view of a system under investigation. To overcome this drawback we need to encapsulate the procedural knowledge into the knowledge base to support behavioural, diagnosis and predictive reasoning (Grundspenkis 2004) as well. The aim of this study is to integrate the static frame based knowledge representation scheme used in structural modelling using the agent-based approach.

FRAME MODEL AND ONTOLOGY

Traditionally a frame system (Minsky 1975) is understood as a class – instance relationships, but in our case we are using a set of different frames, which is called a frame model. The frame model is a structure of four types of frames. Each of them is included for different purposes. There are a typical class frame and a procedure, contact and behaviour frames in the suggested structure. The frame model architecture is shown in Figure 1.

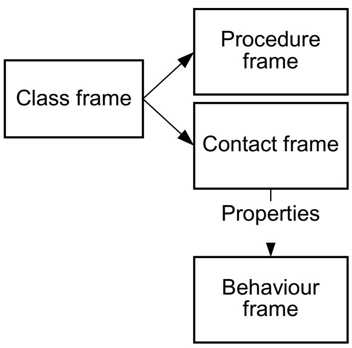

Figure 1: The main elements of the Frame Model Architecture

The special ontology (Gruber 1993) has been designed to define a common vocabulary of basic concepts and relations among frames for the created structure.

The designed ontology contains descriptions of the frame model components using categories. Categories constitute taxonomy and are described by logical expressions, which define frame structures in a formal way. Due to the scope of this paper the details about the created ontology are not presented. We give only some additional information of the objects in the frame model architecture.

Each class frame model structurally includes frame name/ identifier, supper class frame name/ identifier, properties, contact frame and procedure frame. The descriptions of these elements are as follows:

Name/ identifier – shows the meaning of some concepts that represent the existence of identity and is given to the active frame.

Supper class frame name/ identifier – corresponding on Name/identifier; allows to find the location of an active frame in the hierarchy. For the first frame (root frame) in the hierarchy of the frame system this object value is given "Haven't". This object value is predefined because the first frame doesn't have any supper class frame.

Properties – the attributes that represent qualities and quantities and allow to describe an active frame and the class related with it. There are 3 predetermined attributes for class frame {function, number of contact frames, number of procedure frames} and user defined that allows detailed representation of the information.

Contact and behaviour frames provide active components, actions and/or reactions. These frames support needed dynamic connections and the qualitative and quantitative measures that are described by categories of the created ontology.

Contact frame – data structure where information about the active components of specific frame is stored and behaviour of an object is defined. The notion of the behaviour state is introduced that represents effects on the considered active frame or effects provided by the considered active frame on other frames from its environment. These effects are characterised by matter, energy or information flows. Every frame can have more than one contact frames. Contact frame structure includes contact name/identifier, flow name, type {input, output}, connection with another class frame contact and behaviour frame.

Procedure frame structure includes procedure name/identifier, procedure and properties. The procedure frame in our approach makes it possible to separate dynamic information descriptions from static ones that are represented in typical class frame. It stores information about state changes of the frame attributes.

Procedure frame – data structure, which consists of stored information about behaviour state changes of the frame in the system and the consequences of given causes. It inspects the rules, which affects an active frame and the frames that depend on active frame, directly operating with data structures – making the

changes in the system.
Function – an attribute, which value represent the purpose or goal of an active frame that is received historically or conventionally and can be the base for a whole class. Here the purpose/goal is given which had to be done but without a warranty that it will be reached.
Number of Contact frames – an attribute whose value is calculated by counting the contact frames, those defined by the user.
Number of Procedure frames – an attribute whose value is calculated by counting of the procedure frames, defined by the user.
User defined – an attribute whose value, name and quantity are defined by the user, considering personal or expert given knowledge about each frame and/or class.
(Further C denotes Contact frame) Name/identifier – corresponding on Name/Identifier – allows to identify the adherence of contact to the defined frame and to determine the connection with another contact frame.
(C) Properties – corresponding on Properties – the properties of contact frame display the possible parameters of contact: flow name, contact type, connections and identification of the behaviour frame.
Flow name – attribute of contact frame that defines the names of ingoing or outgoing flows connected with another frame contact. If active frame's contact type is ingoing (IN), then flow connects it with another contact, whose type is outgoing (OUT).
Type (ingoing or outgoing) – contact frame attribute that define the type of contact, can be ingoing (IN) or outgoing (OUT). If the flow is going into an active contact then the type is defined as IN, if the flow is going out of the contact then the type is defined as OUT.
Connection with another class frame contact – attribute of the contact frame, whose value is specific identifier or name that can be recognized in the problem domain and which points on the contact frame.
Behaviour frame – data structure that stores the information about observed behaviour and is realized by the contacts and flows. Behaviour is defined as activity or reaction. The behaviour frame structure includes behaviour frame Name/Identifier, properties and parameter.
(Further B denotes Behaviour frame) Name/Identifier – corresponding on Name/Identifier – allows to identify behaviours frame connection with contact frame.
Parameter – component, that represents the grade of efficiency in the particular state for the flow and the contact. Parameter structure includes external link, internal link, parameter value and parameter Name/Identifier.
(B) Properties – corresponding on Properties – the properties of contact frame consists of one predefined attribute: the quantity of parameters.
Number of parameters – Attribute of behaviour frame, which value is calculated by counting user defined number of parameters that is marked as enforceable. It means that the attribute should be defined or calculated by user's defined function.
Name/Identifier of the parameter – corresponding on Name/Identifier – the attribute of parameter, which clearly, completely and briefly shows the meaning of the parameter and connection with the active frame and related flow.
Parameter value – attribute of parameter that shows importance, efficiency and quality or quantity of parameter, related with the name of given parameter.
External link – defines the relationships between current frame parameter values and another external frame parameter values for the current state. Used to sent or receive resources from the external structures.
Internal link – defines the relationships between internal frame parameter values for the current state.
(Further P denotes Procedure frame) Properties – corresponding on Properties – at the properties of procedure frame one attribute is located – the quantity of procedures.
(P) Name/Identifier **-** corresponding on Name/Identifier – procedure frame name is given to identify that the upper (lower) level frame procedures are included.
Procedure – structure, which shows the activity, steps of the task and instructions, and which is realized by the users defined criteria. Procedure consists of the users selected procedure type, action and description. Procedure do not specify the best solution but executes the user's defined one.
Number of procedures – attribute of the procedure frame, which value is calculated by counting users selected number of procedure types, marked as enforceable.
Type – the attribute of the procedure, which value defines semantic meaning of enforceable action and in this case allows a certain type of classification. It is expected that there are 4 types:
"If added" – procedure is executed in case when the property or the value of the property is added to the frame. Procedure can affect the properties of an active frame and a subclass frame (name and/or value);
"If needed" – procedure is executed in case when the property or the value of the property need to be added to the frame. Procedure can affect the properties of an active frame and a subclass frame (name and/or value);
"If deleted" – procedure is executed in case when the property or the value of the property is deleted from the frame. Procedure can affect the properties of an active frame and a subclass frame (name and/or value);
"If changed" – procedure includes 2 types of procedures – "If added" + "If deleted". Procedure is executed in case when the property or the value of the property is deleted and then added to the frame. The procedure can affect the properties of an active frame and a subclass frame (name and/or value);
Action – the attribute of the procedure, which shows the reaction on any selected type of procedure in current a frame or in the operating area that shows the work mechanism.
Description – the attribute of the procedure, which

describes the actions, meaning, relations and consequences of the procedure.

The detailed frame model architecture is shown in Figure 2.

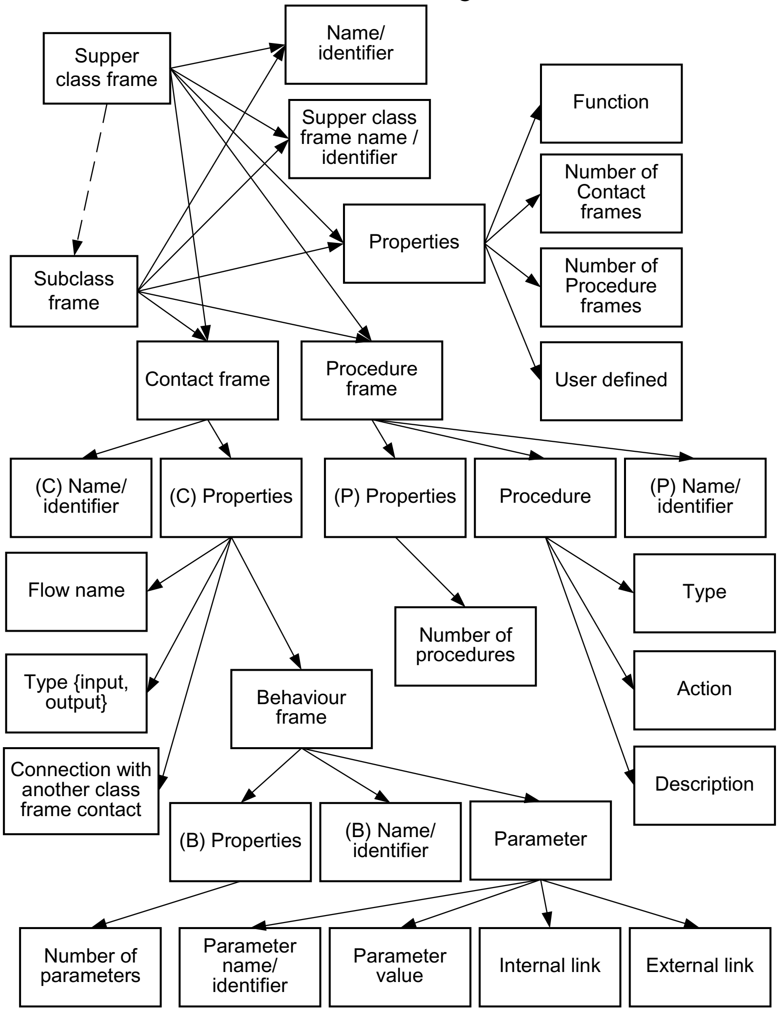

Figure 2: Detailed Frame Model Architecture

Proposed generalisations of the frame system is applicable to a variety of systems that include different types of objects. We suggest to use a frame model as a base for an agent representation. In this case the object of the morphological model is replaced by the created frame model. Replacement makes changes in the frame hierarchy developed in structural modelling.

FRAME SYSTEM AND INTELLIGENT AGENTS

"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors." (Russel and Norvig 2003)

Agents usually are represented without internal structure. We suggest to use frame system as an internal structure of an agent to provide more sophisticated kinds of reasoning.

We assume that it is possible to join several frame models and as a consequence to create a unified system. The result is a frame system skeleton. This skeleton can be considered from two aspects. First, the skeleton can be interpreted as a set of included frame models and their internal links. In this case a system is comprehended as one unified component of an intelligent agent. Second, the skeleton can be interpreted as a multi–agent system considering each frame model as a separately functioning agent. The first aspect can include the second one and they can be viewed as two different elaboration levels.

From the first viewpoint, the structure of frame models may be viewed as a component in the intelligent agent. Below a special defined standard notation for a component is given, which conditions are determined by the created ontology with its logical definitions and categories. The frames are divided into two groups – the frames that belong to the external and the frames that belong to the internal level. The external level frames interact with the environment and with frames that belong to the internal level. The internal level frames interact only with frames that belong to the internal level. Flows from more than one frame model reach the external environment. Thus the resulting flow is gained from the reaction of the whole network of frames. In this case every frame model has its determined meaning and importance level in a system as whole. The common understanding of this kind of agent architecture is shown in Figure 3., but decomposition of the frame system skeleton can be viewed in Figure 3 while the internal structure of the frame system skeleton obtained by the decomposition is presented in Figure 4. In point of fact, Figure 4 depicts a multi agent system in which each F_i is interpreted as an individual intelligent agent represented by the corresponding frame model.

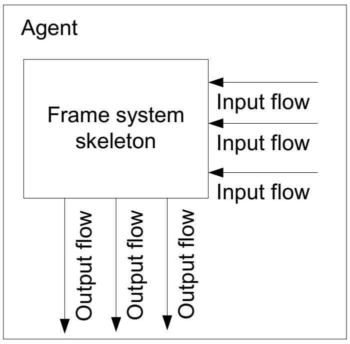

Figure 3: System of Frame Models representing one agent

Each frame model as an individual agent receives several input flows and acts by sending output flows to some other agents according with the functionality of the system under investigation. The frame model that represents the internal structure of an agent's knowledge base is shown in Figure 2. Communications between frames are based on the developed ontology.

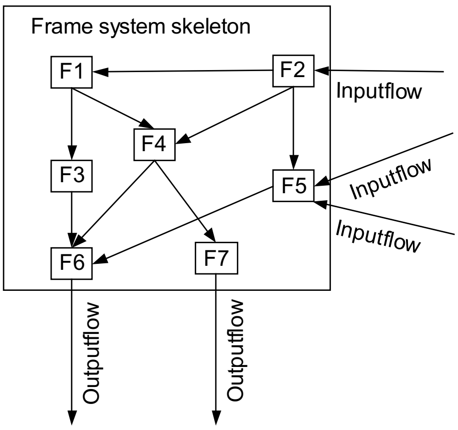

Figure 4: Frame Model representing the agent architecture

INTERPRETATION OF THE FRAME MODEL AS AN AGENT

Proposed approach provides agents with percepts mater, energy and/ or information and their quantitative and qualitative measurements. Following the idea that agents usually can be viewed according to their type, perceptions, actions, performance measures and they are embedded in different types of the external environment, in the proposed frame system skeleton one can draw the following parallel:

1. It is possible to define the agent type with class frame. We propose to define the type using one (if looking from the second aspect) ore more frames (from the first aspect side).
2. Perceptions can be represented by input flows described in the contact frame. It is possible to represent also output flows that deliver the information to executive mechanisms. In the first aspect input conditions have been provided by external environment components and common reasoned outputs affect an external environment. Internal processes on the highest level of abstraction can are hidden. In the second aspect input conditions have been provided by external environment components and/ or another agent from multi–agent environment.
3. Actions can be specified with the frame structure that is provided by the procedure frame and in behaviour frame parameters that determine the function results and quality. In the first aspect actions will be specified with several frame models that have specifically similar meaning in a system. In second aspect actions will be specified only in terms of one frame model.
4. Performance measures are provided by the behaviour frame. The created ontology defines the meaning of measures.
5. In the first aspect the experts can play the role of the external environment. In addition to the frame model an extra structure can be provided. It is used to save the identifier of the frame system creator. In the second aspect the external environment are experts and other agents from multi–agent environment. Information about other agents is stored using contact frame attributes.

To demonstrate the first aspect we use the wumpus world example (Russel and Norvig 2003) where agent is specified by a proposed frame system skeleton. The agent lives in some area (4*4) with pits, wumpus and gold. The goal of the agent is to grab the gold, not to meet the wumpus and not to fall in a pit. It can move in all directions and perform some actions like "Grab" and "Rotate by 90°". The agent can feel the breeze if nearby is the pit, can feel the stench if nearby is the wumpus and can see the glitter if nearby is the gold.

For this example in the frame model architecture the frame name/identifier is "Gold agent", the supper class frame name/identifier is "Logical agent". The function of the Gold agent is "To grab the gold". In this case the number of contact frames and procedure frames is "1", but, in general, there can be any other positive number of frames if needed. The user-defined properties is **step** with value "1", **possible rotation angle_1** with value "90°", **possible rotation angle_2** with value "180°", **possible rotation angle_3** with value "270°", **area** with value "4*4". The number of procedures is “1”. The procedure name/identifier is “Change the step value”. The type of procedure is “If changed” and the action is “IF step>1 THEN step=1. The description of the procedure will be “If some user changes the step value it is changed to previous”. The name/identifier of contact frame can be “square [2,2]” that means agent is located in the square [2,2]. The flow name is “breeze” that means agent’s sensors have detected the breeze in the square [2,2] and type value is “Input”, because agent feels breeze. Connection with another class frame contact can be “pit” to denote that adjacent squares can contain pits. Behaviour frame name/identifier is “move to square [2,3]”. The number of parameters can be “1”. In this example we suppose that parameter name/identifier is “state costs”, parameter value is “-0,4”. The internal link is “2,2_to_2,2” if agent doesn’t move and the external link is 2,2_to_2,3 if agent moves to square [2,3].

In this paper we don’t give interpretation of the second aspect.

CONCLUSIONS AND FUTHER WORK

In this paper is discussed an ongoing research that uses an agent – based modelling to improve the dynamic relations between the objects of the modelled system. The combination of different frames as the representation of agent’s internal structure is considered in the structural modelling domain. Main accent is put to the usage of the created frame system skeleton that is viewed from two aspects. In both cases, as a consequence, the automatic reasoning is effectively supported. All provided assumptions are only conceptual ones now but according to them we will try to create a real system prototype that will provide possibilities to represent necessary knowledge and to realize simulation of communications by implementation of different kinds of reasoning.

REFERENCES

Durkin, J. 1994. „Expert System (Design and Development).” New York, Macmillian Publ. Comp.

Gruber, T. R. 1993. „A Translation Approach to Portable Ontology Specifications.” *Knowledge Acquisition*, 5, (2), 199-220.

Grundspenkis, J. 1997. „Structural modelling of complex technical systems in conditions of incomplete information.” Modern aspects of management science, No.1, Riga, Latvia, 111-135.

Grundspenkis, J. 2004. „Reasoning in structural Model – Based Diagnosis.” *Proceedings of 4th International conference on Quality, Reliability and Maintenance, QRM 2004,* Oxford, March 21-22,G.J. Mc Nulty (ed.), Profesional Engineering Publicity, London, UK, 295-298.

Luger, G.F. and W. A. Stubblefield. 1998. „Artificial Intelligence. Structures and Strategies for Complex Problem Solving”. Addison Weslay Longman, Harlow, England.

Minsky, M. 1975. „A framework for representing knowledge.” *The Psychology of Computer Vision,* P. Winston (ed.), New York, McGraw Hill, 211-277.

Russel, S.J. and P. Norvig. 2003. „Artificial Intelligence: A Modern Approach.” Prentice Hall, Englewood Cliffs, N.J.

AUTHOR BIOGRAPHIES

IEVA VALKOVSKA is a master student at Riga Technical University. Her scientific interests are associated with system analysis, design and development of the knowledge representation systems and the systems for structural modelling. Contact e-mail address is ieva@cs.rtu.lv and contact address is Riga Technical University, 1 Kalku str., LV-1658, Latvia.

JANIS GRUNDSPENKIS is the professor at Riga Technical University. He also is the dean of the Faculty of Computer Science and Information Technology, the director of the Institute of Applied Computer Systems and the head of the Department of Systems Theory and Design. His research interests are agent technologies, knowledge engineering and management, and structural modelling. He has published more than 150 scientific papers. Contact him Riga Technical University, 1 Kalku str., LV-1658, Latvia; jgrun@cs.rtu.lv.