On the representation of expert procedural knowledge

Reprezentace expertní procedurální znalosti

A. Veselý

Czech University of Agriculture, Prague, Czech Republic

Abstract: Procedural knowledge is used by experts for complex system control. In this article, the notion of a complex system is taken in a broad sense. It might be a patient cured by a physician specialist, a biotechnological device, a department of some business enterprise etc. The GLIF model was designed in collaboration of American universities for the formalization of medical guidelines, but it can be used for formal representation of any procedural knowledge. The main objective of the GLIF model was to enable computer processing and comparing of medical guidelines. In this, article also a more sophisticated use of procedural knowledge representation by the means of the GLIF model is proposed. The provided data about expert actions are stored into the database, the formalized knowledge represented by the GLIF model can be used for building up sophisticated reminder systems that warn the user if he decides to make an impropriate action. Different kinds of warnings in the reminder system are proposed and their properties are discussed. At the end, also the possibility of using the GLIF model for decision support is discussed.

Key words: representation of knowledge, reminder system, GLIF model, GLIF browser

Abstrakt: Procedurální znalost je používána experty pro řízení složitých systémů. Pojem složitý systém je v tomto článku chápán v obecném smyslu. Složitý systém může být například pacient léčený lékařem specialistou, biotechnologické zařízení, oddělení obchodní společnosti atd. GLIF model, který byl navržen ve spolupráci amerických univerzit pro formalizaci lékařských doporučení, může být použit pro formální reprezentaci jakékoliv procedurální znalosti. Hlavním cílem GLIF modelu bylo umožnit počítačové zpracování a porovnávání lékařských doporučení. V tomto článku jsou navrženy další, sofistikovanější způsoby jak lze procedurální znalost, reprezentovanou GLIF modelem, použít. Pokud jsou data o akcích experta zaznamenávána do databáze, formalizovaná znalost reprezentovaná GLIF modelem může být použita pro vytvoření sofistikovaných připomínkových systémů, které varují uživatele v případě, že se rozhodne provést nevhodnou akci. Jsou navrženy různé varianty upozornění a jsou diskutovány jejich vlastnosti. Nakonec je diskutována možnost použití GLIF modelu pro podporu rozhodování.

Klíčová slova: reprezentace znalostí, připomínkový systém, GLIF model GLIF prohlížeč

In this article, the representation of procedural knowledge that experts use for the complex system control and its utilization in practice is pursued. The notion of a complex system is taken here in a broad sense. It might be for example a department of a business enterprise, a biotechnological device or a patient who is cured by a physician specialist. The common property of all these complex systems is that their behavior cannot be effectively described by a mathematical model and therefore their control could not be automatized in a simple way. In spite of that, competent experts successfully control these systems. They intuitively know how to respond to changes of the system states or the system behavior. Therefore, they are said to have some kind of implicit procedural knowledge. Explicit and precise formulation of this implicit knowledge, however, is usually very difficult.

Sometimes groups of experts formulate procedural knowledge in natural language in a form of so-called guidelines. For example in 1999 experts in cardiology from the WHO (World Health Organization) worked out medical guidelines that described how to treat patients with the hypertension disease (Mancia 2003).

Formulation of procedural knowledge in natural language is not, however, sufficient and some formal means of representation are needed from the following reasons.

At first, transformation of procedural knowledge from natural language into some formal representation usually finds out inconsistencies and ambiguities that can be corrected after the discussion between knowledge engineers and experts.

At second, it enables more a precise comparing of guidelines used by different teams of experts in different countries or institutions.

In this article, we propose further a more sophisticated use of formally represented guidelines. Formally represented guidelines can be coded and processed by a computer. Suppose that guidelines describe expert procedural knowledge concerning complex system control. If the values of the measured state parameters and values of the control parameters of the controlled system are stored into a database, then formally represented guidelines can be used for decision support and for design of a sophisticated reminder system. Such reminder system can compare the values of state and control the variables stored in the database with guidelines. It can recognize the non-compliance of the stored data and the guidelines and warn the user.

Formal representation of procedural knowledge is at present pursued mainly in the field of medical informatics. The objectives of this effort are improvement and unification of medical care. The author is persuaded that those methods might be utilized with the same benefits also in other application fields, for example in economic management.

REPRESENTATION OF PROCEDURAL KNOWLEDGE

A process is an evolving sequence of states and events. A procedure is a pattern that determines the types of states and events that may occur in the entire family of processes. Each process in the family is called an activation of the procedure. A procedure can consist of more simple procedures. Knowing procedures means to have procedural knowledge (Sowa 1999).

Processes defined by these procedures can run sequentially or concurrently. According to this, the whole procedure can be called sequential or parallel. Procedures determined for running on the original von Neumann computer are sequential, because sequential is the architecture of the von Neumann computer. Sequential procedures can be described by the means of:

1. Flow charts
2. State-transition diagrams
3. Finite-state machines

On modern computers, also parallel or concurrent processes can be run. The Petri nets are a generalization of the state-transition diagrams developed for representing the concurrent processes. The Petri nets have been also adopted as the basis of activity diagrams in the Unified Modeling Language (UML).

Procedures determined for the complex system control are in general parallel. Therefore the Petri nets should be used for their description. The Petri nets diagrams, however, are difficult to understand for the untrained user. And besides, the parallelism in procedures used for the complex system control is limited. Therefore, if the symbolism of flow charts is enlarged in such a way to be able to accommodate the limited parallelism of procedures, flow charts can be taken as a basis of the procedural knowledge representation.

The complex system control starts from some state P that does not need to be the initial state of the system. In addition to that, some states of the controlled system usually have commonly used names what makes understanding flow charts simpler. Therefore, flow charts should be enlarged by symbols for states.

Procedures that define the run of the Von Neumann machine are deterministic. Therefore, branching in flow charts is carried out uniquely according to the result of evaluation of the branching conditions. When representing procedural knowledge, experts are usually unable to formulate such unique conditions of branching. Therefore, branching under condition must be conceived in a more general way.

To control a complex system means:
1. To measure state variables of the system.
2. To control the run of the system by changing values of control variables.

From the ontological point of view, both activities can be considered to be events. When formulating procedural knowledge, experts use notions from some commonly shared ontology. Among these notions, there must be also the state and control variables of the controlled system. These variables can be binary or continuous and according to that, they are described by functions or predicates. Although the ontology is not usually explicitly and exactly defined, from the properties of notions and from the background knowledge validity of some formulae it can be deduced. In knowledge engineering, the set of these valid formulae is usually called the T-box or theoretical knowledge. Their conjunction

we denote Φ_T. Results of the measurements of state variables and the information about the carried out control actions, are called when expressed by formulae the factual knowledge or A-box. The conjunction of these formulae we denote Φ_A.

GLIF MODEL

The GLIF model is a result of collaboration among the Columbia University, the Harvard University, the McGill University and the Stanford University for representation of procedural knowledge contained in medical guidelines (Ohno-Machado 1998). The main goal of the GLIF was to enable sharing of the medical guidelines among institutions and across computer applications. The GLIF specifies an object-oriented model for guidelines representation and syntax for guidelines utilization in software systems as well as for their transport. The GLIF model could be represented in the form of an oriented graph. The nodes of the graph are guidelines steps and the edges represent the continuation from one step to the other. The guidelines steps are *action step*, *state step, decision step*, *branch and synchronization steps* (see Figure 1).

Action steps represent events. In the representation of medical guidelines, action steps specify clinical actions that the physician is supposed to perform. The action step also may name the sub-guidelines, which provide a greater detail for the action.

State steps represent states of the controlled system.

Decision steps are used for conditional branching. Branching conditions that belong to some edge coming from decision step can be one of the following types: *strict in*, *strict out*, *in* and *out*. If a strict out condition is fulfilled, the continuation along the appropriate edge is forbidden. Strict out conditions are evaluated at first. If the edge has no strict out condition or its strict out conditions are not fulfilled, the strict in conditions are evaluated. If the first evaluated strict in condition is fulfilled, the continuation goes along this edge. If strict in and strict out conditions do not define uniquely how to continue, the user must make the decision himself. The conditions out and in facilitate his choice.

Branch and synchronization steps enable to introduce the concurrency into the model. Guidelines steps that follow the branch step can be performed concurrently. Branches with the root in branch step eventually converge in the synchronization step. In the synchronization step, all branches are synchronized. It means, that actions that follow the synchronization step cannot be performed, unless all actions following the branch step and preceding the synchronization step are finished.

The advantages of the procedural knowledge representation by the GLIF model are the following:
1. GLIF graph is clear and easily comprehensible
2. GLIF model can be easily coded and processed by computer

At present time, when computers are widely used in practice, the data that the user utilizes during the complex system control are often stored in a database either automatically or by the user himself. If it is not the case now, it might be true in the near future. Let us take a physician who cures a patient as an example. Nowadays, most of physicians use computer for recording the patient data even though only in the form of free text that computer cannot process automatically. But the electronic health record (EHR) that enables to store patient data in a structural way and thus to process them automatically by computer is under development. At the same time, the interconnection of the EHR databases is considered in order that all relevant data about the examined patient are at hand at the place where the patient is examined and cured.

Databases containing structured data in the connection with formal procedural knowledge models provide new means for rendering the complex system control more efficient. The formal model of procedural knowledge, as for example the GLIF model, can be used together with databases that contain the data about the controlled system for:
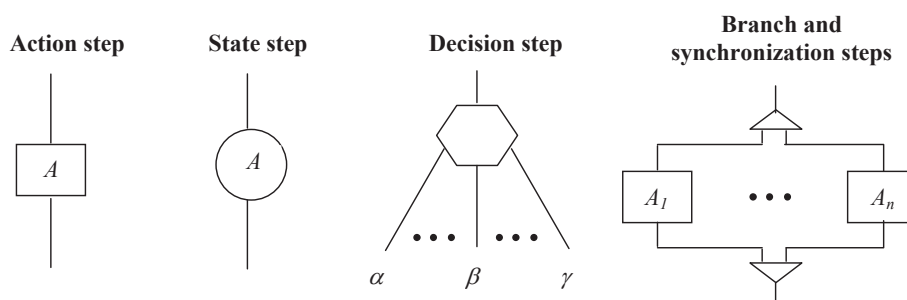

Figure 1. Steps of the GLIF model

1. Design of reminder systems
2. Decision support (Veselý et al. 2003; Zvárová et al. 2004)

USING THE GLIF MODEL
FOR THE REMINDER SYSTEM DESIGN

To be able to build up a reminder system, the user is supposed to monitor the controlled system and to store the results of his monitoring and control actions into the database. The reminder system determines if the executed action (the measurement or control action), the result of which is just stored into the database, is or is not in compliance with the procedure description given by the GLIF model. Therefore, the reminder system must know also the results of the previous actions and therefore it must have a connection to the database, from which these results can be taken.

In the simplest case, the reminder system may work in the tight connection with the GLIF model. Suppose that the system control starts in a state P. Then the reminder system makes inquiry if the events, the results of which are put into the database, are the same as those required by the GLIF model. In decision steps, if the following action is not unambiguously determined by strict conditions, the reminder system follows all admissible branches (admissible branch is that branch, which is not excluded after the evaluation of strict conditions). As soon as the just carried out action is not in compliance with some admissible branch, that branch is excluded. If the last admissible branch is excluded, then the just carried out action is not in compliance with the GLIF model and the reminder system immediately warns the user. The reminder system that behaves in the described way will be further called a reminder system with a tight connection to the GLIF model.

This tight connection between the reminder system and the GLIF model is not always appropriate. The reminder system warns the user any time when the user does not store the result of some action prescribed by the GLIF model into the database or when he does not retain the sequence of actions.

The reminder system may work also in loose connection. To be able to describe the design principles of such reminder system, it is necessary to give the following definitions.

Branch $S(P)$ of the GLIF model graph is a sequence of the consecutive nodes and edges that begins with the node P, in which none of its nodes or edges are repeated.

The support of the branch $S(P)$ is the conjunction $\Phi_{S(P)}$ of all strict in conditions and negation of all strict out conditions, which lay on this branch.

Branch $S(P)$ with the support $\Phi_{S(P)}$ is according to the theoretical knowledge Φ_T and according to the actual knowledge Φ_A ***admissible***, if the formulae

$$\Phi_{S(P)} \wedge \Phi_A \wedge \Phi_T$$

is consistent.

The formula is consistent if it has a model. It means that under certain circumstances, the formula might be valid. In other words, it means that unknown values of parameters and predicates included in the formula might be such that the formula is valid.

Using the preceding definitions, the following less strict, degrees of non-compliance of some user action and the GLIF model could be defined.

Suppose that the user controls the system from the state P. When the user inputs into database the result of some action A and this action is not comprised in any branch $S(P)$ starting in the node P, then the execution of the action surely was not in compliance with the GLIF model. In this case, the reminder system must warn the user and we will say that the action was carried out of the context of the GLIF model.

If the action A was not carried out of the context, then during the input of the result of the action into the database one of the following cases had to occur (see also Figure 2).

Case 1

There is at least one admissible branch $S(P)$ that includes the action node A and there is at least one admissible branch $S(P)$ that does not include the action node A. In this case, the action A might be or

The number of admissible branches without the node A	The number of admissible branches with the node A	
	at least one	none
at least one	Case 1 Weak warning	Case 2 Strong warning
none	Case 3 Without warning	Case 4 Inconsistent data

Figure 2. Possible kinds of warnings in a reminder system

might not be in compliance with the GLIF model. The compliance with the GLIF model could not be resolved, because the database does not contain all measurements required by the GLIF model or in the guidelines, the decision is left on the user.

Case 2

There is no admissible branch $S(P)$ that includes the action node A and there is at least one admissible branch $S(P)$ that does not include the action node A. In this case, the execution of the action A surely was not in compliance with the GLIF model. The necessary data for the unambiguous continuation according to the GLIF model might be or might not be in the database.

Case 3

There is at least one admissible branch $S(P)$ that includes the action node A and there is no admissible branch $S(P)$ that does not include the action node A. This case occurs if the content of the database unambiguously determines that action A should be executed according to the GLIF model.

Case 4

There is no admissible branch $S(P)$ that includes the action node A and there is no admissible branch $S(P)$ that does not include the action node A. This case occurs if the GLIF model or the data are inconsistent. Under the assumption of the consistently defined GLIF model, the reminder system can inform the user about the inconsistency of the data stored in the database.

Suppose that the reminder system with the loose connection to the GLIF model checks the compliance of the action, the result of which is going to be put into the database with the GLIF model:

In the case 1 (weak warning), the reminder system cannot find out whether the user action is in compliance with the GLIF model or not due to the missing data. In this case, the user should be warned that he is going to carry out an action for which he has not the sufficient support in the database.

In the case 2 (strong warning), the user decided to carry out an action that surely contradicts the GLIF model and must be immediately warned.

In the case 3, the reminder system does not generate any warning.

In the case 4 (inconsistent data), the user should be warned that the patient data are not consistent.

The reminder system can be designed with a tight or loose connection to the GLIF model. The reminder system with a tight connection is able to catch even small departures from the GLIF model. In applica-

tions in which the procedure defined by the GLIF model must be strictly adhered to, it is its a great advantage. On the other hand, in the applications in which to stick strictly to the GLIF defined procedure is not important, the user might be deluged by a lot of non-important warnings. It would be for example in situations when it is not necessary for the user to carry out all by the GLIF prescribed measurements or when the sequential order of actions is not important. In this case, it is better to use the reminder system with a loose connection to the GLIF model, which is able to catch only the substantial departures from the GLIF model.

The GLIF model is a formal representation of procedural knowledge, which describes how to control a complex system. Therefore, the GLIF model can be used also as a mean of decision support. In this case, the GLIF model provides a general view. The user can see the right decision in the context. He can see under which conditions the alternative decisions are made and which actions follow.

When the GLIF model is coded in some standard way, it can be represented on computer display by the means of the GLIF browser. If the browser is connected to the database, it can mark in the specified color the branch of right decisions, provided they can be uniquely determined by strict conditions. And it can mark in a different color the branches, which are excluded according to the GLIF model.

The system control with reminder and decision support facilities based on the GLIF model is described on Figure 3. The user is supposed to put into the database all information about the system behavior and the results of his decisions. For example, the user may be a physician who puts into database the patient's symptoms and results of laboratory tests and also the information about his therapeutic decisions. The reminder system compares the input data with the GLIF model and in the case of non-compliance, it warns the user. As soon as the user is not certain about his next action, he can activate the GLIF browser and the GLIF browser will display the relevant part of the GLIF model on the screen.

CONCLUSION

Formal models of procedural knowledge enable computer processing. The methods of the complex system control used by different experts or expert teams can be compared and the best methods can be standardized. The standardization results in a more efficient control. Therefore, the formal models of procedural knowledge representation are intensively
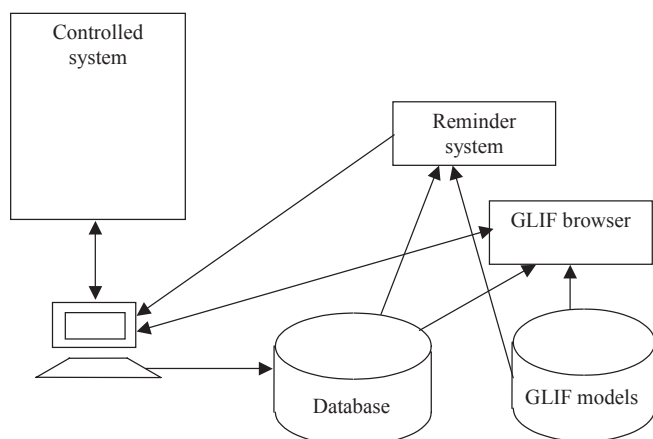
Figure 3. Reminder system and decision support

studied, especially in the field of medical informatics. The provided data about expert actions are stored into the database, the formal models of the procedural knowledge can be used also for building up the sophisticated reminder systems and decision support systems. The first realizations might be expected in the medical informatics, where the procedures of diagnostic making and patient treatment are described in so-called medical guidelines. Medical guidelines have already been worked out by the teams of experts for many groups of diseases. These guidelines are written in natural language, but they can be easily transformed into a formal representation, for example into the GLIF model.

The first applications of the GLIF models concerned decision support. An universal GLIF browser controlled by data was realized and used in the field of heart diseases in the European Centrum for Medical Informatics, the Epidemiology and Statistics (EuroMISE) (Veselý et al. 2004; Buchtela et al. 2004). The above described reminder systems, however, have not yet been realized. The advantage of such reminder systems, however, is obvious and therefore their realization can be expected in the near future.

REFERENCES

Buchtela D., Veselý A., Peleška J., Zvárová J. (2004): Presentation of Medical Guidelines on a Computer. Transformation of Healthcare with Information Technologies. IOS Press Amsterdam, pp. 166–171; ISBN 1-58603-438-3.

Mancia G. (ed.) (2003): 2003 European Society of Hypertension-European Society of Cardiology guidelines for the management of arterial hypertension. Journal of Hypertension, *21*: 1011–1053.

Ohno-Machado L., Gennari J. H., Murphy S.,N., Jain N.,L., Tu S., W., Oliver D., et al. (1998): The Guideline Interchange Format: A model for representing guidelines. Journal of the American Medical Informatics Association, *5* (4): 357–372.

Sowa J.F. (1999): Knowledge representation: Logical, Philosophical and computational foundations. Portland.

Veselý A., Zvárová J., Peleška J., Tomečková M. (2003): On Direct Comparing of Medical Guidelines with Electronic Health Record. In: Proceedings of MIE2003, IOS San Malo.

Veselý A., Zvárová J., Peleška J., Anger Z., Buchtela D. (2004): Computerized Presentation of Medical Guidelines. In: Proceedings Medinfo 2004 AMIA, San Francisco.

Zvárová J., Veselý A., Hanzlíček P., Špidlen J., Buchtela D. (2004): On Direct Comparing of Medical Guidelines with Electronic Health Record. In: Bubak M. et al. (Eds.): Computational Science, Part IV. Springer-Verlag, Berlin, pp. 1133–1139.

Contact address:

Arnošt Veselý, Czech University of Agriculture in Prague, Kamýcká 129, 169 51 Prague 6-Suchdol, Czech Republic
tel.: +420 224 382 356, e-mail: vesely@pef.czu.cz