Industrial Applications of Artificial Intelligence
J.L. Alty and L.I. Mikulich (Editors)
Elsevier Science Publishers B.V. (North-Holland)

pp.115-120

A PROCESS EVENT KNOWLEDGE MODEL FOR INDUSTRIAL EXPERTISE

Alexandre V. YAKOVLEV and Alexej I. PETROV

Department of Computing Science
Leningrad Electrical Engineering Institute
Leningrad 197022 USSR

The modelling of event-oriented knowledge is a key problem in designing intelligent support systems for industrial experts who operate in such domains as equipment assembly/repair process planning or tracing of events in the diagnosis of faulty equipment. In this context, we use labeled event nets, a subclass of Petri nets, for which two types of semantics, the event semantics and the equipment state semantics, are introduced with the examples of properties inherent to them. Finally, a prototype system architecture, called PREFIX, is briefly described.

1. INTRODUCTION. MOTIVATION OF EVENT KNOWLEDGE MODELING

One of the most crucial problems in designing intelligent support systems for the industrial process control staff involved in (Case A) the development of process technology charts for assembly-disassembly-repair of various equipment or (Case B) diagnostic activities for faulty equipment consists of modeling and representing the several kinds and levels of event-oriented knowledge [1-3], which can be regarded as a specific kind of procedural knowledge [4]. (The application domain of such knowledge may also include analysis and design of organizational systems [5], office automation [6] to name but a few.) The importance of event knowledge stems from the following, though not all possible, factors. In Case A, the correct cost evaluation of a process operation (e.g. the "changing a car brake disk" operation requiring the "removing a wheel" operation, which in its turn requires "removing a hubcap", "unscrewing a bolt" etc.) can only be achieved if a proper estimation of the corresponding process chart is ensured. In Case B, when some device must be diagnosed, the expert should establish the course of events relevant to the fault. In doing so, he or she matches the equipment operator's "story" with some of the generic "event cliches" of potential faults.

Unlike many application domains using static knowledge representations, such as semantic networks, descriptive frames, AND-OR trees and production rules, the above paradigms require an adequate formal model for representing the system of events (the so-called "deep knowledge" [7]) which are temporally ordered by causal relationships and occur through the interaction of a set of communicating agents. An agent can be interpreted by such operations, atomic for a certain abstraction level, as "removing a wheel", "unscrewing a bolt", "switching-on a fuel pump" etc. An agent is the object (person) having a sequential, or linear, behavior (at any moment, at most one action or operation is allowed), which can be synchronized, at certain actions, with the other agent behaviors, e.g. in adjusting the ignition mechanism of a car, one of the agents operates the starter while the other agent assesses the quality of spark. Such a "joint venture" must therefore be included in the behavior of each of the participating agents.

Petri nets have been suggested [8] as a useful tool for event knowledge representation. In this paper, we use labeled event nets, which are syntactically close to occurence nets of [7]. They are a subclass of Petri nets, quite sufficient for the indicated purposes.

2. LABELED EVENT NETS

A labeled event net (LEN) is defined as a partially ordered set (poset) $< E,0,\mu,\leq >$ represented by an acyclic graph, whose vertices correspond to events in set E, labeled with "process technology" operations in set 0 by means of a labeling function $\mu : E \dashrightarrow 0$, and arcs stand for the $\leq$ (partial order) relation.

Each operation in 0 is assigned to a particular technological asset - the element or component of equipment involved in the operation. Hence, each operation implies some discrete change of a state of a particular component. For a given LEN there is a partition of the set of ordered event pairs (e_i,e_j), $e_i \leq e_j$, associated with the behavior of each individual agent.

For every agent, all the events corresponding to the actions it is involved in are linearily ordered. The entire LEN is thus a superposition of the linear LENs of agents interacting on joint events.

3. EVENT SEMANTICS OF LENs

For efficient use of LEN framework in the knowledge base of an industrial expert one should clearly define the semantics of LENs. As such semantics, with which the expert can solve necessary problems, we can consider the semantics of execution sequences, also known as "interleaving semantics", and the relational event semantics, known as "causal semantics" [9]. The first is suitable for a human, who is supposed to have a sequential way of thinking, and hence, in Case A, it can be used, for example, for matching a particular schedule of actions on the process chart, executed by one of the agents, with the entire chart.

In Case B, such semantics correspond to eliciting a fact of matching between some sequence of the equipment operator's actions and equipment response observations and the poset stereotype of a "fault fabula". This matching can be mimicked as either an identity, thus being a strong form, or as "covering" by a given fault fabula, thus meaning a weaker form, i.e. the top-down similarity.

The second semantics is needed when one cannot distinguish between certain effects using the first one. These effects may pertain to concurrency between events. For example, using the interleaving semantics, one cannot precisely claim HOW the two operations marked with events e_i and e_j are executed if there are at least two such sequences have been observed that in one, "e_i precedes e_j", and in the other, "e_j precedes e_i".

Here, the concurrent execution of both e_i and e_j by the two independent agents is possible, with a possibility of either of them to be the first at finish, and the alternative execution of both e_i and e_j by the same agent ("today, I shall do e_i first and then e_j but, tomorrow, the other way round") may be realized, too.

On the other hand, the interleaving semantics may also be unsuitable in the step-wise refinement of events [10], which one may need to specify in changing the process description level, e.g. the refinement of the "removing a wheel" operation into the component actions - "removing a hubcap", "unscrewing a bolt" etc.

The extraction of the both of these semantics can be done using the efficient graph-based analysis algorithms [11]. For example, for a LEN, one can easily compute the binary relations between events which are defined as follows.

The relation of causal dependence denoted by LI is the transitive closure of $\leq$, while the relation of causal independence, denoted by CO, is given by

$$e_i \text{ CO } e_j = \text{not}(e_i \text{ LI } e_j) \ \& \ \text{not}(e_j \text{ LI } e_i) \ .$$

Using the relational semantics of LENs, the reasoning system can reply to such queries of a user as "if the specified process chart allows for executing the different actions on the same equipment component" , which is a kind of conflict checking paradigm, or "if the continuity between adjacent assembly-disassembly actions on the same component is violated", which presents a kind of interoperational consistency checking.

4. EQUIPMENT STATE SEMANTICS

In addition to the above semantics of LENs, one may also consider semantics defined in terms of the states of components which are involved in a process chart. Such semantics can be extracted by building the reachability tree, starting from the global state of the components which correspond to the set of minimal vertices of the LEN and, hence, is associated with an initial equipment status. Using the interpretation given by labeling μ, this tree allows for the transition from one global state to another global state by firing of a vertex in the LEN and thereby making a corresponding change of the local states of the involved components. Having this semantics, the user may query "if there are any closed non-efficient sequences of actions (loops), after which the equipment, within the single process chart, may return to some, already visited,states." Such checks are very useful in assessing the quality of process schedule documentation and cost evaluation. For example, this gives assurance of correctness of a price list for the various technological services with respect to avoidance of the so-called technological "add-ups". The latter can be utilized, say, by the insurance company which needs appropriate certification of the cost of repair for a damaged equipment.

When checking this semantics with respect to presence of non-efficient operation loops, one can use an alternative technique - the analysis of the dynamic coupledness of the described event structure directly on the LEN, without building a state graph. From the viewpoint of computational complexity, this may yield a crucial speed advantage. In such an analysis, one has to compute the hierarchy of the so-called coupledness relations between components and partition the set of components into coupledness classes [12]. The set of such relations presents a special kind of semantics, the relational semantics of dynamic coupledness of the equipment components.

5. THE PREFIX SYSTEM

To implement the above approach we have developed the prototype architecture of a software package, called PREFIX (from PRocess Events Framework for Industrial eXpert). It consists of the following modules: "LEN Editor", "LEN Librarian", "LEN Manipulator", "LEN Syntax Analyzer", "LEN Semantics Generator", and "LEN Verifier". The latter is loaded together with the knowledge base which supports the semantic analysis of LENs and tailors the system to a given application domain.

The modules have the following basic functions:

"Editor" supports the process of building new and modifying the library LENs, with their graphical representation and interactive inclusion of a textual interpretation of events and agents.

"Librarian" has the capability of a package database and provides necessary facilities for storing and maintaining the LENs.

"Manipulator" serves as an algebraic processor, which implements, on a user demand, the set of operators on LENs, such as "weaving" two given LENs with a possibility of having a non-empty subset of joint events, the projection of a LEN onto a set of specified events (or components), the reduction of a LEN with respect to a given subset of agents, the event refinement (stepping down to the lower abstraction levels), the encapsulation of subsets of events into a single event (steping up to the higher abstraction levels). Under certain conditions, "Manipulator" implements some target functions of the process chart analysis: for example, in Case A, when building a process chart from the set of partial subcharts (in Case B, the composition of a unified "fault history" from separate partial observations) of individual agents, we need to cohere them on joint actions. The checking of coherence is automatically provided by executing the weaving operator between the corresponding LENs, in an incrementally pairwise way.

"Syntax Analyzer" checks the syntactic correctness of the graphical and textual LEN specifications.

"Semantic Generator" extracts the semantics of the above mentioned types, the interleaving semantics, the relational event semantics, the equipment state graph, and the relational semantics of component operation coupledness, from a given LEN.

"Semantic Verifier" assumes an appropriate knowledge base, the rules of semantic analysis, which is formulated (and learned) according to the application domain defined by the user. It responds to the user queries the examples of which have been already presented.

At present, the PREFIX prototype is being implemented using Prolog and Pascal in the MS DOS environment.

6. CONCLUSION

An efficient framework for event-oriented knowledge has been developed. It is based on the labeled event net formalism. Such nets have several fundamental kinds of semantics, within which the user, an industrial expert, can reason about various properties of technological schedules and fault fabulas.

We plan further development of the methods for LEN analysis and reasoning about event knowledge:

(i) towards the capabilities of manipulating uncertain event knowledge structures, and

(ii) towards more creativity of the analysis, which means brining the analysis closer to the process of searching for most adequate process charts or most appropriate "fabulas" of faults.

This being so, one of the important methodological concepts can be the concept of distinction systems and dystinction dynamics [13], which may help to realize real-time reasoning in the PREFIX software.

REFERENCES

[1] Williams, T.P. and Kraft,R., COGSYS: an expert system for process control, Proceedings of the Fifth International Conference on Expert Systems, London, 1989.

[2] Gallanti, M., Guida, G., Spampinato, L. and Stefanini, A., Representing procedural knowledge in expert systems: an application to process control, Proceedings of the IJCAI 9, Los Angeles, 1985.

[3] Gallanti, M., Gilardoni, L., Guida, G. and Stefanini, A., Exploiting physical and design knowledge in the diagnosis of complex industrial systems, Proceedings of the ECAI 7, Brighton, 1986.

[4] Bullock, H., et al., A "bee-hive" model for heterogenious knowledge in expert systems, Proceedings of the ACM 14th Annual Computer Science Conference, Cincinatti, Ohio, 1986.

[5] De Cindio, F., De Michelis, G. and Simone, C., GAMERU: a language for the analysis and design of human communication pragmatics within organizational systems, Lecture Notes in Computer Science, No. 266 (Springer-Verlag, Berlin, 1987).

[6] De Cindio, F., De Michelis, G. and Simone, C., CHAOS as a coordination technology, Proceedings of the Conference on Computer Supported Cooperative Work, Austin, Texas, 1986.

[7] Bandini, S., Carniel, B. and Pomello, L., Narrative context model for representing deep knowledge, Proceedings of the 9th European Meeting on Cybernetics and Systems, Vienna (Kluwer AP, Dordrecht, 1988) Part 2.

[8] Fidelak, M., Petri nets: a formal language for knowledge representation, Proceedings of the ECAI 7, Brighton, 1986.

[9] Rosenblum, L., Yakovlev, A., and Yakovlev, V., A look at the concurrency semantics through lattice glasses, Bulletin of the EATCS, No. 37, 1989.

[10] Castellano, L., De Michelis, G. and Pomello, L., Concurrency versus interleaving: an instructive example, Bulletin of the EATCS, No. 31, 1987.

[11] Kondratyev, A. Yu., Rosenblum, L. Ya. and Yakovlev, A.V., Signal graphs: a model for designing concurrent logic, Proceedings of the International Conference on Parallel Processing, August 1988, (The Pennsylvania State University Press, University Park, 1988) Vol. 1.

[12] Rosenblum, L.Ya. and Yakovlev, A.V., Analyzing semantics of concurrent hardware specifications, in print.

[13] Heylighen, F., Formulating the problem of problem-formulation, Proceedings of the 9th European Meeting on Cybernetics and Systems, Vienna (Kluwer AP, Dordrecht, 1988) Part 2.