TITLE Executable Knowledge

BYLINE Mor Peleg, Department of Management Information Systems, University of Haifa, 31905, Israel, http://mis.hevra.haifa.ac.il/~morpeleg/

SYNONYMS computer-interpretable formalism, knowledge-based systems

DEFINITION Knowledge represented in a symbolic formalism that can be understood by human beings and interpreted and executed by a computer program. Executable knowledge allows a computer program to match case data to the knowledge, reason with the knowledge, select recommended actions that are specific to the case data, and deliver them to users. Executed knowledge can be delivered in the form of advice, alerts, and reminders, and can be used in decision-support or process management.

HISTORICAL BACKGROUND

Representing knowledge in a computer-interpretable format and reasoning with it so as to support humans in decision making started to be developed by the artificial intelligence community in the 1970's. According to Newell [1], knowledge is separate from its representation. At the knowledge level, an agent has as parts bodies of knowledge, actions, and goals. An agent processes its knowledge to determine the actions to take. An agent, behaving through the principal of rationality, selects those actions that attain his goals. Representing knowledge in a symbolic way (i.e., data structures and algorithms) allows the expression of knowledge in a way that would be understandable by humans and would allow people and computer programs to reason with the knowledge and select actions in the service of goals. According to Davis et al. [2], knowledge representation serves five roles: (1) a surrogate to enable an entity to determine the consequences of thinking; (2) a set of ontological commitments about how and what to see in the world; (3) a fragmentary theory of intelligent reasoning; (4) a medium for efficient computation, and (5) a medium for human expression.

Research on knowledge representations started to evolve in the 1970's with rule-based systems. One of the most famous rule-based systems is Mycin [3]. This system represented clinical knowledge about diagnosis and treatment of infectious diseases and allowed clinician users to enter patient findings and, through execution of the if-then-else rules, arrive at probable diagnoses and appropriate treatments. In the 1980's decision-theoretic models, such as Bayesian Networks, influence diagrams, and decision trees [4] started to become popular ways of representing and reasoning with decision knowledge in a probabilistic way, under uncertainty.

By the late 1980's, researchers started to realize the importance of integrating knowledge-based systems with databases. In this way, the case data coming from a database could drive the execution of encoded knowledge. An example of an early formalism that addressed data integration is the Arden Syntax for Medical Logic Modules (MLMs) [5], discussed in detail below.

In the 1990, ontologies [6] have been developed to formalize a shared understanding of a domain. In knowledge engineering, the term ontology is used to mean definitions of concepts in a domain of interest and the relationships among them ("a specification of a conceptualization of a domain" [6]). An ontology enables software applications and humans to share and reuse the knowledge consistently. Ontologies, as represented in a formal language such as frames or description logic, allow logical inference over the set of concepts and relationships to provide decision support and explanation facilities. Ontologies are much more maintainable than rule-based systems. In rule-based systems, the knowledge is represented as individual rules. It is difficult to foresee the affect of addition, deletion, or modification of a rule on the performance of the rule-based system. This problem is solved by ontologies. Research in the 1990's, led by Musen [7], addressed modeling of domain knowledge and problem-solving methods as ontologies that could be combined together, such that a problem solving method could be applied to several domain ontologies, and decision-support systems in a single domain could utilize several generic problem-solving methods.

In the 1990's, a new generation of executable knowledge-based systems started to be developed by two separate communities: the medical informatics community and the business process community. The common theme to the new developments by both of these communities was that the executable knowledge was no longer representing individual decisions, but rather a process that unfolds over time and includes many activities, and the fact that the knowledge-based process system had to be integrated with other systems in the organization. The process modeling languages in both communities were developed as ontologies.

SCIENTIFIC FUNDAMENTAL

Executable knowledge formalisms were developed by several communities, including the artificial intelligence community, the medical informatics community, and the business process management community. The rest of this article reviews work done by the medical informatics and business process communities.

Following the success of rule-based decision-support systems, and at the same time, recognizing the advantages of linking a knowledge-based system to case data, a standard for encoding individual medical decisions, known as the Arden Syntax for Medical Logic Modules (MLMs) [5], was defined in 1989. Arden Syntax was developed initially under the sponsorship of the American Society for Testing and Materials and subsequently of Health Level Seven (HL7). Arden Syntax is a published as an American National Standards Institute (ANSI) standard. MLMs, in Arden Syntax, define decision logic via a knowledge category that has data, evoke, logic, and action slots. The event, logic and action slots specify respectively the events that trigger the MLM (e.g., storage of a new serum potassium test result into a database), the logical criterion that is evaluated (e.g., serum potassium value < 3.5), and the action that is performed if the logical criterion holds, which is often an alert or reminder (e.g., potassium replacement therapy). The data slot specifies mappings between the specific database records and the MLM's variables. However, only part of this specification is defined by the syntax, as it does not contain standard terminology or a data model for electronic medical records.

In the 1990's, decision-support systems whose knowledge is based on evidence-based clinical guidelines started to be developed. Clinical guidelines are recommendations that are developed by healthcare organizations based on evidence from clinical trials, and are aimed at assisting practitioner and patient decisions about appropriate healthcare for specific clinical circumstances. Unlike individual clinical decisions, such as MLMs, clinical guidelines involve multi-step decisions and actions that unfold over time. Guideline-based decision-support systems help clinicians in the process of patient care, including decision making and task management. These systems are based on process-flow ontologies termed Task-Network Models (TNM) [8] – a hierarchical decomposition of guidelines into networks of component tasks that unfold over time. The task types vary in different TNMs, yet all of them support modeling of medical actions, decisions, and nested tasks. These models contain computer-interpretable specifications of decision criteria and clinical actions that enable an execution engine to interpret the guideline representation and execute it for a given patient case data. Important themes in executable guideline models include the ability of some of them to integrate with electronic databases (e.g., electronic health record systems (EHR)), using standard terminologies to express medical actions (e.g., laboratory tests, drug prescriptions) and patient data items upon which decision criteria are written, using standard expression languages for writing decision criteria, and using messaging standards for exchange of (clinical) data.

In 2003, researchers from six groups that developed TNMs participated in a study that compared their models using two clinical guidelines that served as case studies [8]. The TNMs studied were Asbru, EON, GLIF, Guide, PRODIGY, and PRO*forma*. An example of a guideline model represented in GLIF is shown in Figure 1. Although these formalisms all depict a guideline as a TNM, they each have their own emphasis. Asbru emphasizes specifying intentions as temporal patterns; EON views the guideline model as the core of an extensible set of models, such as a model for performing temporal abstractions. EON uses a task-based approach to define decision-support services that can be implemented using alternative techniques; GLIF emphasizes the ability to share and integrate guideline specifications among software tools and implementing institutions; GUIDE focuses on integration with organizational workflow using a workflow-based model and linkage to decision-theoretic models, PRODIGY (a project that is no longer active) aimed at producing the simplest, most readily comprehensible model necessary to represent chronic disease management guidelines. It models guidelines as decision maps organized as a collection of clinical contexts; in each context, selection among relevant clinical actions is made; PRO*forma* advocates the support of safe guideline-based decision support and patient management by combining logic programming and object-oriented modeling. Its syntax and semantics are formally defined. One aim of the PRO*forma* project is to explore the expressiveness of a deliberately minimal set of modeling constructs: actions, compound plans, decisions, and inquiries of patient data from a user.

The study compared the TNM models in term of eight components that capture the structure of computerized guidelines: (1) organization of guideline plans, (2) goals, (3) model of guideline actions, (4) decision model, (5) expression language, (6) data interpretation/abstractions, (7) medical concept model, and (8) patient information model.

The purpose of the study was to find consensus among the different formalisms that could be a starting point for creating a standard computerized guideline formalism. Differences between the guideline modeling languages were most apparent in underlying decision models (ranging from simple switching constructs to argumentation rules for and against decision alternatives, and even use of decision-theoretic models such as influence diagrams and decision trees), goal representation, use of scenarios (as a plan component that defines a particular patient management context that serves as entry points into guidelines), and structured medical actions which could be mapped into controlled vocabulary terms.

Consensus was found in plan organization (plans could be nested and structured as plan components arranged in sequence, in parallel, and in iterative and cyclic structures), expression language for specifying and sharing decision and eligibility criteria, patient state definitions, and preconditions on system actions, conceptual medical record model, medical concept model, and data abstractions (i.e., definitions of abstract terms using mathematical functions of other concepts, temporal abstractions, and concept hierarchies that allow reasoning at different levels of the hierarchy).

The HL7 Clinical Decision Support Technical Committee (CDSTC) has focused on standardization of two of the components for which consensus was found: expression language and conceptual virtual medical record (vMR) model. The object-oriented guideline expression language, GELLO [9], is an extensible guideline expression language that can be used for formally defining decision and eligibility criteria, as well as patient states. It is based on the Object Constraint Language (OCL) (http://www-306.ibm.com/software/rational/uml/resources/documentation.html). In 2004, it was established as a standard of HL7. The CDSTC started the process of standardizing a vMR, based on experiences with the patient information models of PRODIGY, EON and the HL7 RIM, which is also the basis of GLIF's default patient information model. An object oriented vMR would ease the process of mapping guideline patient data items to real EMRs, allowing decision criteria, eligibility criteria and patient states to be defined by in guideline models by reference to the VMR rather than specific EMRs.

The second community that started to develop a new generation of executable knowledge-based systems is the business process management community. Workflow management systems started to be developed based on workflow formalisms that define a model of business processes and a model of the organization, including its individual actors, roles, organizational units, and resources that participate in workflow activities. Several industrial groups defined standards for workflows. They Web Services Business Process Execution Language (BPEL) by the Advanced Open Standards for the Information Society (OASIS), is an executable formalism where the business process behavior is based on Web Services. XML Process Definition Language (XPDL) is a standard that is under development since 1998 by the Workflow Management Coalition - An industry group dedicated to creating software standards for workflow applications. The goal of XPDL is to store and exchange the process diagrams among different

Comment [MSOffice1]: Cross-reference to definition on p. 9

workflow tools, including workflow modeling tools (editors) and workflow engines. In In 2004 the Workflow Management coalition endorsed the graphical workflow standard called Business Process Modeling Notation (BPMN) developed by Object Management Group, to standardize the way that process definitions are visualized.

The business process management community has developed many tools that enable modeling, analyzing, verifying, simulating, and executing a business process. While the medical informatics community devoted much research to supporting clinical decision making in addition to patient care task management, the business process community made much progress on modeling task management within an organization, where the business process often involves many departmental units and organizational roles. Workflow systems aim to help in resource management, a task that is not addressed by current executable guideline systems.

KEY APPLICATIONS

A generic free and open-source tool for creating ontologies, or knowledge-bases, is Protégé (protégé.stanford.edu). The Protégé platform supports two main ways of modeling ontologies via the frames and Ontology Web Language (OWL) formalisms. Protégé is implemented in the Java programming language and is extensible. Protégé-frames knowledge bases could be reasoned with several rule-engines, such as Jess and Algernon. Additionally, the Protégé Axiom Language could be used to define first-order-logic constraints and check them for instances in the knowledge base. Several reasoners can be used with Protégé-OWL to classify instances or classes according to class definitions and class hierarchies. They include the commercial product Racer, and the commonly used free reasoners Pellet and Fact++. The SWRL rule-based engine could also be used to reason with Protege-OWL.

The Arden Syntax for Medical Logic Modules has been applied by universities such as Columbia University Medical School and by several health information systems vendors, such as Micromedex, Eclypsis Corporation, McKesson Information Solutions, and Siemens Medical Solutions Health Services Corporation to create MLMs that deliver clinical recommendations in the form of alerts and reminders. MLMs are being used by many clinical institutions in the United States (see http://cslxinfmtcs.csmc.edu/hl7/arden/ for a partial list). The Department of BioMedical Informatics at Columbia University developed a knowledge-base of over 240 MLMs that are now available from http://cslxinfmtcs.csmc.edu/hl7/arden/.

Many tools exist to support guideline development, modeling, verification, and execution [10, 11]. Asbru modeling tools include Delt/A (http://ieg.ifs.tuwien.ac.at/projects/delta/) and URUZ, both focusing on easing the transition from narrative to formal representations via a mark-up stage, AsbruView (http://www.ifs.tuwien.ac.at/asgaard/asbru/tools.html), which focuses on visualization and user interface for authoring, and CareVis (http://ieg.ifs.tuwien.ac.at/projects/carevis/), which provides multiple simultaneous views to cover different aspects of a complex underlying data structure of treatment plans and patient data. Verification of Asbru guidelines can be done using formal verification methods [12]. iimplementations in Asbru were developed for diabetes, artificial ventilation, and breast cancer guidelines.

EON guidelines can be authored in the knowledge-modeling tool Protégé-2000 (protégé.stanford.edu) and executed by an execution engine that uses a temporal data mediator to support queries involving temporal abstractions and temporal relationships. A third component provides explanation services for other components. EON has been used to create hypertension and opiods guidelines that are implemented in various hospitals and clinics of the Veteran Affairs Hospital. Protégé-2000 is also the modeling tool for GLIF guidelines. The GLIF execution engine (GLEE) has been used to implement two guidelines: diabetic foot diagnosis and management and flu vaccinations.

Guide has a new implementation of an authoring tool and an execution engine called NewGuide. Guide has been used to implement guidelines for stroke and for management of patients with heart failure.

A number of software tools have been created for creating, visualizing, and executing PRO*forma* guidelines. They include Arezzo and Tallis. Several PRO*forma* guidelines have been implemented and some have undergone clinical trials to establish their safety and utility. More information on PRO*forma* implementations as well as on implementations of other guideline formalisms can be found at the openclinical web site (www.openclinical.org) – a repository of resources about decision support, clinical workflow and other advanced knowledge management technologies for patient care and clinical research.

Key application for workflow management systems include tools for modeling and executing workflows, such as the open-source workflow tool Bonita (bonita.objectweb.org), FLOWer (http://www.workflowdownload.com/workflow/flower.html), YAWL (http://www.yawl-system.com/), and Together Workflow Editor and server (http://www.together.at/together/prod/twe/), and Oracle Workflow (http://www.oracle.com/technology/products/integration/workflow/workflow_fov.html). Other tools exist for verifying [13] workflows (Oracle Worklow). There are business process management engines available from several vendors including Software AG, Savvion, Lombardi, Appian, JBoss, and Tibco.

FUTURE DIRECTIONS

The challenge of creating executable knowledge that can be shared by multiple implementing institutions and mapped to their information systems started to be addressed by projects such as the Arden Syntax, GLIF, Shareable Active Guideline Environment (SAGE) [14] and Knowledge-Data Ontology Mapper [15], yet more work needs to be done in this area to define how knowledge can be authored in a way that is institution-specific and sharable.

One of the most interesting future directions of executable knowledge concerns synergetic development that draws upon developments made in the medical informatics and the business process communities. Such collaborations are emerging, as manifested by health-care related workshops that are taking place in business-process management and information systems conferences, such as the ProHealth Workshop, which is part of the Business Process Management conference.

CROSS REFERENCES Knowledge representation, rule-based systems, ontologies, Bayesian Networks, Influence Diagrams

RECOMMENDED READING

1. Newell A. The knowledge level. Presidential address. American Association for Artificial Intelligence, AAAI80; 1980; Stanford University; 1980.
2. Davis R, Shrobe H, Szolovits P. What is a Knowledge Representation? AI Magazine 1993;14(1):17-33.
3. Shortliffe EH. Computer-based medical consultations: Mycin. New York: Elsevier/North Holland; 1976.
4. Pearl J. Probabilistic reasoning in intelligent systems: Morgan Kaufmann; 1988.
5. Hripcsak G, Ludemann P, Pryor TA, Wigertz OB, Clayton PD. Rationale for the Arden Syntax. Comput Biomed Res 1994;27(4):291-324.
6. Gruber TR. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Int. Journal of Human-Computer Studies 1995;43(907-928).
7. Musen MA, Tu SW, Eriksson H, Gennari JH, Puerta AR. PROTEGE-II: An Environment for Reusable Problem-Solving Methods and Domain Ontologies. International Joint Conference on Artificial Intelligence; 1993; Chambery, Savoie, France; 1993.
8. Peleg M, Tu SW, Bury J, Ciccarese P, Fox J, Greenes RA, et al. Comparing Computer-Interpretable Guideline Models: A Case-Study Approach. J Am Med Inform Assoc 2003;10(1):52-68.
9. Sordo M, Ogunyemi O, Boxwala AA, Greenes RA, Tu S. GELLO: A Common Expression Language. 2004. http://cslxinfmtcs.csmc.edu/hl7/arden/2004-09-ATL/v3ballot_gello_aug2004.zip
10. Peleg M. Guideline and Workflow Models. Greenes RA, editor. Clinical Decision Support - The Road Ahead: Elsevier/Academic Press; 2006.
11. de Clercq PA, Blom JA, Korsten HHM, Hasman A. Approaches for creating computer-interpretable guidelines that facilitate decision support. Artificial Intelligence in Medicine 2004;31:1-27.
12. Teije At, Marcos M, Balser M, Croonenborg Jv, Duelli C, Harmelen Fv, et al. Improving medical protocols by formal methods. Artificial Intelligence in Medicine 2006;36(3):193-209.
13. van der Aalst WMP. The application of Petri Nets to Workflow Management. The Journal of Circuits, Systems and Computers 1998;8(1):21-66.
14. Tu SW, Campbell JR, Glasgow J, Nyman MA, McClure R, J McClay J PC, Hrabak KM, Berg D, Weida T, Mansfield JG, Musen MA, Abarbanel RM. The SAGE Guideline Model: achievements and overview. J Am Med Inform Assoc 2007;14(5):589-98.
15. Peleg M, Keren S, Denekamp Y. Mapping Computerized Clinical Guidelines to Electronic Medical Records: Knowledge-Data Ontological Mapper (KDOM). Journal of Biomedical Informatics 2008;41(1):180-201.

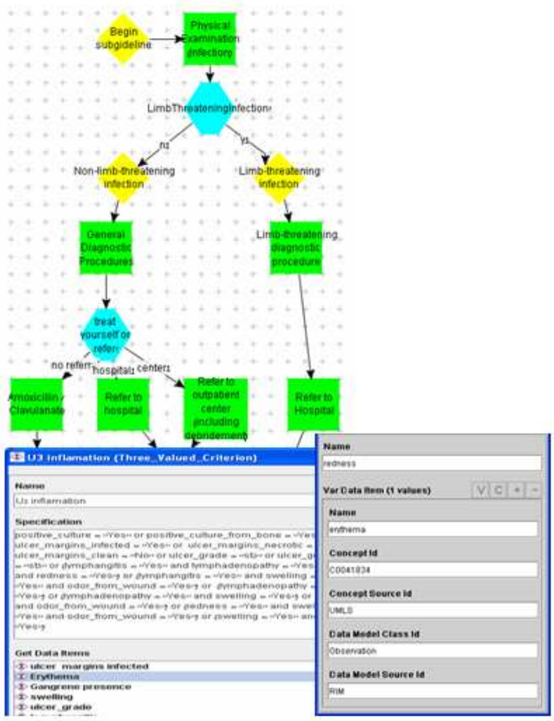

Figure 1. Part of a GLIF model for a Diabetic Foot guideline, modeled using the Protégé-200 tool. Squares denote actions, diamonds – patient state steps, hexagons – decisions. The bottom part of the figure shows the computable specification of the decision step "Limb-threatening Infection?" The decision criterion is based on values of patient data items. Each patient data item is defined using a concept code taken from a controlled medical vocabulary (Unified Medical Language System (UMLS) in this case) and by a patient data model class (e.g., Observation) taken from a data model source, such as Health Level 7's Reference Information Model (RIM).

TITLE Workflow management system

BYLINE Mor Peleg, Department of Management Information Systems, University of Haifa, 31905, Israel, http://mis.hevra.haifa.ac.il/~morpeleg/

SYNONYMS Workflow management system, business-process management systems

DEFINITION According to the Workflow Management Coalition, a workflow management system is a system that completely defines, manages, and executes workflows through the execution of software whose order of execution is driven by a computer representation of the workflow logic

MAIN TEXT A workflow model includes two parts: a model of the workflow process, including the activities, their order, and their start and termination conditions, and a model of the organization at which the workflow takes place, including the individual actors, the organizational roles, and departmental units who all perform workflow activities. A workflow process is performed for cases that trigger it (e.g., item order, a candidate's application). A workflow task performed for a specific case is called "workflow item". Workflow items are performed by a participant: an organizational actor, role, unit, or resource. An activity is a work item performed by a participant. As stated by Wil van der Aalst, "Workflow management is the glue between the cases, the tasks, and the organization"

CROSS-REFERENCES Executable Knowledge

REFERENCES

van der Aalst WMP. “The Application of Petri Nets to Workflow Management”, Journal of Circuits, Systems and Computers, vol. 8(1), 21-66, 1998