MAKO-PM: Just-In-Time Process Model

Riki Y. Morikawa and Larry Kerschberg

E-Center for E-Business, Department of Information and Software Engineering,
George Mason University, MSN 4A4, Fairfax, Virginia, 22030-4444
rikimorikawa@yahoo.com, kersch@gmu.edu, http://eceb.gmu.edu/

Abstract. Unlike traditional assembly lines, service-oriented processes require individualized and timely information during the execution of steps or phases. This paper discusses the Multi-layered Analytical Knowledge Organization (MAKO) Just-in-Time Process Model (JITPM), which is a framework and methodology for building service-oriented processes that are interrelated to an organizational knowledge base. Process model instances are connected to a temporal reference, which ensures the timely retrieval of pertinent information, and compliance with procedural constraints

1 Introduction

Today's service-oriented economy requires providers to be flexible in catering to the specific needs of individuals, while still following a predetermined process that guarantees quality and efficiency for both the organization and customer. Unlike traditional assembly line processes, which are typically predictable and unchanging, many service-oriented processes require that a level of customization be made possible through the timely identification of current pertinent information. In the medical field, for example, patients are assigned specific standardized protocols based upon a physical ailment. Depending upon the patient's reaction to the assigned protocol, the attending physician may make modifications, or may decide to terminate treatment in favor of an alternative protocol. In order to enable the making of real-time process decisions, the decision maker must have access to the latest data regarding the specific instance of the process, as well as literature regarding state-of-the-art practices within the field [1, 2].

Previous solutions have addressed knowledge and information requirements using "pull" type scenarios based upon user queries [3, 4]. Other solutions include the introduction of relevant information during specific phases of a process [5], or through the creation of a temporal ontology used to track procedural steps through a protocol [6]. The uniqueness of our approach consists of the adoption of a standards-based knowledge framework, the creation of a simple temporal ontology, and the use of roles to identify process phases or steps.

This paper describes the Multi-layered Analytical Knowledge Organization (MAKO) Just-In-Time Process Model (JITPM) [7], which enables the service worker to follow established processes and to retrieve the latest and most relevant information available in order to make timely knowledgeable decisions. The JITPM provides the worker with the knowledge and information deemed necessary to perform a single phase, or step, of an overall procedure. By doing this, information overload to the worker is minimized. Based upon the XML Topic Map Standard (XTM) [8], the JITPM model ensures interoperability with other systems.

2 Multi-layered Analytical Knowledge Organization (MAKO) Framework

MAKO represents a framework methodology for building, maintaining, modifying, and operating a knowledge base system. The strength of the MAKO concept resides in the fundamental structure of the framework. Any number of customized ontologies can be developed to address the conceptualizations that are most pertinent to an organization.

The MAKO framework, shown in Table 1, is a cross matrix of several major concepts. The Multidimensional Ontology Model (MOM) decomposes the knowledge base into separate conceptualizations based upon common sense groupings. The Temporal Layer Model (TLM) provides a reference by which topics can be segregated according to their time points and intervals. The Interpretive Layer[1] (IL) represents non-validated assessments or opinions that are available for sharing, critiquing, and analysis amongst members of the organization. Although the IL represents information considered to be conjecture or hypothetical, it is vital for the creation of knowledge. Until fully validated by the organization, the new potential knowledge represented in the IL is segregated from the validated part of the MAKO knowledge base.

Table 1. MAKO Framework consists of the MOM, TLM and IL.

		Multidimensional Ontology Model		
		Ontology A	**Ontology B**	**Processes**
	INTERPRETIVE LAYER	Interpreted Knowledge		
Temporal Layer Model	**OCCURRENT LAYER**	DYNAMIC Subjects	DYNAMIC Subjects	DYNAMIC Subjects
	CONTINUANT LAYER	STABLE Subjects	STABLE Subjects	STABLE Subjects

2.1 Multidimensional Ontology Model (MOM)

---

[1] The IL model is not discussed in this paper. However, its construction is similar to other ontologies within the MOM except for unique scope and class elements that distinguish it from the validated parts contained in the knowledge base.

An ontology is defined as a "specification for a conceptualization". The conceptualization is comprised of a set of concepts, or classes, that relate to one another in some logical fashion. In essence, an ontology describes concepts that are in a domain of discourse. We use the term Semantic Layer to refer to an ontology that has been populated with class instances. The Semantic Layer, combined with rules and constraints, constitute the knowledge base [9, 10].

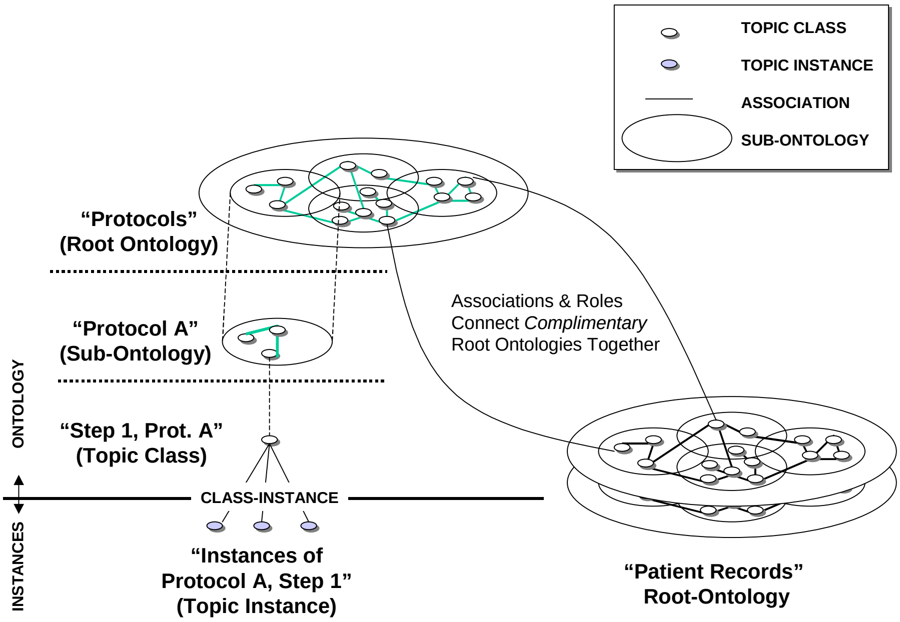

Fig. 1. The Medical Protocol ontology is decomposed into separate sub-ontologies. Multiple instances of each step, or phase, within a specific protocol are created as patients are assigned. The "Protocols" root ontology is connected to the "Patient Records" root ontology via class associations, and they are considered complementary to one another.

In terms of the Multidimensional Ontology Model, a set of *root* ontologies are integrated through standard XTM 1.0 association element interfaces. A root ontology represents a specific conceptualization. Within the MOM, several root ontologies co-exist in a complementary manner within the knowledge base. There are several qualities that a root ontology possesses in our model. The first quality is *independence*, which means that the ontology can represent a complete domain of knowledge without depending upon other domains (e.g., temporal, patient records, pharmaceutical domains). The second is *reusability*, where identified reference root ontologies can be reused by other knowledge base applications with minimal, if any, modification. While reusability is required for all reference ontologies (e.g. temporal, almanacs, medical dictionaries, etc.), it is not a necessary condition for other ontologies where highly specialized domains are uniquely specified per organization (e.g., medical treatment protocols may be unique to each medical center). Third, root ontologies must have *standardized interfaces* that enable them to interconnect to other root ontologies. This is accomplished by defining a set of standard association class elements such as “related_to”, “begins_on”, “ends_on”, etc. Finally, root ontologies can be further decomposed into multiple *sub-ontologies* when necessary. As an example, the root ontology containing all medical protocols is decomposed into *sub-ontologies* that represent separate steps within a single protocol. The steps are then decomposed further into separate topics that are pertinent to the execution of the single step. This decomposition process serves to refine conceptualizations into simpler and more manageable representations, and continues until no further simplification is deemed necessary (Figure 1).

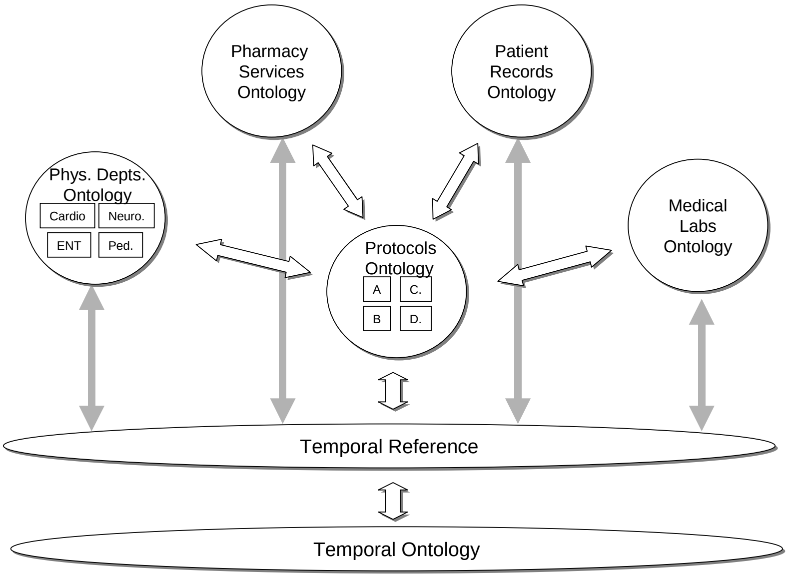

Fig. 2. Multidimensional Ontology Model for medical protocol example. Each ontology is interconnected to other relevant ontologies through the class association elements (depicted as double-headed arrows in figure). Each ontology is also connected to the Temporal Reference. The Temporal Layer Model (TLM) is defined as the combination of the Temporal Reference and Temporal Ontology.

Complementary ontologies are defined as two or more root ontologies that are interrelated to one another, thereby serving to enhance the overall knowledge represented. Complementary ontologies are connected through a subset of ontology defining *association* and *role* class elements. In the case of Figure 1, ontology defining classes within the “Protocols” and “Patient Records” ontologies are interconnected to one another (e.g., patient classes are assigned to particular protocol classes). Through inheritance, any instance of a patient assigned to a protocol, retains this interrelationship. The desired result is to provide relevant patient information to the medical professional when a specific protocol has been assigned (e.g., name, billing information, medical history, etc.). Doing this helps to provide a more complete picture of the patient/protocol instance.

An ontology within the MAKO-MOM is constructed using a set of topic, association and occurrence elements as defined by XTM 1.0. These interconnected sets of elements define the classes, which in turn, define the conceptualization being modeled.

In our example of the medical center, ontologies include “Pharmacy Services”, “Patient Records”, “Medical Labs”, “Physician Departments”, and “Protocols” (see Figure 2). Each ontology is considered independent (i.e., each is a complete specification of its conceptualization), and complementary to other ontologies. For example, protocol “A” in Figure 2 represents a complete step-by-step process that interfaces with other ontologies such as “Patient Records”, “Medical Labs”, “Phys. Depts.”, and “Pharmacy Services”, but is not dependent upon them conceptually[2]. When a physician assigns a patient to an instance of a protocol, association instances are automatically created between complementary ontologies

2.2 Temporal Layer Model (TLM)

The Temporal Layer Model consists of the Temporal Ontology and the Temporal Reference. The Temporal Ontology consists of a set of class elements that define temporal concepts such as year, month, day, season, etc., and the rules, constraints, and ordering relationships by which they are governed. The Temporal Reference consists of Temporal Ontology instances that are ordered via predecessor/successor relationships (e.g., the instance of year “2001” precedes the instance of year “2002”). The ordering relationships established by both the Temporal Ontology and Temporal Reference allows serialization (i.e., a partial ordering) of all interrelated knowledge base elements. Figure 3 depicts a simplified portion of the Temporal Ontology. Class association elements relate topic classes together in pre-defined relationships. Each topic or association class also indicates the type of data allowed, and the constraints or rules by which instances can occur.

When an instance of “year” is created, topics for “months”, “weeks”, “days”, “hours”, “minutes”, etc., which comprise the concept of “year” are also created. When declaring an instance of “year”, constraints and ordering relationships are inherited in a straightforward manner. For example, every year has 12 months starting with “JAN” and ending with “DEC”, and each month has a predetermined number of days. However, when assigning a day of the week (“SUN” through “SAT”) to a particular date, simple algorithms need to be called upon to ensure the correct matching between “date” and “day-of-the-week”. This is because assignment is dependent upon the particular year in question. Leap year adjustments are also handled in the same manner. Figure 4 depicts an example of two dates: December 31, 2003 and January 1, 2004. The Temporal Reference provides the following information:

1. The year “2003” precedes the year “2004”
2. The month “DEC” has 31 days, and is the last month of the year.

---

[2] Protocol “A” is described as a set of steps. Some of these steps may refer to lab results or patient reactions; however, the existence of the protocol itself is not dependent upon either the medical lab or the patient ontologies.

3. The month “JAN” has 31 days and is the first month in the year.
4. December 31, 2003 occurred on a Wednesday. Therefore, the *variable name* for December 31, 2003 is “Wednesday”.
5. January 1, 2004 occurred in a Thursday. Therefore, the *variable name* for January 1, 2004 is “Thursday”.
6. December 31, 2003 preceded January 1, 2004.
7. Following constraints require “WED” to immediately precede “THR”, “DEC 31” to precede “JAN 1”, etc.

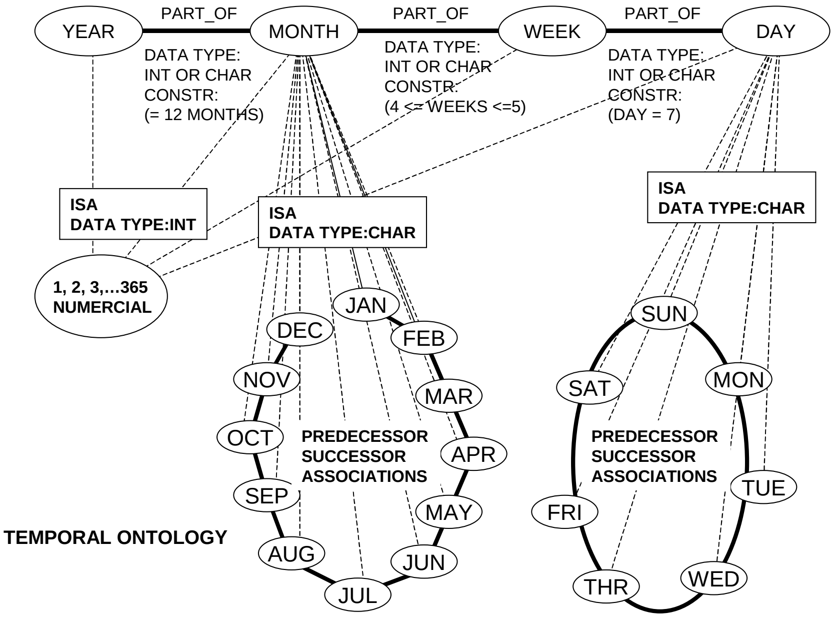

Fig. 3. The Temporal Ontology provides class relationships within the TLM. “ISA”, “Part_of”, “super-subclass”, “aggregate”, and ”predecessor-successor” relationships are established within this ontology. Figure shows a partial example of the type of relationships and constraints that exist. Instances of ontology elements are constrained by data type and upper/lower bounds. For example, “JAN” must have 31 days, year “2001” must have 12 months (not shown).

All ontology-defining class elements and their instances are connected to the temporal reference for the purpose of object serialization and temporal inferencing. Each root or sub-ontology is temporally divided into *occurrent* and *continuant* layers, which serve to maintain order among knowledge base objects. Continuant objects are *stable* topics and associations that do not change over the valid time interval[3] of the knowledge base. Occurrent, or *dynamic*, objects are created, destroyed, or their definitions modified during the valid time interval [11]. Segregating continuant and occurrent objects has benefits when considering knowledge base serialization. The

---

[3] The valid time interval of the knowledge base represents the time interval for which the knowledge base was intended.

reason is simple, continuant objects, which exist in a stable form regardless of time, are not required to be temporally ordered as long as their core identity (i.e., subject identity, base name, and class) does not change. This reduces computer resource needs during temporal serialization. For knowledge bases that are populated with numerous occurrent elements whose associative "mappings" change frequently, the temporal reference helps to preserve the historical relationships between objects thus making post analysis for any previous time period possible.

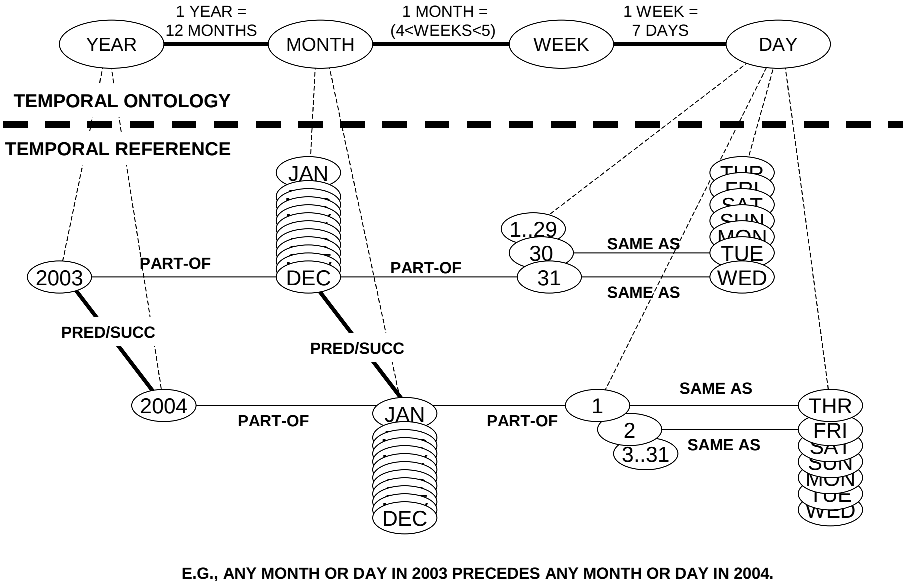

Fig. 4. TLM example for dates December 31, 2003 and January 1, 2004.

3 MAKO Just-In-Time Process Model (JITPM)

The JITPM is captured within the MAKO framework as an ontology consisting of a number of separate or integrated processes. Each process is represented by a set of topics, which in turn, represent steps or phases. The relationships between steps (e.g., "predecessor-successor", "overlapping steps", "gap between steps") are represented by XTM 1.0 association class elements and member roles. Processes are connected to complementary ontologies and the temporal reference via association elements. By connecting process steps and complementary ontologies to the temporal reference, time critical information (e.g., patient test results, latest drug information or treatment techniques, etc.), is readily available to the medical professional during specific steps or phases when needed. In addition, since the start of any protocol is relative to the patient (i.e., patients on the same protocol will most likely have differing start-finish times), the temporal reference helps the system automatically keep track of patient progress, and can alert medical professionals when a protocol decision point has been reached.

Each phase within the process is associated with a time interval whose beginning and end time points are represented by t_{open} (i.e., start of the phase) and t_{close} (end of the phase). The relationships between phases (e.g., “gap”, “overlap”, “pred/succ”) are defined by the roles each phase, or topic, plays in the adjacent association. These role types are shown in Figure 5. For example, if two phases are connected to one another in a “pred/succ” relationship, then the predecessor phase takes on the role of SB/EB or “Starts_Before/Ends_Before”, and the successor phase takes on the role of SA/EA or “Starts_After/Ends_After”. The role assignments in this relationship means that the predecessor phase must start and end before the successor phase begins. From the viewpoint of the successor phase, it cannot start until the predecessor phase has ended. In addition, our model identifies the “pred/succ” relationship as one in which the successor must start immediately after the predecessor has ended. In contrast, the definition of the “gap” relationship with phases that are assigned similar roles, allows a period where neither the predecessor nor the successor are active. Roles dictate “overlap” relationships between phases in a similar manner. In each “overlap” case, the unique relationship between phases is strictly dictated by the roles each plays. Constraints, which are part of the protocol’s ontological description, dictate the minimum and maximum times a phase overlap or gap can exist. The concept of applying roles to relationships provides flexibility in representing complex processes that do not adhere to strict sequential order (e.g., several parallel phase paths executed in a concurrent manner).

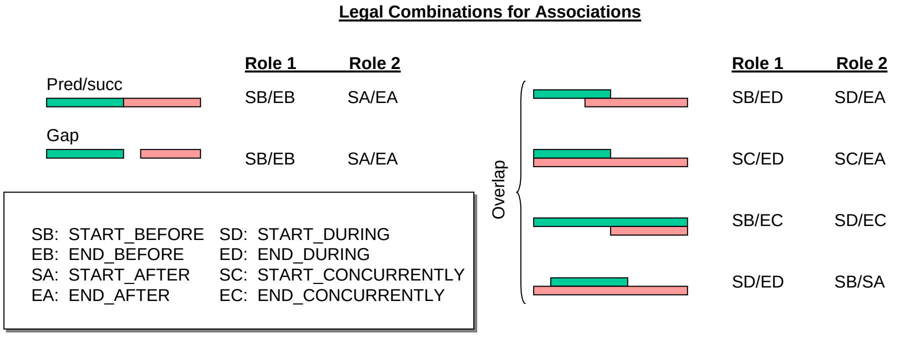

Fig. 5. Protocol process is shown with relative time line for each phase. Each protocol is created as a class of topic and association elements that are linked to relevant ontologies as appropriate. Upon initiation, an instance of the protocol is created with full inheritance, and the first phase is anchored via the temporal reference.

Figure 6 depicts a fictional treatment protocol. Relationships between phases are defined by their roles per Figure 5 and are defined ahead of time in the protocol ontology. The protocol is a set of topics related together via association and role elements. Each protocol within the ontology is carefully created by the organization based upon current medical practices for use in treating a specific ailment. Each step in the protocol is represented by either a single topic or set of topics, which in turn represent an activity such as retrieving patient information or lab results, linking and retrieving relevant medical documentation, executing or terminating a process such as the start of a drug regimen, and identifying decision points within the procedure. Each protocol instance inherits data typing and constraints from parental classes contained within the protocol ontology. Protocol instances, which are essentially copies, i.e., *instantiations*, of the protocol ontology, are uniquely created for individual patients and are attached to the temporal reference. Attaching the individual patient's protocol instance to the temporal reference essentially anchors the start date, enabling the system to automatically track patient progress. Each instance of a protocol also inherits parental associations that are connected to other relevant ontologies such as the "Medical Lab", "RX Services", "Patient Records", and "Phys. Dept.". In other words, as soon as a patient is assigned to a specific protocol, associations between the protocol and other relevant ontologies are created. For example, upon protocol initiation, the topic "retrieve_patient_history" tells the system to automatically retrieve the patient's medical record from the "Patient Records" database. As the patient progresses through the protocol, timely information, such as lab results, availability of new drugs, or new procedures, are revealed to the physician at the appropriate time when informed decision-making or observation is required.

As an example, in Figure 7, patient "X" is assigned to protocol "A" by the physician. Upon assignment, an instance of protocol "A" is created for "X".

At "p1" of the protocol instance "A", information is automatically linked to patient "X's" pertinent drug, medical lab and treatment histories. Only data that is specific to protocol "A" is returned (e.g., patient "X" billing records are not needed, however "X's" medical history is considered relevant and therefore retrieved[4]). By initiating an instance of protocol "A", relevant knowledge is automatically pushed to the physician as each phase, "p1" through "p5", is executed. Initiation of "p1" also causes a time stamp, t_{open}, to be created via relationship to the Temporal Reference. This time stamp provides the reference by which the remainder of the protocol is executed.

---

[4] This applies to drug, medical and physician department information as well. Only patient information deemed necessary for executing the protocol are returned.

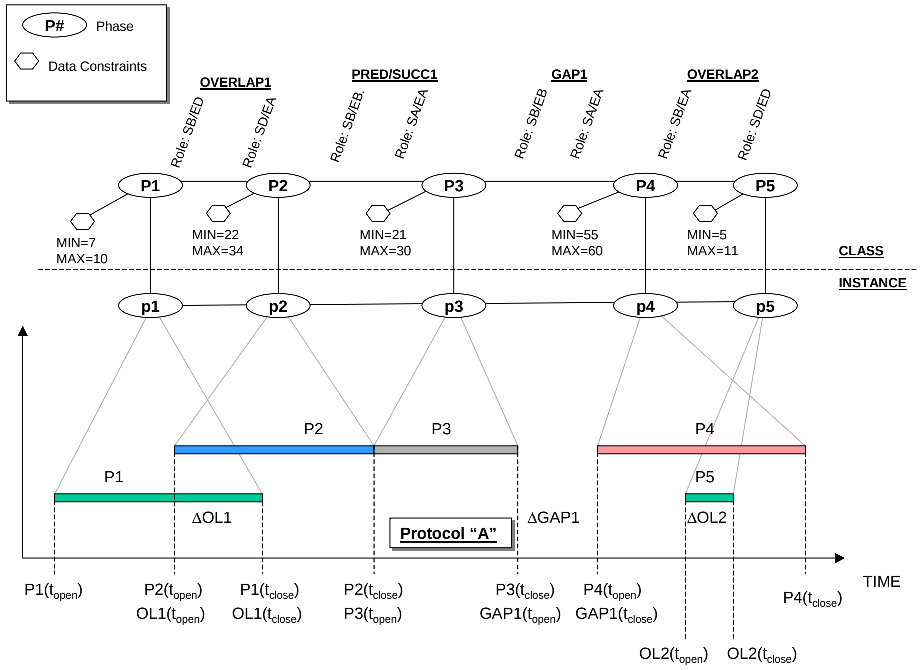

Fig. 6. Protocol "A" is created within the Protocols Ontology as a set of topic and association class elements. Data constraints, such as the minimum and maximum number of days allowed for each phase, are identified as part of the protocol "A" ontology. Upon assignment to a patient, an instance of protocol "A" is created and the start time P1(topen) is anchored to the temporal reference. Bars on the time graph represent the length of each phase, the order in which each phase must occur, phase time limits, and the time in which "gaps" or "overlaps" between phases can occur.

Finally, the physician is alerted to information (e.g., current studies, research material, medical findings) contained in databases and web sites that are considered relevant to protocol "A", "p1" (Figure 7).

An advantage of modeling protocols using this method is the ability to capture changes within the ontology that can be applied to all future instances of the protocol. This makes the implementation of procedural changes (i.e., between roles or the sequencing of phases, and data constraints) straightforward. Other advantages include the ability to track multiple patients with different start times, mid protocol re-direction per patient, and the system's ability "push" relevant information to the medical professional based upon patient progress and physician decision making.

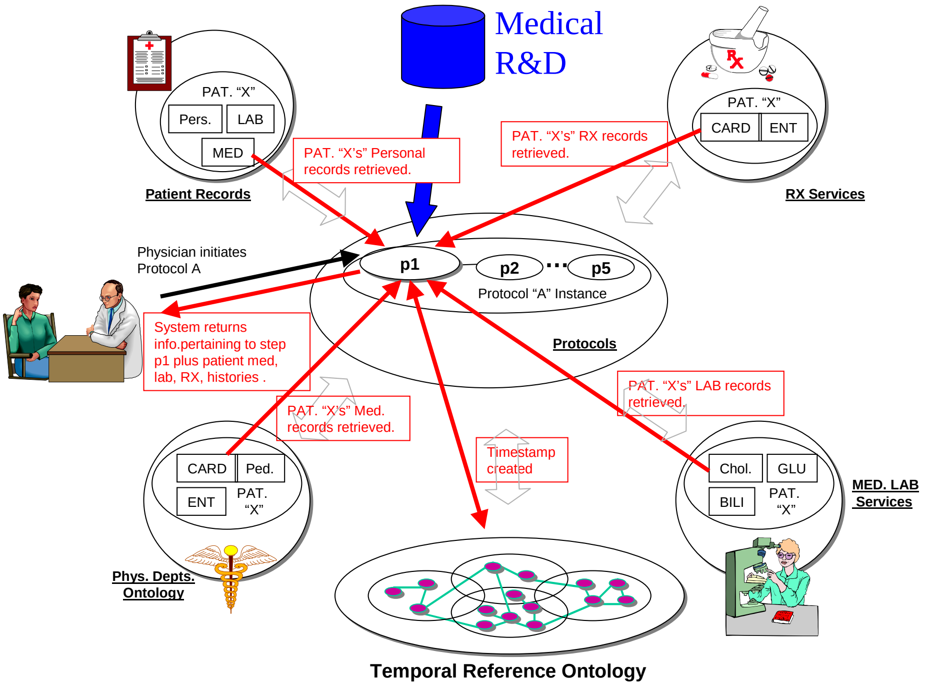

Fig. 7. Protocol is represented by a set of connected topic and association elements. During the execution of each phase, the most recent and pertinent medical treatment data is also pushed to the physician for review.

4 Conclusions

The MAKO-PM model enables timely information to be displayed to a user when specific steps or phases in a process are reached. The information retrieved is derived from separate, and independent, ontologies maintained by subject matter experts for use by an organization. By creating associations between individual steps in a process and related data in complimentary ontologies, the user is supplied with the critical and timely data needed to make informed decisions.

The MAKO-PM uses the XTM standard as a basic building block for maximum interoperability. However, the basic concept should be equally viable using RDF and OWL standards. A deficiency of the current XTM standard is it's lack of a data typing or data constraint mechanism. Future investigation should include ways in which RDFS-like templates can be formally adopted to address such deficiencies.

Acknowledgements. This work was sponsored by a NURI from the National Geospatial-Intelligence Agency (NGA).

References

[1] Davenport, T.H. and Glaser, J. Just-in-Time Delivery Comes to Knowledge Management. in Harvard Business Review, Best Practice (R0207H), July 2002.

[2] Flower, J. Beyond the Digital Divide. in Health Forum Journal, Winter 2003.

[3] Fagrell, H., Forsberg, K., Sanneblad, J. FieldWise: A Mobile Knowledge Management Architecture. in CSCW'00, ACM Press, 2000.

[4] Hauk, R.V. and Chen, H., COPLINK: A Case of Intelligent Analysis and Knowledge Management. in 20th International Conference on Information Systems, Charlotte, North Carolina, ACM Press, 15-28, 1999.

[5] Golebiowska, J., Dieng-Kuntz, R., Corby, O. and Mousseau, D. Building and Exploiting Ontologies for an Automobile Project Memory. in International Conference on Knowledge Capture, Victoria, British Columbia, Canada, ACM Press, 52-59, 2001.

[6] Weng, Chunhua, M. Kahn, and J. Gennari, Temporal Knowledge Representation for Scheduling Tasks. in Clinical Trial Protocols, http://faculty.washington.edu/gennari/papers/AMIA02-final.pdf, 2002.

[7] Morikawa, R.Y. and Kerschberg, L., MAKO: Multi-Ontology Analytical Knowledge Organization based on Topic Maps. in Fifth International Workshop on Theory and Applications of Knowledge Management, (Zaragoza, Spain, 2004), IEEE.

[8] XML Topic Maps (XTM) 1.0, TopicMaps.Org Specification, www.topicmaps.org/xtm/1.0, 2003.

[9] L. Orbst and H. Liu, "Knowledge Representation, Ontological Engineering, and Topic Maps," in XML Topic Maps: Creating and Using Topic Maps for the Web, J. Parker and S. Hunting, Eds. Boston: Addison-Wesley, 2003, pp. Chapter 7.

[10] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880 March 2001.

[11] J. F. Sowa, Knowledge Representation: Logical, Philosophical, and Computational Foundations: Brooks/Cole Thomas Learning, 2000.