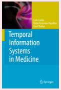
Chapter 5: Abstraction of Time-Oriented Clinical Data[1]

Carlo Combi[1] Elpida Keravnou-Papailiou[2]
Yuval Shahar[3]

[1]Department of Computer Science, University of Verona

[2]Department of Computer Science, University of Cyprus

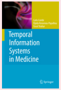
[3]Department of Information Systems Engineering, Ben Gurion University



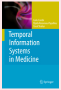
[1]Slides edited by Elena Gaspari, Univ. of Verona

Outline

1. Introduction: Temporal-Data Abstraction
2. Approaches to Temporal Data Abstraction
3. Time-Oriented Monitoring
4. Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators
5. Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology



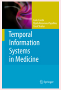
Outline



Knowledge-Based Systems in Medicine

Medical knowledge-based systems involve the application of medical knowledge to patient-specific data with the goal of

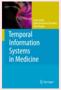
- reaching diagnoses or prognoses,
- deciding the best therapy regime for the patient.



Medical Knowledge and Data

To perform any kind of medical problem solving, patient data have to be "matched" against medical knowledge:

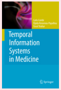
- a forward-driven rule is activated if its antecedent can be unified against patient information;
- a patient management protocol is activated if its underlying preconditions can be unified against patient information.



Temporal-Data Abstraction

The difficulty encountered here is that often the abstraction gap between the highly specific, raw patient data, and the highly abstract medical knowledge does not permit any direct unification between data and knowledge.

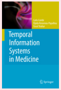
The process of data abstraction aims to close this gap.



Significance of Temporal-Data Abstraction

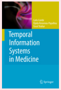
A knowledge-based system that does not possess data abstraction capabilities would require its user to express the case data at the level of abstraction corresponding to its knowledge.

- the user is burdened with the task of not only observing, measuring, and reporting data, but also of interpreting such data for the special needs of the particular problem solving;
- manual abstraction is prone to errors and inconsistencies even for domains where it can be considered "doable".



Types of (Atemporal) Data Abstraction

- **Qualitative abstraction**, where a numeric expression is mapped to a qualitative expression,
  - "a temperature of 41 degrees C" is abstracted to "high fever".
- **Generalization abstraction**, where an instance is mapped to (one of) its class(es),
  - "paracetamol is administered" is abstracted to "drug is administered".
- **Definitional abstraction**, where a datum from one conceptual category is mapped to a datum in another conceptual category that happens to be its definitional counterpart in the other context.
  - "generalized platyspondyly" is mapped to "short trunk".



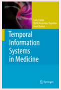
Towards Temporal Data Abstraction

In all the above types of data abstraction time is implicit. Thus in an atemporal situation, where everything is assumed to refer to 'now', we have the general implication

$$holds(P, D) \Longrightarrow holds(P, abs(D))$$

Predicate *holds* can be extended to have a third argument giving an explicit time:

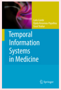
$$holds(P, D, T_1) \Longrightarrow holds(P, abs(D), T_2)$$



Types of Temporal Data Abstraction

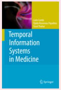
From point-based data to interval-based information

- *Merge (or State) Abstraction*: Deriving maximal intervals for some concatenable property from a group of time-stamped data for that property.
- *Persistence Abstraction*: Applying (default) persistence rules to project maximal intervals for some property, both backwards and forwards in time and possibly on the basis of a single datum.
- *Trend Abstraction*: Deriving significant changes and rates of change in the progression of some parameter.
- *Periodic Abstraction*: Deriving repetitive occurrences, with some regularity in the pattern of repetition.



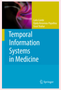
Using Temporal Data Abstraction I

1. **Data summaries** of time-oriented electronic data, such as patient medical records, have an immediate value to a human user, such as to a care provider scanning a long patient record for meaningful trends.
2. Temporal abstractions support **recommendations** by intelligent decision-support systems, such as diagnostic and therapeutic systems.
3. Abstractions support **monitoring** of plans (e. g., therapy plans) during execution of these plans (e. g., application of clinical guidelines).



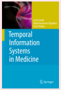
Using Temporal Data Abstraction II

4. Meaningful time-oriented contexts enable generation of **context-specific abstractions**, maintenance of **several interpretations** of the same data within different contexts, and certain hindsight and foresight inferences.
5. Temporal abstractions are helpful for **explanation** of recommended actions by an intelligent system.
6. Temporal abstractions are a useful representation for the intentions of designers of clinical guidelines, and enable real time and retrospective critiquing and quality assessment of the application of these guidelines by care providers.



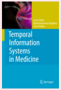
Using Temporal Data Abstraction III

7. Domain-specific, meaningful, interval-based characterizations of time-oriented medical data are a prerequisite for effective **visualization** and dynamic exploration of these data by care providers and researchers.



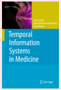
Computational Aspects of Abstraction Methods I

1. The method should be able to accept as input both *numeric* and *qualitative* data. Some of these data might be at *different levels of abstraction*. The data might also involve different forms of temporal representation (e.g., time *points* or time *intervals*).
2. The output abstractions should also be available for query purposes *at all levels of abstraction*, and should be created as time *points* or as time *intervals*, as necessary, aggregating relevant conclusions together as much as possible.



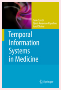
Computational Aspects of Abstraction Methods II

3. Input data should be used and incorporated in the interpretation even if they arrive *out of temporal order* (e.g., a laboratory result from last Tuesday arrives today). Thus, the past can change our view of the present. This phenomenon has been called a *view update*. Furthermore, new data should enable us to reflect on the past; thus, the present (or future) can change our interpretation of the past, a property referred to as *hindsight*.



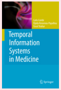
Computational Aspects of Abstraction Methods III

4. Several possible interpretations of the data might be reasonable, each depending on additional factors that are perhaps unknown at the time (such as whether the patient has AIDS); interpretation should be specific to the context in which it is applied.
5. The method should leave room for some *uncertainty* in the input and the expected data *values*, and some uncertainty in the *time* of the input or the expected temporal pattern.



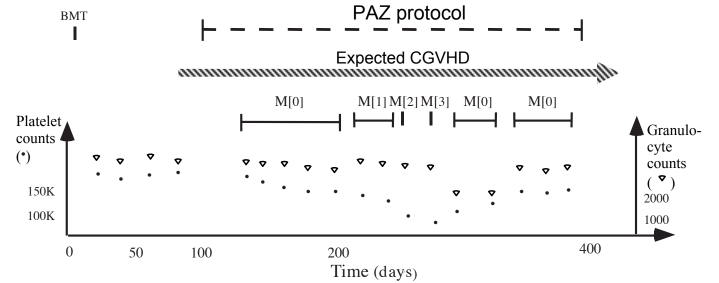
Computational Aspects of Abstraction Methods IV

6. The method should be generalizable to other clinical domains and tasks. The domain-specific assumptions underlying it should be explicit and as declarative as possible (as opposed to procedural code), so as to enable reuse of the method without rebuilding the system, *acquisition* of the necessary knowledge for applying it to other domains, *maintenance* of that knowledge, and *sharing* that knowledge with other applications in the same domain.



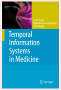
Synopsis of Temporal-Data Abstraction

Example



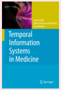
Temporal abstraction of platelet and granulocyte values during administration of a prednisone/azathioprine (PAZ) clinical protocol for treating patients who have chronic graft-versus-host disease (CGVHD).

Outline



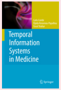
Approaches to Temporal Data Abstraction

- We now discuss a number of specific approaches to temporal data abstraction.
- Temporal data abstraction is primarily used for converting the raw data on some patient to more useful information. One of the first programs developed for this purpose is **Rx**.
- The principal role of temporal data abstraction, summarization of patient records, is addressed more extensively, through two pioneering systems, **IDEFIX** and **TOPAZ**.



Data Abstraction for Knowledge Discovery I

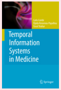
- Data abstraction and more specifically temporal data abstraction can be utilized for the discovery of medical knowledge.
- Data is patient specific, while knowledge is patient independent, it consists of generalizations that apply across patients.
- Machine learning for medical domains aims to discover medical knowledge by inducing generalizations from the records of representative samples of patients.



Data Abstraction for Knowledge Discovery II

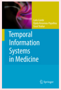
Different raw data can yield the same abstractions, even if they differ substantially in volume. The number of derived abstractions is relatively constant across patients with the same medical situation, and of course this number is considerably smaller than the number of raw data.

One of the goals behind the staging of a series of international workshops called **IDAMAP (Intelligent Data Analysis in Medicine and Pharmacology)** is to bring together the machine learning and temporal data abstraction communities interested in medical problems.



Data Abstraction for Knowledge Discovery III

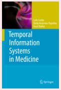
- The efforts have led to significant progress in a new area usually referred to as **temporal data mining**, in which interval-based abstractions are used as features for a supervised or non supervised data mining process that either characterizes the data as containing meaningful temporal patterns, or associates such patterns with a given outcome.



Discovery in Time-Oriented Clinical Databases: Blum's Rx Project I

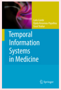
Rx was a program that examined a time-oriented clinical database, and produced a set of possible causal relationships among various clinical parameters.

- it used a **discovery module** for automated discovery of statistical correlations in clinical databases,
- a **study module** used a medical knowledge base to rule out spurious correlations by creating and testing a statistical model of a hypothesis.



Discovery in Time-Oriented Clinical Databases: Blum's Rx Project II

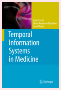
Data for Rx were provided from the **American Rheumatism Association Medical Information System (ARAMIS)**, a chronic-disease time-oriented database that accumulates time-stamped data about thousands of patients who have rheumatic diseases and who are usually followed for many years.



Discovery in Time-Oriented Clinical Databases: Blum's Rx Project III

The representation of data in the Rx program included:

- *point events*, such as a laboratory test,
- *interval events*, which required an extension to TOD (Time Oriented Database) to support diseases, the duration of which was typically more than one visit.



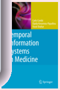
Discovery in Time-Oriented Clinical Databases: Blum's Rx Project IV

The medical knowledge base was organized into two hierarchies: **states** (e.g., disease categories,symptoms, and findings) and **actions** (drugs).

The Rx program determined whether interval-based complex states, such as diseases, existed by using a hierarchical **derivation tree**:

Definition

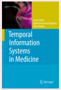
Event A can be defined in terms of events B_1 and B_2, which in turn can be derived from events C_{11}, C_{12}, C_{13} and C_{21}, C_{22}, and so on.



Discovery in Time-Oriented Clinical Databases: Blum's Rx Project V

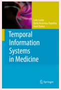
The main method used to access data at time points when a value for them did not necessarily exist used **time-dependent database access functions.**

- **delayed-action***(variable, day, onset-delay, interpolation-days)*: which returned the assumed value of *variable* at *onset-delay* days before *day*, but not if the last visit preceded day by more than interpolation-days days;
- **delayed-interval**: whose variable was an interval event, checked that no residual effects of the interval event remained within a given carryover time interval;



Discovery in Time-Oriented Clinical Databases: Blum's Rx Project VI

- **previous-value** *(variable, day)*: which returned the last value before day;
- **during** *(variable, day)*: which returned a value of variable if day fell within an episode of variable;
- **rapidly tapered** *(variable, slope)*: which returned the interval events in which the point event *variable* was decreasing at a rate greater than *slope*.

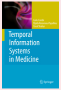


De Zegher-Geets' IDEFIX Program for Medical-Record Summarization I

De Zegher Geets' **IDEFIX** program creates an intelligent summary of the patient's current status, using an electronic medical record.

IDEFIX used the ARAMIS project's database (in particular, for patients who had systemic lupus erythematosus (SLE)).
The IDEFIX medical knowledge ontology included:

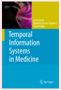
- abnormal primary attributes (APAs), such as the presence of protein in the urine;
- abnormal states, such as nephrotic syndrome;
- and diseases, such as SLE-related nephritis.



De Zegher-Geets' IDEFIX Program for Medical-Record Summarization II

APAs were derived directly from ARAMIS attribute values. IDEFIX inferred abnormal states from APAs: these states were essentially an intermediate-level diagnosis.

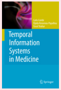
De Zegher-Geets added a novel improvement to Downs' program by using **time-oriented probabilistic functions (TOPFs)**.



De Zegher-Geets' IDEFIX Program for Medical-Record Summarization III

Definition

A **TOPF** was a function that returned the conditional probability of a disease D given a manifestation M, $P(D|M)$, as a function of a time interval, if such a time interval was found.

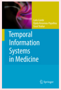


De Zegher-Geets' IDEFIX Program for Medical-Record Summarization IV

Example

PREVIOUS. ADJACENT. EPISODE(LUPUS.NEPHRITIS) looked for the time since the last episode of lupus nephritis. Thus, *as time progressed, the strength of the (probabilistic) connection between the disease and the manifestation could be changed in a predefined way*.

The goal of the IDEFIX reasoning module was to explain, for a particular patient visit, the various manifestations for that visit, taking as certain all previous data.



De Zegher-Geets' IDEFIX Program for Medical-Record Summarization V

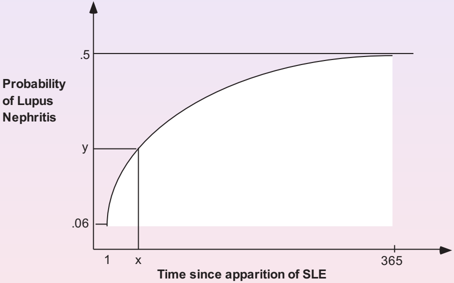
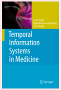


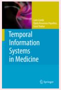
Figure: A time-oriented probabilistic function (TOPF) associated with the predicate "previous episode of lupus nephritis. ".



Kahn's TOPAZ System: an Integrated Interpretation Model I

Kahn has suggested using more than one temporal model to exploit the full power of different formalisms of representing medical knowledge.

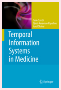
1. A numeric model represented quantitatively the underlying processes. When the system processed the initial data, the model represented a *prototypical-patient model*. That model was specialized for a particular patient, thus turning it into an *atemporal patient-specific model*-by addition of details. Finally, the parameters in the atemporal patient-specific model were adjusted to fit actual



Kahn's TOPAZ System: an Integrated Interpretation Model II

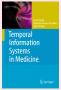
patient-specific data that accumulate over time, turning the model into a *patient-specific temporal model*.

2. A symbolic interval-based model aggregated intervals that were clinically interesting in the sense that they violated expectations. The model encoded abstractions as a hierarchy of symbolic intervals.



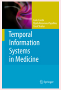
Kahn's TOPAZ System: an Integrated Interpretation Model III

3. A symbolic state-based model generated text paragraphs that used the domain's language, from the interval-based abstractions, using a representation based on augmented transition networks (ATNs). The ATNs encoded the possible summary statements as a network of potential interesting states.



Kahn's TOPAZ System: an Integrated Interpretation Model IV

TOPAZ used the patient-specific predictions, not the actual observed data, for comparisons to the expected population data. The reason for this choice was that data produced for patient-specific predictions (assuming a correct, complete, patient-specific model) should be *smoother* than actual data and should contain fewer spurious values.



Kahn's TOPAZ System: an Integrated Interpretation Model V

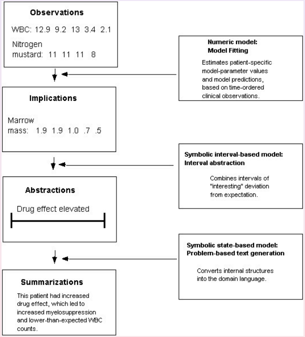
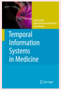
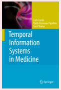


Figure: Summarization of time-ordered data in the TOPAZ system.



Outline

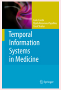
1. Introduction: Temporal-Data Abstraction
2. Approaches to Temporal Data Abstraction
3. Time-Oriented Monitoring
4. Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators
5. Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology



Time-Oriented Monitoring I

- Most clinical monitoring tasks require measurement and capture over time of numerous patient data, often on electronic media.
- Most stored clinical data include a time stamp in which the particular datum is valid; an emerging pattern over a stretch of time has much more significance than an isolated finding or even a set of findings.

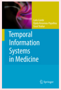
Thus, it is desirable to provide short, informative, context-sensitive summaries of time-oriented clinical data stored on electronic media, and to be able to answer queries about abstract concepts that summarize the data.



Fagan's VM Program: A State-Transition Temporal-Interpretation Model I

Fagan's **VM** system was one of the first knowledge-based systems that included an explicit representation for time. It was designed to assist care providers managing patients on ventilators in intensive-care units.

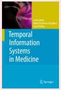
- VM could reason explicitly about time units;
- accept time-stamped measurements of patient parameters;
- calculate time-dependent concepts such as rates of change;
- relied on a state-transition model of different intensive-care therapeutic situations, or contexts.



Fagan's VM Program: A State-Transition Temporal-Interpretation Model II

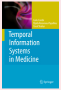
In each context, different expectation rules would apply to determine what, for instance, is an ACCEPTABLE mean arterial pressure in a particular context.

Another point to note is that the VM program used a classification of expiration dates of parameters, signifying for how long VM could assume the correctness of the parameter's value if that value was not sampled again. The expiration date value was used to fill a GOOD-FOR slot in the parameter's description.



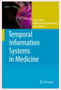
Fagan's VM Program: A State-Transition Temporal-Interpretation Model III

- VM could not accept data arriving out of order, such as blood-gas results that arrive after the current context has changed, and thus could not revise past conclusions. In that sense, VM could not create a valid *historical* database.
- it did store the last hour of parameter measurements and all former conclusions; in that respect, VM maintained a *rollback* database of measurements and conclusions.



Temporal Bookkeeping: Russ' Temporal Control Structure I

Russ designed a system called the temporal control structure (**TCS**), which supports reasoning in time-oriented domains, by allowing the domain-specific inference procedures to ignore temporal issues, such as the particular time stamps attached to values of measured variables.

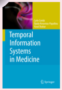


Temporal Bookkeeping: Russ' Temporal Control Structure II

Definition

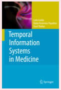
The main emphasis in the TCS methodology is creating what Russ terms as a **state abstraction**:

- an abstraction of continuous processes into steady-state time intervals, when all the database variables relevant for the knowledge-based system's reasoning modules are known to be fixed at some particular value.



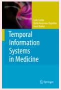
Temporal Bookkeeping: Russ' Temporal Control Structure III

- The TCS system allows user-defined code modules that reason over the homogeneous intervals, as well as user-defined data variables that hold the data in the database.
- It also creates a process for each time interval in which a module is executed; the process has access only to those input data that occur within that time interval.



Temporal Bookkeeping: Russ' Temporal Control Structure IV

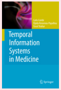
- The underlying temporal primitive in the TCS architecture is a time *point* denoting an exact date. Propositions are represented by point variables or by interval variables. Intervals are created by an abstraction process that employs user-defined procedural Lisp code inside the TCS modules to create steady-state periods.



Temporal Bookkeeping: Russ' Temporal Control Structure V

Features:

- **truth-maintenance** capability of the system: the abilities to maintain dependencies among data and conclusions in every steady-state interval, and to propagate the effects of a change in past or present values of parameters to all concerned reasoning modules. Thus, the TCS system creates a *historical* database.
- the ability to reason by **hindsight**: to reassess past conclusions based on new, present data.



Temporal Bookkeeping: Russ' Temporal Control Structure VI

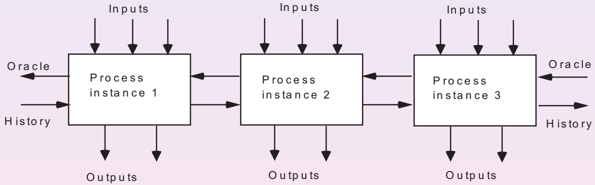
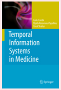


Figure: A chain of processes in the TCS system. Each process has in it user-defined code, a set of predefined inputs and outputs, and memory variables connecting it to future processes (oracle variables) and to past processes (history variables).

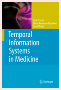


Kohane's Temporal Utilities Package I

Kohane has written a general-purpose **temporal-utilities package** (**TUP**) for representing qualitative and quantitative relations among temporal intervals, and for maintaining and propagating the constraints posed by these relations through a *constraint network* of temporal (or any other) intervals.

The TUP system used a point-based temporal ontology:

- Intervals were represented implicitly by the relations between their start points and end points, or by the relations between these points and points belonging to other intervals.
- These relations were called **range relations** (**RRELs**).

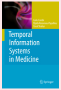


Kohane's Temporal Utilities Package II

The RREL constrains the temporal distance between two points to be between the given lower bound and the upper bound in a certain context.
Here is a simplified RREL:

*(<first-point specification> <second-point specification>
<lower-bound distance> <upper-bound distance>
<context>)*

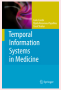
Kohane had implemented a point-based strategy for representing some of Allen's interval-based relations, namely those that can be expressed solely by constraints between two points.



Kohane's Temporal Utilities Package III

Example

For instance, to specify that interval *A* precedes interval *B*, it is sufficient to maintain the constraint that
"the end of *A* is between $+\infty$ and $+\varepsilon$ before the start of *B*. "

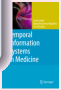


Haimowitz and Kohane's TrenDx System I

TrenDx focuses on using efficient general methods for representing and detecting predefined temporal patterns in raw time-stamped data.

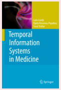
Definition

Trend templates (TTs) describe typical clinical temporal patterns, such as normal growth development, or specific types of patterns known to be associated with functional states or disease states, by representing these patterns as *horizontal* (temporal) and *vertical* (measurement) constraints.



Haimowitz and Kohane's TrenDx System II

- TrenDx has the rather unique ability to match *partial* patterns by maintaining an agenda of candidate patterns that *possibly* match an evolving pattern.
- A TT indeed provides a powerful mechanism for expressing the dynamics of some process, in terms of the different phases comprising it, the uncertainty governing the transitions from one phase to the next, the significant events marking these transitions and various constraints on parameter-values associated with the different phases.



Haimowitz and Kohane's TrenDx System III

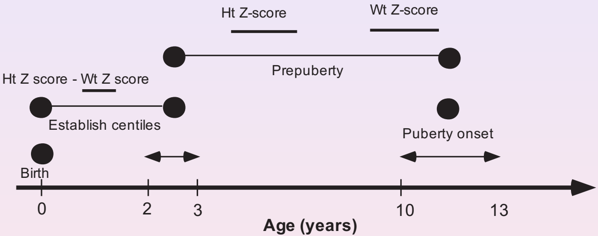
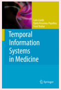


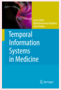
Figure: A portion of a trend template (TT) in TrenDx that describes the male average normal growth as a set of functional and interval-based constraints.



Larizza et al. 's Temporal-Abstraction Module in the M-HTP System I

M-HTP is a system for monitoring heart-transplant patients that has a module for abstracting time-stamped clinical data.

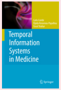
- The system generates abstractions such as HB-DECREASING, and maintains a temporal network (TN) of temporal intervals, using a design inspired by Kahn's TNET temporal-maintenance system.
- Uses an object-oriented visit taxonomy and indexes parameters by visits.



Larizza et al. 's Temporal-Abstraction Module in the M-HTP System II

- Has an object-oriented knowledge base that defines a taxonomy of significant-episodes-clinically interesting concepts such as DIARRHEA or WBC_DECREASE.

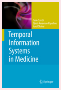
The temporal model of the M-HTP system includes both time points and intervals. The M-HTP system uses a temporal query language to define the antecedent part of its rules.



Miksch et al. 's VIE-VENT System I

Miksch et al.have developed **VIE-VENT**, a system for *data validation* and therapy planning for artificially ventilated newborn infants.

- The overall aim is the context-based validation and interpretation of temporal data, where data can be of different types.
- The interpretation contexts are not dynamically derived, but they are defined through schemata with thresholds that can be dynamically tailored to the patient under examination.

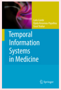


Miksch et al. 's VIE-VENT System II

- The context schemata correspond to potential treatment regimes; which context is actually active depends on the current regime of the patient.

The types of knowledge required are classification knowledge and temporal dynamic knowledge.

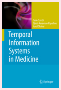
Everything is expressed declaratively in terms of schemata that can be dynamically adjusted depending on the state of the patient. First quantitative point-based data are translated into qualitative values, depending on the operative context.



Miksch et al. 's VIE-VENT System III

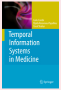
The system deals with four types of trends:

1. *very short-term*,
2. *short-term*,
3. *medium-term*,
4. *long-term*.



Outline

1. Introduction: Temporal-Data Abstraction
2. Approaches to Temporal Data Abstraction
3. Time-Oriented Monitoring
4. **Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators**
5. Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology

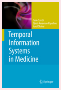


Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators I

In addition to **reasoning** about time-oriented medical data, it is also necessary to consider the **management** of these data: insertion, deletion, and query, tasks often collectively referred to as **temporal-data maintenance**.

Includes the capability to store and retrieve also the different **temporal dimensions**:

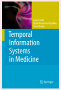
- the *transaction time*, that is, the time at which data are stored in the database;
- the *valid time*, that is, the time at which the data are true for the modeled real world entity;



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators II

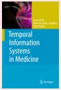
- the *user-defined time*, whose meaning is related to the application and thus is defined by the user.

Several systems allow not only the modeling of complex clinical concepts at the database level, but also the maintenance of certain inference operations at that level. For example, **active databases** can also store and query derived data.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators III

- When building a time-oriented decision-support application, one needs to consider the mode of integration between the application, the data-abstraction process (essentially, a temporal-reasoning task), and the temporal-maintenance aspect of the system.
- Data abstraction is a critical auxiliary process. It is usually deployed in the context of a higher-level problem solving system, it is knowledge-based, and it operates in a goal- or event-driven fashion, or both.

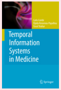


Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators IV

Definition

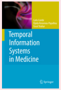
In general, one can conceive of three basic modes of integration among the temporal-data abstraction process, the temporal-data management process, a medical decision-support application (e.g., diagnosis, therapy), and a time-oriented database.

1. Incorporating the abstraction process within the database;
2. Adding a data-management capability to the application system, assuming an inherent data-abstraction capability;



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators V

3. Out-sourcing both the temporal-data management and the temporal-data abstraction capabilities, by encapsulating them within an intermediate *mediator* that is independent of either.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators VI

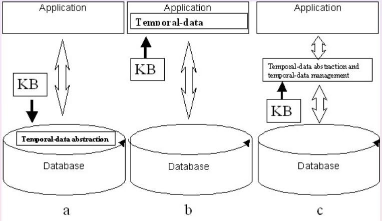
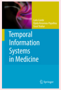


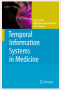
Figure: Three modes of integrating the temporal-data abstraction and temporal-data management processes with a medical decision-support application (e.g., diagnosis, therapy) and a time-oriented database.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators VII

The concept of a mediator has been proposed in the early 1990s. It is called a **mediator** because it serves as an intermediate layer of processing between client applications and databases.

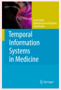
Modern architectures, such as the Chronus-2 temporal mediator, which uses a highly expressive temporal-query language, and the **IDAN** temporal-abstraction architecture have extended the temporal-mediation idea.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators VIII

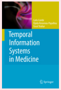
The IDAN architecture is more uniform, because a subset of the temporal- and value-constraints language, the temporal-abstraction rule (TAR) language, which is used in its internal temporal-abstraction computational component, the ALMA system, is used in the query interface of the temporal-abstraction mediator's controller.

- IDAN is also fully distributed and accesses multiple clinical databases, medical knowledge bases, and, in theory, multiple computational temporal-abstraction modules.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators IX

- An IDAN session starts by defining a particular data, knowledge, and processing configuration and then referring raw-data or abstract-concept queries to the controller.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators X

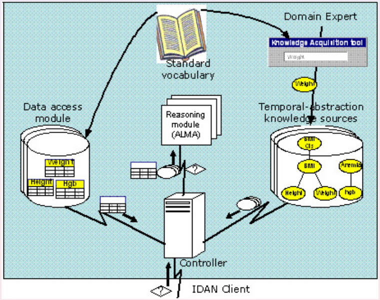


Figure: An overall view of the IDAN temporal-abstraction architecture.

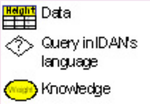
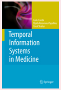


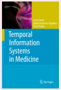
Figure: Legend.



Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators XI

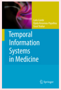
The looseness or otherwise of coupling between a data abstraction process and a problem solving system can be decided on the basis of the following questions:

1. *Is the ontology underlying the specific knowledge domain independent?* If so, then removing that knowledge and incorporating a knowledge-acquisition component that functions to fill the given knowledge base with the relevant knowledge from another domain will result in a traditional skeletal system for data abstraction, applicable to different domains for the same task.



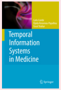
Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators XII

2. *Is the overall ontology task independent?* If so, we can obtain a skeletal system for data abstraction, applicable to different tasks within the same domain.
3. *Is the specific knowledge task independent?* If so, the data abstraction process is already applicable to different tasks within the same domain.

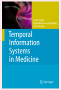


Merging Temporal Reasoning and Temporal Maintenance: Temporal Mediators XIII

4. *Do generated abstractions constitute the system's main and final output?* If so, the data abstraction process is strongly coupled to the problem solving system. In the spirit of the new generation of knowledge-engineering methodologies, the objective should be to form a *library* of generic data abstraction methods, with different underlying ontologies and computational mechanisms.



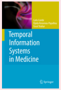
Outline



Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology I

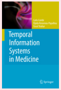
Shahar's **knowledge-based temporal-abstraction method (KBTA)** is a temporal-data abstraction ontology that addresses the issue of providing a comprehensive conceptual model for that process.

- The main motivation for Shahar's methodology was clinical-data summarization and query for monitoring, therapy, quality assessment, and clinical research.



Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology II

- Shahar defined a knowledge-based framework, including a formal temporal ontology and a set of computational mechanisms using that ontology specific to the task of creating abstract, interval-based concepts from time-stamped clinical data.



Shahar's Knowledge-Based Temporal-Abstraction Method and Ontology III

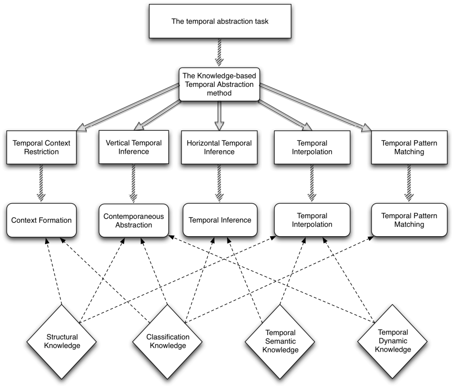
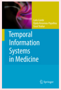


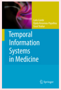
Figure: The knowledge-based temporal-abstraction (KBTA) method.



The Knowledge-Based Temporal-Abstraction Ontology I

The KBTA theory defines the following set of entities:

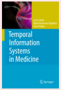
1. The basic time primitives are **time stamps**, $T_i \in \mathrm{T}$. Time stamps are structures (e.g., dates) that can be mapped, by a time-standardization function $f_s(T_i)$, into an integer amount of any element of a set of predefined *temporal granularity units* $G_i \in \Gamma$ (e.g., DAY). A *zero-point* time stamp (the start of the positive time line) must exist. Time stamps are therefore either positive or negative shifts from the zero point measured in the G_i units. The domain must have a time unit G_0 of the lowest granularity (e.g., SECOND); there must exist a mapping from any integer



The Knowledge-Based Temporal-Abstraction Ontology II

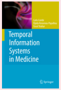
amount of granularity units G_i into an integer amount of G_0. A finite negative or positive integer amount of G_i units is a *time measure*.

The special symbols $+\infty$ and $-\infty$ are both time stamps and time measures, denoting the furthest future and the most remote past, respectively.



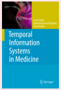
The Knowledge-Based Temporal-Abstraction Ontology III

2. A **time interval** *I* is an ordered pair of time stamps representing the interval's end points: [*I.start*, *I.end*]. *Time points* T_i are therefore represented as zero-length intervals where $I.start = I.end$. Propositions can be interpreted only over time intervals.
   A time interval can be closed on both sides, open on both sides, or closed on the left and open on the right.



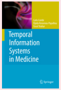
The Knowledge-Based Temporal-Abstraction Ontology IV

3. An **interpretation context** $\xi \in \Xi$ is a proposition. Intuitively, it represents a state of affairs that, when interpreted (logically) over a time interval, can change the interpretation (abstraction) of one or more parameters within the scope of that time interval. IS-A and SUBCONTEXT relations are defined over the set of interpretation contexts. *Basic* interpretation contexts are atomic propositions. An interpretation context in conjunction with one of its subcontexts can create a *composite interpretation context*. Composite interpretation contexts are interpretation contexts.



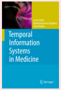
The Knowledge-Based Temporal-Abstraction Ontology V

Formally, the structure $< \xi_i, \xi_j >$ is an interpretation context, if the ordered pair (ξ_j, ξ_i) belongs to the SUBCONTEXT relation (i.e., ξ_j is a subcontext of ξ_i). In general, if the structure $< \xi_1, \xi_2, ...\xi_i >$ is a (composite) interpretation context, and the ordered pair (ξ_j, ξ_i) belongs to the SUBCONTEXT relation, then the structure $< \xi_1, \xi_2, ...\xi_i, \xi_j >$ is a (composite) interpretation context.



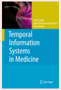
The Knowledge-Based Temporal-Abstraction Ontology VI

4. A **context interval** is a structure $<\xi, I>$, consisting of an interpretation context ξ and a temporal interval I. Intuitively, a context interval represents an interpretation context interpreted over a time interval; the interpretation of one or more parameters is different within the temporal scope of that interval.



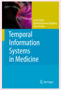
The Knowledge-Based Temporal-Abstraction Ontology VII

5. An **event proposition** $e \in E$ (or an **event**, for short, when no ambiguity exists) represents the occurrence of an external volitional action or process, such as the administration of a drug (as opposed to a measurable datum, such as temperature). Events have a series a_i of *event attributes* (e.g., dose) and a corresponding series ν_i of *attribute values*.



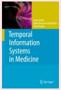
The Knowledge-Based Temporal-Abstraction Ontology VIII

An IS-A hierarchy of *event schemata* (or event types) exists. Event schemata have a list of attributes a_i, where each attribute has a domain of possible values V_i, but do not necessarily contain any corresponding attribute values. Thus, an *event proposition* is an event schema in which each attribute a_i is mapped to some value $\nu_i \in V_i$. A PART-OF relation is defined over the set of event schemata. If the pair of event schemata (e_i, e_j) belongs to the PART-OF relation, then event schema e_i can be a *subevent* of an event schema e_j.



The Knowledge-Based Temporal-Abstraction Ontology IX

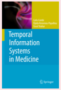
6. An **event interval** is a structure $< e, I >$, consisting of an event proposition e and a time interval I. The time interval I represents the duration of the event.
7. A **parameter schema** (or a **parameter**, for short) $\pi \in \Pi$ is, intuitively, a measurable aspect or a describable state of the world, such as a patient's temperature. Parameter schemata have various *properties*, such as a domain V_π of possible symbolic or numeric values, measurement units, and a measurement scale.



The Knowledge-Based Temporal-Abstraction Ontology X

Intuitively, parameters denote either input (usually raw) data, or any level of abstraction of the raw data (up to a whole pattern).

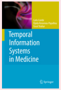
Primitive parameters are parameters that play the role of raw data in the particular domain in which the TA task is being solved. They cannot be inferred by the TA process from any other parameters (e.g., laboratory measurements).



The Knowledge-Based Temporal-Abstraction Ontology XI

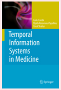
Abstract parameters are parameters that play the role of intermediate concepts at various levels of abstraction; these parameters can be part of the output of the TA task, having been *abstracted* from other parameters and events, or they may be given as part of the input (e.g., the value of the state of Hemoglobin is MODERATE_ANEMIA).

Constant parameters are parameters that are considered atemporal in the context of the particular interpretation task that is being performed, so their values are not expected to be time-dependent (e.g., the patient's gender, the patient's address, the patient's father's height).



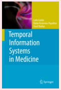
The Knowledge-Based Temporal-Abstraction Ontology XII

8. **Abstraction functions** $\theta \in \Theta$ are unary or multiple-argument functions from one or more parameters to an abstract parameter. The "output" abstract parameters can have one of several *abstraction types*. We distinguish among at least three basic abstraction types: *state*, *gradient*, and *rate*.



The Knowledge-Based Temporal-Abstraction Ontology XIII

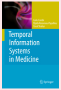
The θ abstraction of a parameter schema π is a new parameter schema $\theta(\pi)$-a parameter different from any of the arguments of the θ function (e.g., STATE(Hemo-globin), which we will write as Hemoglobin_state). This new parameter has its own domain of values and other properties (e.g., scale), typically different from those of the parameters from which it was abstracted. It can also be abstracted further (e.g., GRADIENT(STATE(Hemoglobin))).



The Knowledge-Based Temporal-Abstraction Ontology XIV

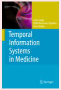
A special type of abstraction function (and a respective proposition type) is **pattern**: A function that creates a temporal pattern from temporal intervals, over which hold parameters, events, contexts, or other patterns (e.g., a QUIESCENT-ONSET pattern of chronic graft-versus-host disease). Patterns can be *linear* or *periodic*.

Statistics such as **minimum**, **maximum**, and **average value** are not abstraction types in this ontology. Rather, these statistics are *functions* on parameter *values* that return simply a *value* of a parameter.



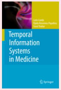
The Knowledge-Based Temporal-Abstraction Ontology XV

9. A **parameter interval** is a tuple $< \pi, \nu, \xi, I >$, where $< \pi, \nu, \xi >$ is a parameter proposition and *I* is a time interval. If *I* is in fact a time point (i.e., *I.start* $=$ *I.end*), then the tuple can be referred to as a *parameter point*. Intuitively, a parameter interval denotes the value ν of parameter π in the context ξ during time interval *I*. The value of parameter π at the beginning of interval *I* is denoted as *I.start*.π, and the value of parameter π at the end of interval *I* as *I.end*.π. *Pattern intervals* are defined in a similar fashion.



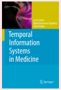
The Knowledge-Based Temporal-Abstraction Ontology XVI

10. An **abstraction** is a parameter or pattern interval $< \pi, \nu, \xi, I >$, where π is an abstract parameter or pattern. If I is in fact a time point (i.e., $I.start = I.end$), the abstraction can also be referred to as an *abstraction point*; otherwise, we can refer to it as an *abstraction interval*.
11. An **abstraction goal** $\psi \in \Psi$ is a proposition that denotes a particular goal or intention that is relevant to the TA task during some interval (e.g., diagnosis).



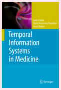
The Knowledge-Based Temporal-Abstraction Ontology XVII

12. An **abstraction-goal interval** is a structure $< \psi, I >$, where ψ is an abstraction goal and I is a time interval. Intuitively, an abstraction-goal interval represents the fact that an *intention* holds or that a TA *goal* (e.g., the goal of monitoring AIDS patients) should be achieved during the time interval over which it is interpreted.



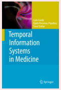
The Knowledge-Based Temporal-Abstraction Ontology XVIII

13. **Induction of context intervals**: Intuitively, context intervals are inferred dynamically (at runtime) by certain event, parameter, or abstraction-goal propositions being true over specific time intervals. The contexts interpreted over these intervals are said to be **induced** by these propositions (e.g., by the event "administration of 4 units of regular insulin").



The Knowledge-Based Temporal-Abstraction Ontology XIX

Two or more context-forming propositions induce a *composite interpretation context*, when the temporal spans of their corresponding induced context intervals intersect, if the interpretation contexts that hold during these intervals belong to the SUBCONTEXT relation. Figure below shows inducement of context intervals within, after, and even before an inducing event.



The Knowledge-Based Temporal-Abstraction Ontology XX

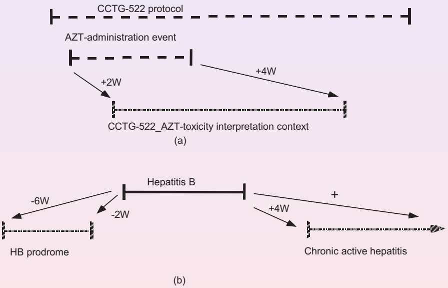
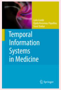
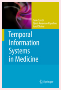


Figure: Dynamic induction relations of context intervals (DIRCs).



The Knowledge-Based Temporal-Abstraction Ontology XXI

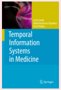
Formally, a **dynamic induction relation of a context interval** (**DIRC**) is a relation on propositions and time measures, in which each member is a structure of the form $< \xi, \varphi, ss, se, es, ee >$. The symbol ξ is the interpretation context that is induced. The symbol $\varphi \in P$ is the *inducing proposition*: an event, an abstraction-goal, or a parameter proposition. Each of the other four symbols denotes a time measure or the wildcard symbol *. A proposition φ that is an inducing proposition in at least one DIRC is a *context-forming proposition*.



The Knowledge-Based Temporal-Abstraction Ontology XXII

There exists also a set of *temporal queries*, expressed in a predefined **TA query language** that includes constraints on parameter values and on relations among start-point and end-point values among various time intervals and context intervals. The TA language is used:

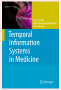
- to define the relationship between a pattern-type abstraction and its defining component intervals,
- to ask arbitrary queries about the result of the TA inference process.



The Knowledge-Based Temporal-Abstraction Ontology XXIII

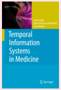
The *TA task* solved by the KBTA method is defined as follows:
Given

- at least one abstraction-goal interval $< \psi, I >$,
- a set of event intervals $< e_j, I_j >$,
- a set of parameter intervals $< \pi_k, \nu_k, \xi_k, I_k >$,
- the domain's temporal-abstraction ontology,



The Knowledge-Based Temporal-Abstraction Ontology XXIV

produce an interpretation (a set of context intervals $< \xi_n, I_n >$ and a set of (new) abstractions $< \pi_m, \nu_m, \xi_m, I_m >$) that can answer any temporal query about all the abstractions derivable from the transitive closure of the input data and the domain knowledge.

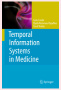


The RÉSUMÉ system I

The KBTA method has been implemented within the RÉSUMÉ problem solver.

The RÉSUMÉ system has been evaluated for the purpose of summarizing data in multiple clinical domains, such as:

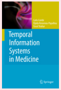
- oncology,
- monitoring of children's growth,
- management of insulin-dependent diabetes.



The RÉSUMÉ system II

In addition, the RÉSUMÉ system and its various versions had been used to support guideline-based application and guideline-based quality assessment of medical care, as well as for visualization and exploration of time-oriented patient data and their abstractions, a task in which the system and its various interfaces have been extensively evaluated.

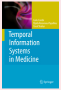
- The KBTA method has been implemented within multiple applications, in addition to the RÉSUMÉ system.
- The KBTA method has been extended into an incremental, data-driven version, in which raw data arrive in real-time and are continuously abstracted into meaningful interval-based abstractions.



The RÉSUMÉ system III

The incremental KBTA (**IKBTA**) method has been implemented within the *Momentum* system.

A Key operation in the IKBTA method is the incremental temporal-interpolation mechanism, whose time complexity is $O(n)$ or $O(nlog(n))$, depending on whether the data are assumed to arrive in order or out of order (thus requiring a truth-maintenance system), respectively.

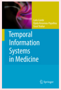


The Temporal-Abstraction Knowledge-Acquisition Tool I

Definition

One of the principles of modern knowledge-based problem-solving methodologies is the emphasis on supporting maintenance of the domain knowledge by the domain experts.

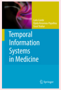
The knowledge required by the knowledge-based temporal-abstraction method (KBTA) needs to be acquired from medical domain experts.



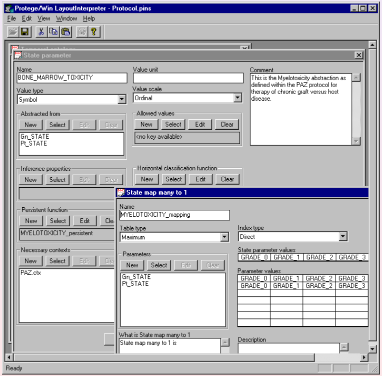
The Temporal-Abstraction Knowledge-Acquisition Tool II

A tool for elicitation and maintenance of temporal-abstraction knowledge had been designed using the **Protégé** framework:

- aims to develop a library of highly-reusable, domain-independent, problem-solving method.
- advantage: the production, given the relevant problem-solving-method and domain ontologies, of automated knowledge-acquisition tools, tailored for the selected problem-solving method and domain.



The Temporal-Abstraction Knowledge-Acquisition Tool III



Use of the PROTÉGÉ-based temporal abstraction knowledge-acquisition tool in the oncology domain.