Visualizing Complex Notions of Time

Robert Kosara, Silvia Miksch

Institute of Software Technology, Vienna University of Technology, Vienna, Austria

Abstract

Time plays an important role in medicine. Conditions are not just evaluated at single instants in time, but traced over periods. Medications must be administered within specified temporal limits, and their effects observed with regard to time.

When planning treatments, the temporal aspect becomes even more complicated. The planner has to deal with uncertainty and allowable intervals. A visual representation of the information would be helpful, but there are few visualizations of time that are powerful enough.

We present a visualization that graphically represents a complex notion of time, and has also been implemented in a program that allows users to directly specify this information. The results of a small user study are reported.

Keywords:

Time; Time Management; Data Display; User-Computer Interface; Clinical Protocols;

Introduction

Treatments in medicine involve many decisions and processes that are very time-dependent. The simplest example are time series of patient data, which form the basis for many decisions. Treatment monitoring is another example: if a treatment fails to reach its goal in a given amount of time, it is aborted and another one is tried.

These two examples have one thing in common: they deal with information from the past. As soon as a data value is recorded, the exact point in time at which the measurement was taken is known, and the value can be plotted into a diagram, for example.

When planning treatments, this is different. Actions can never be planned precisely, each patient reacts differently to medication, and there are complications that cannot be foreseen for a single patient.

Most graphical representations of time do not provide means for displaying such uncertain information. With uncertainty, we mean time intervals where bounds for start and end times as well as duration are at least partially known.

The Asgaard Project

In the Asgaard Project, we develop methods for representing clinical guidelines and protocols, and assisting the medical staff in implementing them. In this section, a short overview over some of the key features of Asbru is given. A thorough introduction is far beyond the scope of this paper; please refer to [1] for more information.

We are using a plan representation language called Asbru [2] for specifying medical protocols. Each protocol is translated into many Asbru plans — therefore, we use the term *plan* instead of *protocol* in this paper.

An Asbru plan can have subplans. If it does, its type specifies how its subplans are arranged in time. The type can be sequential, parallel, any-order or unordered (this includes partly parallel execution, which any-order does not allow). Depending on their definition, subplans do not have to be performed. They can be optional, and their success or failure can be ignored. By default, all plans are non-optional and their success affects the success of the containing plan.

If a plan does not have subplans, it is considered an action (that is performed by medical staff or a device, like a respirator).

Plans are governed by conditions that specify when a plan can be applied, when it has to be aborted, has completed (i.e., reached its goal), etc. There are also other so-called *knowledge roles* that specify the effects of a plan, its intentions, etc.

Asbru's Time Annotation

One of the main components of Asbru is the *time annotation*. It has several uses: the most common is specifying the temporal extent of a condition. In Asbru, a condition does not consist only of a value comparison ("if (body_temperature > 38) do ..."), but also contains a specification of the time span in which this condition must hold ("if ((body_temperature > 38) for at least 10 hours but not longer than 20 hours) do ..."). Because medical therapy plans have to deal with uncertainty, this time span specification does not just consist of a simple interval, but has to have much more power. A condition with a time annotation is called a *temporal pattern*.

Another use for time annotations is to specify the time

Figure 1- Time Lines

frame in which an action has to take place (e.g., a treatment has to be performed within certain limits after diagnosis).

Time annotations consist of the parts described in the following list. Any of these parts (except the reference point) can be left unspecified, to denote that this information is not important.

Reference Point. This is the point that all the other points in time are defined relative to. It can be an abstract point in time (e.g., *conception*), or refer to a plan state transition (e.g., the point at which the previous plan was completed).

Earliest Starting Shift (ESS). The smallest offset from the reference point when the action or condition can start. If it has started earlier, the time annotation is not fulfilled. By leaving this field undefined, it is possible to have the condition start at any point before the latest starting shift.

Latest Starting Shift (LSS). The latest point in time when the action must start, or the condition must be true. If it has not started at this point, the temporal pattern fails.

Earliest Finishing Shift (EFS). The earliest point in time when the action can end. If it ends earlier, the temporal pattern fails.

Latest Finishing Shift (LFS). The greatest offset from the reference point when the action must end, or the condition must become false for the temporal pattern to be true. If the LFS is specified, the temporal pattern can only be decided after it has passed.

Minimum Duration (MinDu). The minimum amount of time the action or condition must last. This is not necessarily identical with the interval between LSS and EFS. It cannot be shorter, however, than this difference, and not be longer than the maximum duration.

Maximum Duration (MaxDu). The maximum duration that the condition or action may last. It is bounded by the difference between LFS and ESS, and the minimum duration.

Information Visualization

Information Visualization [3] graphically depicts information that does not have an inherent spatial structure. Examples for such data are file systems, patient data (body temperature, ventilation parameters, etc), general data bases, and of course temporal information.

Many techniques exist to make information visible so as to get a better overview, to understand correlations, and to perceive the data more quickly (looking at an image makes it possible to get an idea of the structure of the data much faster than reading hundreds or thousands of numbers).

The importance of information visualization is only starting to be understood by practitioners in many fields. But everybody is aware of visualizations such as maps, bar and pie charts, flow diagrams, etc.

Visualization Requirements

Based on the description above, we want to develop a visualization of temporal constraints. These are the requirements we want this visualization to fulfill.

Allen's Relations. Any visual representation of time must of course be able to visually represent all possible relationships between intervals. There are 13 such relations described in [4]: A before B (A ends before B starts), A meets B (B starts at the same instant that A ends), A overlaps B (B starts before A ends), A starts B (A and B start at the same time), A contains B (A is shorter than B and starts after B starts), A finishes B (B ends at the same time A ends).

For each of these six relations, there also is a mirrored one plus the commutative relation A equals B.

Temporal Uncertainty. The representation must be able to deal with temporal uncertainty as described in the previous section.

Undefined/Unknown Parts. If parts of a time annotation are unknown (denoted in Asbru as "_", see Figure 3), they must be depicted in a way to make this easily recognizable without at the same time being too dominant.

Resolution. In addition to the temporal uncertainty of actions (see previous section), it should also be possible to see to what precision a point in time has been specified relative to the scale it is currently being viewed on.

Hierarchical Decomposition. It must also be possible to communicate the fact that plans are made up of sub-plans. This hierarchical decomposition is important not only to structure a plan and to make parts reusable, but also to be able to select the amount of information visible by showing more or fewer levels at the same time.

Facets. Different kinds of information about the same object should be visible at the same time. One of the key ideas for this are facets (see Figure 1).

Related Work

This section describes two types of time visualization. Time Lines are relatively widely used, and have been extended into LifeLines. SOPOs are a very powerful but little intuitive depiction of time.

Time Lines, LifeLines

LifeLines [5] are an extension of a rather old concept called time lines [6], which is a very intuitive way of representing time spans. On a two-dimensional, Cartesian coordinate system, one axis (usually the horizontal one) represents time, and one axis is divided into segments for different events (Figure 1). A line (or box) is drawn over the time span that the event takes place in.

One of the extensions introduced in [5] are facets. Facets are vertical segments that group similar events. They can be opened and closed to provide the user with information on different aspects of the same structures without cluttering the display with too much information. They are usually used to show different aspects (views) of the same information.

A similar idea to facets is used in [7] to visualize different paths through a vaccination plan. In addition to the basic time line idea, arrows are drawn from the decision points of the plan to alternatives forking off at this point.

Sets of Possible Occurrences (SOPOs)

An interesting way of looking at time are sets of possible occurrences (SOPOs) [8]. This diagram uses two time axes that represent the begin and end times of an interval, respectively (Figure 2). Any point in this diagram represents a whole interval, specified by its start (x coordinate) and end time (y coordinate). The area a SOPO covers (black in the figure) contains all intervals that fit the specification given by means of an earliest start, latest start, earliest end, latest end, and minimum and maximum durations.

AsbruView's Time Annotation Glyph

This diagram is a part of AsbruView [9], which is a user interface for the plan representation language Asbru. The glyph[1] described here is an extension of LifeLines and was specifically designed to represent Asbru's time annotation.

Basic Principle

It is based partly on a simple metaphor (Figure 3). Four vertical "pillars" along the time axis represent the earliest and latest starting and ending times. On top of these vertical bars lies a bar that is as long as the maximum duration. On top of the MaxDu bar, supported by two diamonds (which lie above the LSS and EFS supports, see Figure 3), lies another bar representing the minimum duration. Undefined parts of a time annotation are displayed in gray, and in addition the diamonds supporting the MinDu bar become "rolls".

It is possible to understand a few simple constraints based on this metaphor. One is that the minimum duration can never be shorter than the difference between LSS and EFS — if it was, the MinDu bar would fall down between its supports. If LSS or EFS are undefined, the corresponding diamonds become rolls, which means that the implicitly defined EFS moves if the MinDu changes, for example.

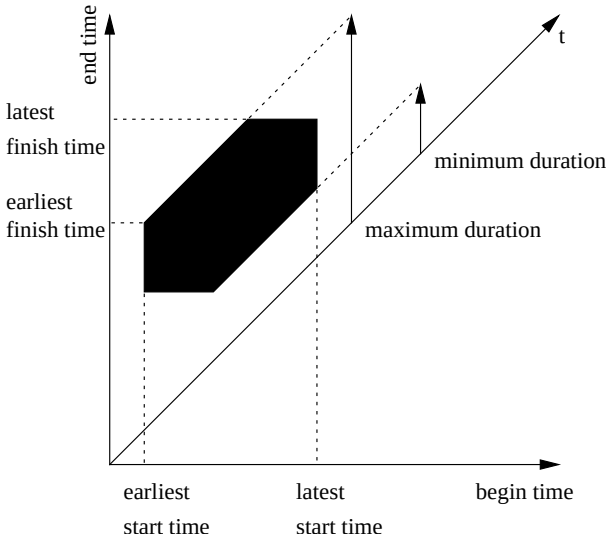

Figure 2- Sets of Possible Occurrences (SOPOs)

On the right side of Figure 3, the depiction of different temporal scales can be seen. If the current scale is coarser than the precision to which the time points were defined, a circle appears (similar to the notion of an open interval in mathematics, where the end point of the interval "from 1 to, but not including, 2" cannot be depicted directly). On a smaller scale, a zigzag line appears that covers the area of "additional uncertainty" due to the lower resolution of the actual point in time.

User Study

We implemented the described method in a prototype of a more general user interface called AsbruView (Figure 4). To assess its usefulness in practice, we performed a user study with six physicians [9] to test the glyph (and also other parts of AsbruView). For this study, we let the participants perform simple tasks with the program, after having given them a short (ca. 30min) introduction to Asbru and AsbruView. The participants were asked to fill out questionnaires before and after the test to find out how they judged the system.

The results were quite satisfying. The participants immediately understood the (rather complex) time annotation explained in terms of the metaphor underlying the time annotation glyph. They were able to specify time annotations for their own use, and understood the meaning of undefined values.

[1] A glyph is a graphical object (often vaguely representing a real object, like a face) whose features express the values of certain attributes that are to be shown [10, 11].

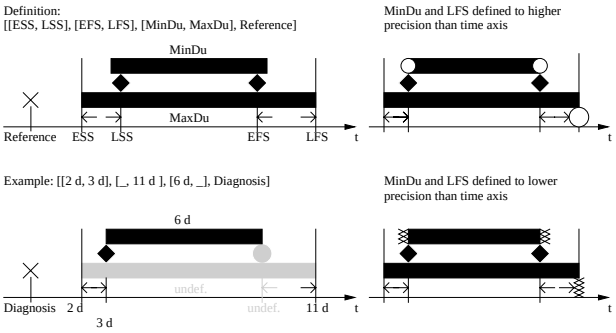

Figure 3- Time Annotation Glyphs

Discussion

Even if only a part of the functionality of the time annotation is used in each temporal pattern or action specification, none of its capabilities can be ignored.

Lifelines clearly do not satisfy the requirements for displaying time annotations. They cannot represent uncertainty or even undefined values. As Rit points out [8], this is simply due to the fact that they are one-dimensional, so any information exceeding one-dimensional time results in an ambiguous diagram. For recorded patient information, however, they are very useful. This mostly includes information about events, rather than recorded device readings (but a combination of both is easily imaginable).

SOPOs were designed for the easy graphical propagation of temporal constraints, not for making a complex notion of time easy to understand. Specifically, parallel plans and hierarchical decomposition are very hard to depict and to work with (if plans from several levels are drawn into the same diagram, their relationship is not immediately visible; parallel plans cover the same area in the diagram). A notion of undefined parts is missing in the original design.

We extended SOPOs so that they met almost all the requirements described at the beginning of this paper [12]. In a usability study we performed with these extended SOPOs, one of the results was that this type of diagram is hard to understand for people outside of computer science (and even many within ...).

AsbruView's time annotation glyph was specifically designed to meet the requirements specified at the beginning of this paper, so it does satisfy all of them. The simple metaphor it is based on has proven to be very helpful, without it limiting the representative power of the glyph (which can happen if a glyph is too tightly coupled to a metaphor: then, the constraints of the metaphor become constraints of the glyph. This is of course very undesirable).

Other methods like Gantt and PERT charts are quite common in many areas, including medicine. Gantt charts are similar to LifeLines, but can display hierarchies, and it is easy to imagine how to add facets (similar to LifeLines) to a Gantt chart display.

PERT charts, on the other hand, do not show any precise temporal information, only the order in which actions should occur. This can be useful, of course, especially in early stages of planning. But the approach is quite limited, especially when dealing with such complex things as treatment protocols.

This discussion is summarized in Table 1.

Table 1 – Comparison of Time Visualization Methods

Method	Allen's Relations	Temporal Uncertainty	Undefined Parts	Resolution	Hierarchical Decomposition	Facets
TimeLines	●					●
Gantt	●				●	
PERT					●	
SOPOs	●	●				
Extended SOPOs	●	●	●	●	●	
Time Ann. Glyph	●	●	●	●	●	●

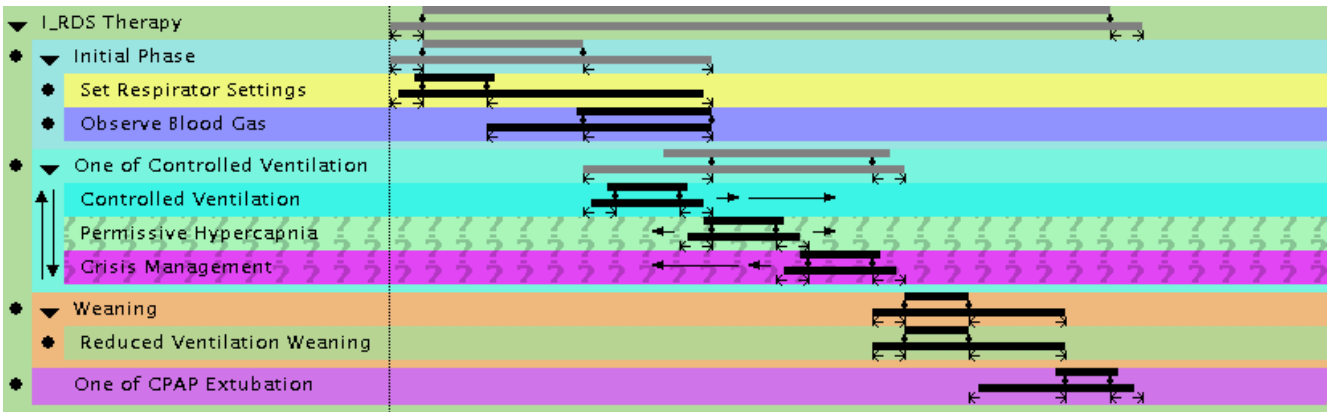

Figure 4- A screenshot of a part of the AsbruView prototype showing a real therapy plan in terms of time annotation glyphs

Conclusions and Future Work

Visualization supports complex tasks such as medical treatment planning. Simple one-dimensional time representations are not powerful enough for this task, however.

We presented a method for visualizing such complex notions of time and showed that it is useful in practice by performing a small user study.

A few open problems remain. One is the problem of how to represent cyclical as well as other types of events. This is made complicated through the fact that cyclical events might be specified to be performed until a certain state is reached. In such a case, the temporal extent of the action is not known (not even possible intervals), and a way of depicting this would have to be found.

Another problem is that of abstract reference points. If a point in time (like date of birth) is not (yet) known, nothing can be drawn. Only when the reference point is bound to an absolute point in time can the interval be drawn relative to other intervals. The question is, how such a time annotation could be specified graphically and depicted before this is possible.

There are also other types of uncertainty than the one addressed in this paper. If only the set of actions to be performed is known, how can it be made clear that the order in which they are depicted does not prejudge their actual order of execution?

A related problem is that of optional actions or plans. If a plan does not have to be performed (but can, depending on conditions), it must be depicted in a different way than an obligatory one.

Acknowledgements

We would like to thank the participants in the user study (in alphabetic order and ignoring their titles): Shahram Adel, Sophie Brandstetter, Maria Dobner, Gerhard Miksch, Franz Paky, and Christian Popow.

The Asgaard Project is supported by “Fonds zur Förderung der wissenschaftlichen Forschung” (Austrian Science Fund), grant P12797-INF.

References

[1] Seyfang S, Kosara R, and Miksch S. Asbru Reference Manual. Vienna University of Technology, Institute of Software Technology, Vienna, Technical Report, Asgaard-TR-2000-3, 2000.

[2] Miksch S, Shahar Y, and Johnson P. Asbru: A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans. In *Proceedings of the 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97)*. Milton Keynes, UK, Open University, 1997.

[3] Ware C. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers, 2000.

[4] Allen J. Maintaining Knowledge about Temporal Intervals. *CACM* 1983: 26(11):832–843.

[5] Plaisant C, Milash B, Rose A, Widoff S, and Shneiderman B. LifeLines: Visualizing Personal Histories. In *Proceedings of ACM CHI 96 Conference on Human Factors in Computing Systems*, volume 1 of *PAPERS: Interactive Information Retrieval*, pages 221–227, 1996.

[6] Tufte ER. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press, 1983.

[7] Brandt CA, Frawley SJ, Powsner SM, Shiffman RN, and Miller PL. Visualizing the Logic of a Clinical Guideline: A Case Study in Childhood Immunization. *Meth Inform Med* 1997: 36:179–83.

[8] Rit J-F. Propagating temporal constraints for scheduling. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 383–388. Morgan Kaufman Publishers, Inc, 1986.

[9] Kosara R and Miksch S. Metaphors of Movement: A Visualization and User Interface for Time-Oriented, Skeletal Plans. *Art Int Med*, special issue on *Information Visualization in Medicine*. Forthcoming, 2001.

[10] Chernoff H. The use of faces to represent points in k-dimensional space graphically. *J Am Stat Ass* 1973: 68:361–368.

[11] Chuah MC and Eick SG. Glyphs for Software Visualization. In *5th International Workshop on Program Comprehension (IWPC '97) Proceedings*, pages 183–191. Dearborn, Michigan: IEEE Computer Society Press, 1997.

[12] Messner P. Time Squares: A Two-Dimensional Representation of Temporal Aspects in Skeletal Plans — Evaluation of Different Approaches. Master's thesis, Vienna University of Technology, Vienna, Austria, 2000.

Address for correspondence

Robert Kosara, Institute for Software Technology, Vienna University of Technology, Favoritenstraße 9–11/E188, A-1040 Vienna, Austria. Email: rkosara@ifs.tuwien.ac.at.

`http://www.ifs.tuwien.ac.at/~rkosara`,
`http://www.ifs.tuwien.ac.at/asgaard/`.