Web services for time granularity reasoning*
Extended Abstract

Claudio Bettini
DICO - University of Milan,
via Comelico 39, I-20135 Milan, Italy
bettini@dico.unimi.it

Abstract

We first briefly illustrate the concept of time granularity and review the emerging approaches for modeling and reasoning with it. We then advocate the need for a set of web services that distributed applications can use to define and manipulate time granularities. As an example of these services we discuss in detail the GSTP service, a constraint solver for networks of temporal constraints with time granularities, developed at the University of Milan, and based on recently published theoretical results. The GSTP service can be accessed from applications through the Internet as a web service, or can be accessed by a human user through a sophisticated java interface. A system demonstration will be given during the talk.

1 Modeling and reasoning with time granularity

Any time granularity can be viewed as the partitioning of a time domain in groups of elements, where each group is perceived as an indivisible unit (a granule). The description of a fact, action, or event can use these granules to provide them with a time qualification, at the appropriate abstraction level. Hence, from the standpoint of representation formalisms, time granularity can be defined as the resolution power of the time qualification of a statement. For the same reason time granularities have been historically devised to model astronomical (e.g., year, day, month) as well as social and economic phenomena (e.g, business day, trading day, weekends, academic semesters). Reflecting the complexity of these phenomena, the relationships among time granularities are not always straightforward. Moreover, some of the granularities have the same name but different semantics in countries adopting different calendars or even when used by different organizations (e.g., business days may vary depending on the country but also on specific companies).

The naive approach commonly used in computer science is simplifying the domain representation forcing a conversion into a standard date format and/or to a single time granularity. Alternatively, custom solutions are built into the applications. However, a lot of work has been done recently to formalize the concept of time granularity, to identify relationships, and to reason with non-standard time granularities. Despite the literature on time granularity is quite rich, we identify two main approaches: a temporal logic approach [6] and a set-theoretic approach [2]. The logic approach is focused on embedding the notion of granularity in a temporal logic in order to be able to qualify logic statements according to different granularities, and to relate statements qualified in terms of different granularities. Technically, multi-layered metric temporal logics have been proposed and extensively studied.

The set-theoretic approach is focused on providing a mathematical characterization of time granularities, including arbitrary user-defined time granularities as academic semesters, banking days, or business weeks, and a formal characterization of their relationships. While a time granule can be defined as an indivisible set of elements of an underlying time domain (usually modeled by $\mathbb{R}$ or $\mathbb{Z}$), a large class of time granularities can be modeled as a periodic set of such granules, with some technical restrictions and generalizations that we leave for the interested reader [2]. Starting from these formal notions, from their properties, and from a set of algebraic operations relating granules of different granularities, a number of applications have been investigated, from the design of multi-granularity temporal databases, to temporal data mining, from querying databases storing data in different granularities to constraint satisfaction problems involving multi-granularity temporal constraints.

*This work has been partially supported by Italian MIUR (FIRB "Web-Minds" project).

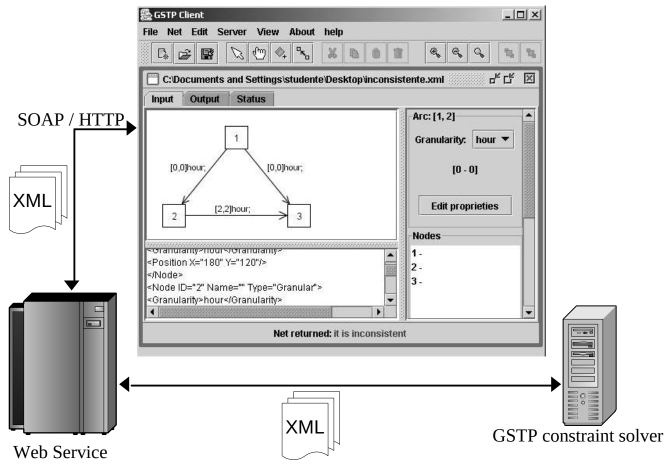

Figure 1. The GSTP Architecture

2 Granularity web services

The set-theoretic approach naturally leads to the definition of a set of basic services, regarding the specification and manipulation of time granularities, that may be useful in a wide range of applications. Since the need for multi-granularity reasoning naturally arises in applications involving processes and actors spread in different sites if not in different world regions, it is quite natural to envision a distributed architecture based on web services. An essential requirement is a common representation for time granularities in the implementation of each service. Along these lines we are working at three specific services:

- *User-friendly specification of new time granularities*. It is often the case that a granularity can be easily specified based on another granularity, for example by grouping its granules according to a specific pattern. This service should offer access to repositories of granularities and allow the application of algebraic operations to create new granularities from existing ones.

- *Granularity conversion services*. This service should be able to answer queries like “Give me the day in date format (yyyy-mm-dd) corresponding to the second NYSE trading day of 2004”. We currently have implemented a first web service prototype performing this task and the previous one; However, currently it is a simple adaptation of a client-server application originally developed at George Mason University, implementing the ideas in [7].

- *Multi-granularity constraint satisfaction*. This service should be able to solve instances of the Simple Temporal Problem [4] where each constraint is given in terms of a specific time granularity (e.g., “the time distance between shipment and delivery of a package must be less than two business days”).

In the following we briefly illustrate the third service.

3 The GSTP system

The GSTP system has been developed at the University of Milan with the objective of providing universal access to the implementation of a set of algorithms for multi-granularity temporal constraint satisfaction. The many formalisms and algorithms proposed in the literature for Temporal Constraint Satisfaction Problems (TCSP) have essentially ignored the subtleties involved in the presence of multiple time granularities in the temporal constraints. . The GSTP system allows the user to specify binary constraints of the form $Y - X \in [m, n]\, G$, where m and n are the minimum and maximum values of the distance between X and Y in terms of granularity G. Variables take values in the positive integers, and unary constraints can be applied on their domains. This can be considered the extension of STP [4] to multiple and arbitrary granularities.

GSTP adopts the set-theoretic time granularity formalism, first introduced in [8, 2]. In order to guarantee a finite representation, granularities in GSTP are limited to those that can be defined in terms of periodic sets. Hours, days, weeks, business days, business weeks, fiscal years, and academic semesters are common examples.

It is not an easy task to reduce a network of constraints given in terms of different granularities into an equivalent one with all constraints in terms of the same granularity, so that some of the standard algorithms for CSP, like consistency checking through arc- or path-consistency [5, 1, 4], could be successfully applied. Indeed, any conversion necessarily introduces an approximation; For example, a constraint imposing that the shipment of a package should start the next business day after payment may be translated in terms of hours with a minimum of 1 hour and a maximum of 95 hours. (The number 1 takes into account the possibility of a payment just before midnight on Tuesday, for example, and a shipment just after midnight, while the number 95 takes into account a payment at the beginning of a Friday and a shipment at the end of next Monday.) However, if the payment is actually done on Monday, the new constraint, considering the 95 hours, would allow a shipment on Thursday which is clearly a violation of the original constraint. Approximate conversion algorithms are discussed in [2]. We have shown that any consistency algorithm adopting these conversions as the only tool to reduce the problem to a standard CSP is inevitably incomplete, and have proposed a different algorithm, called ACG, which has been proved to be complete [3].

GSTP, in addition to implementing the reasoning algorithms, assists the user in the definition of constraint networks, in their submission to a remote processing service and in the analysis of the output.

3.1 The GSTP Architecture

Fig. 1 shows the general architecture of the GSTP system. There are three main modules: the constraint solver, the web service, which enables external access to the solver, and a user interface that can be used locally or remotely to design and analyze constraint networks.

The constraint solver is the C implementation of the algorithms described above, and it runs on a server machine. The Web Service defines, through a WSDL specification, the parameters that can be passed to the constraint solver, including the XML schema for the constraint network specification; It accepts connections through soap/http from client applications or other web services which require constraint processing, it invokes the solver after validating the parameters, and it passes back the results. The web service can be used by any scheduling or planning application that needs to solve multi-granularity STPs. However, we do provide a remote java-based user interface as a third module of the GSTP system. The interface allows the user to easily edit constraint networks, to submit them to the constraint solver, and to analyze results. In particular, it is possible to have views in terms of specific granularities, to visualize implicit constraints, to browse descriptions of domains, and to obtain a network solution. Fig. 2 shows a screenshot from the interface.

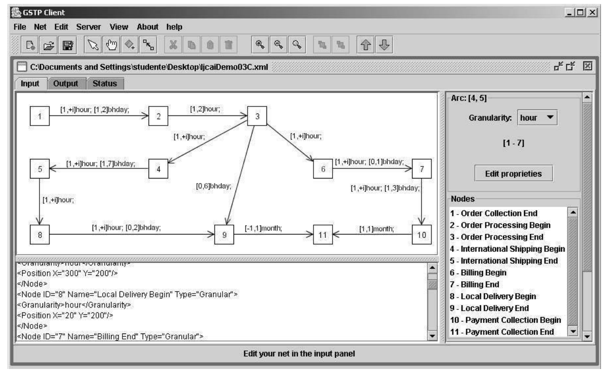

Figure 2. The User Interface

References

[1] C. Bessière. Arc-Consistency and Arc-Consistency Again. *Artificial Intelligence* 65(1):179–190, 1994.

[2] C. Bettini, S. Jajodia, and X. Wang. *Time Granularities in Databases, Temporal Reasoning, and Data Mining*. Springer, 2000.

[3] C. Bettini, X. Wang, S. Jajodia, Solving Multi-Granularity constraint networks, *Artificial Intelligence*, 140(1-2):107–152, 2002.

[4] R. Dechter, I. Meiri, J. Pearl. Temporal constraint networks. *Artificial Intelligence* 49:61–95, 1991.

[5] A. Mackworth, E. Freuder. The complexity of some polynomial network consistency algorithms for constraint satisfaction problems. *Artificial Intelligence* 25:65–74, 1985.

[6] A. Montanari. *Metric and Layered Temporal Logic for Time Granularity*. PhD Thesis, University of Amsterdam, Amsterdam (NL), September 1996, ILLC Dissertation Series 1996-02.

[7] P. Ning, X. Wang, S. Jajodia. An Algebraic Representation of Calendars. *Annals of Mathematics and Artificial Intelligence* 36(1-2): 5-38, 2002.

[8] X. Wang, C. Bettini, A. Brodsky, S. Jajodia. Logical design for temporal databases with multiple granularities. *ACM Transactions on Database Systems* 22(2):115–170, 1997.