Web Services for Time Granularity Reasoning

Claudio Bettini
Data, Knowledge and Web Engineering Laboratory
University of Milan, Italy

TIME-ICTL 03 Invited Talk

Overview

- Time granularities
- Current approaches to time granularities
- Time granularity web services
- GSTP: the Granularity Simple Temporal Problem
- The GSTP web service

Time granularities

What is a time granularity?

- A granularity can be viewed as the partitioning of a temporal domain in groups of elements, where each group is perceived as an indivisible unit (a granule).

What is a time granularity?

- A granularity can be viewed as the partitioning of a temporal domain in groups of elements, where each group is perceived as an indivisible unit (a granule).
- The set of granules can be used for temporal qualification of statements (e.g., *It has been raining for two weeks* or *Revenues in July 2003 are $500.000*) and for specifying temporal relationships (e.g., *The merchandise should reach its destination in 3 business days from the date of its order.*)

What is a time granularity?

- A granularity can be viewed as the partitioning of a temporal domain in groups of elements, where each group is perceived as an indivisible unit (a granule).
- The set of granules can be used for temporal qualification of statements (e.g., *It has been raining for two weeks* or *Revenues in July 2003 are $500.000*) and for specifying temporal relationships (e.g., *The merchandise should reach its destination in 3 business days from the date of its order.*)
- It is a powerful abstraction tool

Which time granularities should we model?

Not only the usual ones: *hour*, *day*, *week*, *month*, *year*, ...

Which time granularities should we model?

Not only the usual ones: *hour*, *day*, *week*, *month*, *year*, ...

- From Ameritrade policies:
  GTC orders are only open for the traditional trading session each *trading day*.

Which time granularities should we model?

Not only the usual ones: *hour*, *day*, *week*, *month*, *year*, ...

- From Ameritrade policies:
  GTC orders are only open for the traditional trading session each *trading day*.
- From UC Berkeley: Students wanting to do a part-time internship during an *academic semester* must be enrolled at UC Berkeley for the semester of their internship

Which time granularities should we model?

Not only the usual ones: *hour, day, week, month, year, . . .*

- From Ameritrade policies:
  GTC orders are only open for the traditional trading session each *trading day*.
- From UC Berkeley: Students wanting to do a part-time internship during an *academic semester* must be enrolled at UC Berkeley for the semester of their internship
- From Morgan Stanley Mutual Fund Prospects:
  By *business day* we mean any day in which Banks are open in Luxemburg, New York and Tokyo depending on the specific case.

Time granularities can have complex definitions

Trading, banking and business days are examples of granularities that may depend on local holidays (e.g., California Admission Day).

Time granularities can have complex definitions

Trading, banking and business days are examples of granularities that may depend on local holidays (e.g., California Admission Day).

From NYSE web site:
"If an observable holiday occurs on a Sunday it is observed on Monday, If on Saturday, it is observed on Friday. The exception to this rule is New Years. If it falls on a Saturday, the Market will be open on Friday, as the NYSE is ALWAYS open on the last trading day of the year."

Time granularity research issues

- Modeling time granularities
- Granule conversions
- Conversion of information in terms of different granularities
- Time granularity constraint reasoning

Current approaches to time granularities

Current approaches to time granularities

- *The multilayered logic approach.*
  Goal: a full-fledged logic to reason about multi-granularity temporally qualified statements.

Current approaches to time granularities

- *The multilayered logic approach.*
  Goal: a full-fledged logic to reason about multi-granularity temporally qualified statements.

- *The set-theoretic approach.*
  Goal: a mathematical model for arbitrary granularities and an associated algebra to manipulate them, plus a set of domain-specific reasoning tools.

The multilayered logic approach

- Extensions of topological temporal logic
- Temporal universe consisting of a (possibly infinite) set of inter-related differently-grained temporal domains
- Logical tools are provided to qualify temporal statements with respect to the temporal universe and to switch temporal statements across temporal domains.
- Main application: real time systems specification and verification

The multilayered logic approach

- Extensions of topological temporal logic
- Temporal universe consisting of a (possibly infinite) set of inter-related differently-grained temporal domains
- Logical tools are provided to qualify temporal statements with respect to the temporal universe and to switch temporal statements across temporal domains.
- Main application: real time systems specification and verification

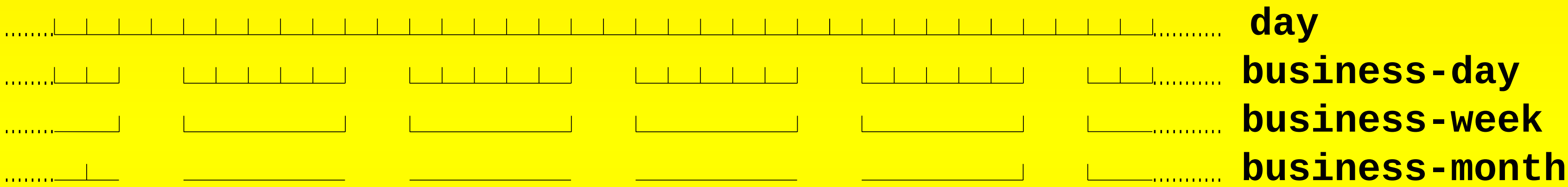
Angelo Montanari. *Metric and layered temporal logic for time granularity.* PhD thesis, University of Amsterdam. ILLC DIssertation Series 1996-02.

Time granularities in the set-theoretic approach

A *granularity* is a mapping G from the positive integers to $2^{\mathcal{T}}$ (i.e., all subsets of a lineraly ordered temporal domain) such that for all positive integers i and j with $i < j$, the following two conditions are satisfied:

- $G(i) \neq \emptyset$ and $G(j) \neq \emptyset$ imply that each element in $G(i)$ precedes all elements in $G(j)$, and
- $G(i) = \emptyset$ implies $G(j) = \emptyset$.

Examples of time granularities



Example of granules:

day(1) = 0001/01/01

day(731405) = 2003/07/09

The set theoretic approach

- Many granularity relationships formally defined (groups-into, finer-than, periodically groups-into, sub-granularity, shifting equivalent, ...)
- Different time granularity systems have been investigated identifying systems with nice properties (e.g., lattices).

The set theoretic approach

- Many granularity relationships formally defined (groups-into, finer-than, periodically groups-into, sub-granularity, shifting equivalent, ...)
- Different time granularity systems have been investigated identifying systems with nice properties (e.g., lattices).

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

System representations of time granularities

Assume `hour` is the bottom granularity with `hour`(1) mapped to the instants corresponding to 2001/1/1:01, then:

- `monday` can be represented in terms of `hour` by:
  Period $P = 168$;
  Description of one of the periods: $\{[1, 24]\}$
  Bounds: none.

System representations of time granularities

Assume `hour` is the bottom granularity with `hour`(1) mapped to the instants corresponding to 2001/1/1:01, then:

- `monday` can be represented in terms of `hour` by:
  Period $P = 168$;
  Description of one of the periods: $\{[1, 24]\}$
  Bounds: none.

- `business-day-until-2003` can be represented by:
  Period $P = 168$;
  Description of one of the periods:
  {[1,24][25,48][49,72][73,96][97,120]}
  Bounds: Up=775 (number of business days from 2001 to 2003).

Primitive granule conversion operations

If $\mathtt{day}(1)$ is January 1st 2001, then:

- $\lceil 33 \rceil^{\mathtt{month}}_{\mathtt{day}} = 2$ since Feb. 2nd 2001, represented in the system as $\mathtt{day}(33)$, is contained in Feb. 2001, the second month, represented in the system as $\mathtt{month}(2)$.

Primitive granule conversion operations

If $\mathtt{day}(1)$ is January 1st 2001, then:

- $\lceil 33 \rceil^{\mathtt{month}}_{\mathtt{day}} = 2$ since Feb. 2nd 2001, represented in the system as $\mathtt{day}(33)$, is contained in Feb. 2001, the second month, represented in the system as $\mathtt{month}(2)$.
- $\lfloor 2 \rfloor^{\mathtt{month}}_{\mathtt{day}} = \{[32, 59]\}$ since February 2001 contains the 28 days indexed from 32 to 59 in the granularity system.

Primitive granule conversion operations

If $\mathtt{day}(1)$ is January 1st 2001, then:

- $\lceil 33 \rceil^{\mathtt{month}}_{\mathtt{day}} = 2$ since Feb. 2nd 2001, represented in the system as $\mathtt{day}(33)$, is contained in Feb. 2001, the second month, represented in the system as $\mathtt{month}(2)$.
- $\lfloor 2 \rfloor^{\mathtt{month}}_{\mathtt{day}} = \{[32, 59]\}$ since February 2001 contains the 28 days indexed from 32 to 59 in the granularity system.
- $\lfloor 2 \rfloor^{\mathtt{b\text{-}month}}_{\mathtt{day}} = \{[32, 33][36, 40][43, 47][50, 54][57, 59]\}$ since the second business month includes only the days of February 2001 which are not Saturday nor Sunday.

Investigated applications

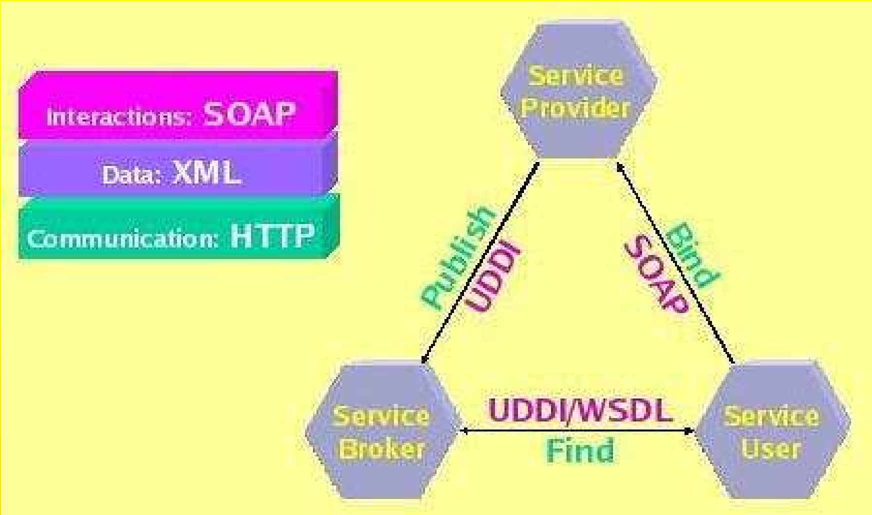
- Databases: temporal query processing and temporal database interoperability, temporal database design, multiple granularity integrity constraint checking;
- Data mining: discovering frequent temporal patterns, discovering temporal relationships;
- Artificial Intelligence: multi-granularity constraint processing, scheduling in inter-organizational workflows.

The set-theoretic model and its basic services were a common basis in the proposed solutions (but different techniques were applied).

Time granularity web services

Web services

- A technology to enable distributed Web applications
- Based on: HTTP, SOAP, WSDL, UDDI
- Language and platform independent



Why time granularity web services?

- Managing distributed repositories of XML time granularity specifications
- Offering processing services to web applications. For example:
  - Specification of new granularities (using common algebra operators)
  - Search for equivalent specifications (name clash problem)
  - Conversion of granules
  - Constraint processing
  - ...

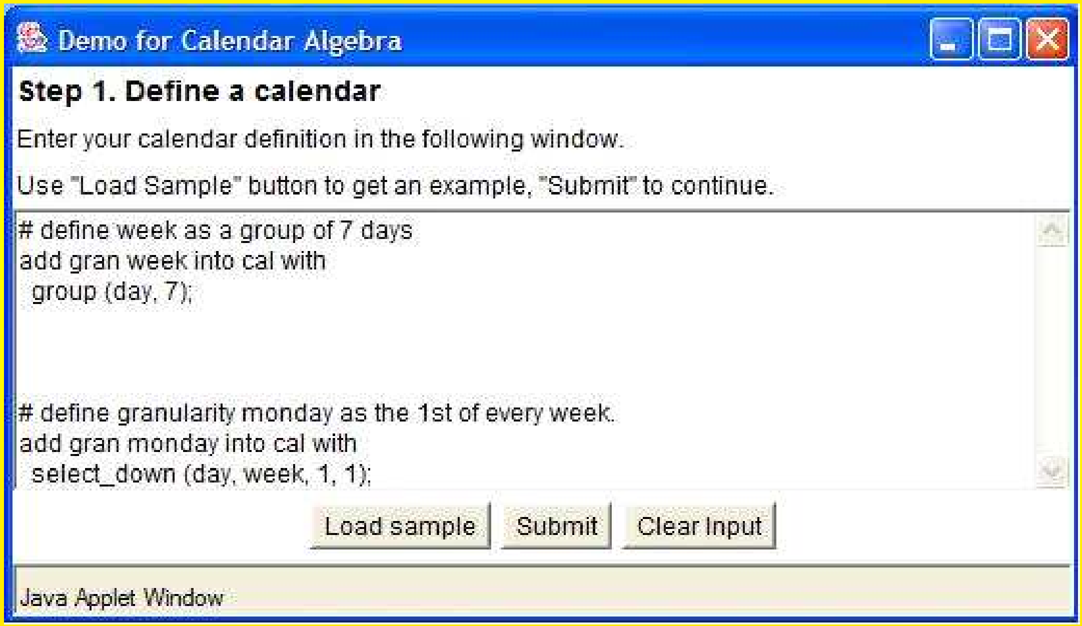
Which applications

Essentially all applications requiring multi-granularity time-management.

Focus on some of them:

- Inter-organizational workflows
- Personal (or group) Information Management (e.g., different views in Outlook-like apps, appointment scheduling)
- Medical applications (e.g., monitoring)
- ...

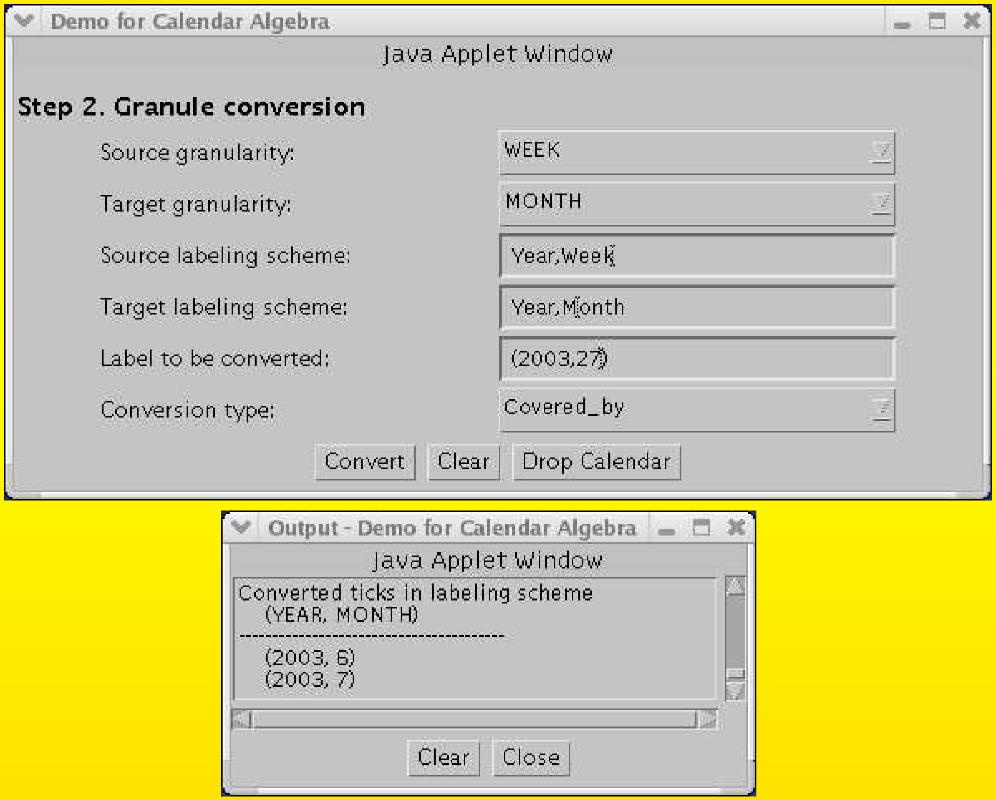
Granularity specification web services



Screenshot of a client-server application developed at GMU.

Granule conversion web services

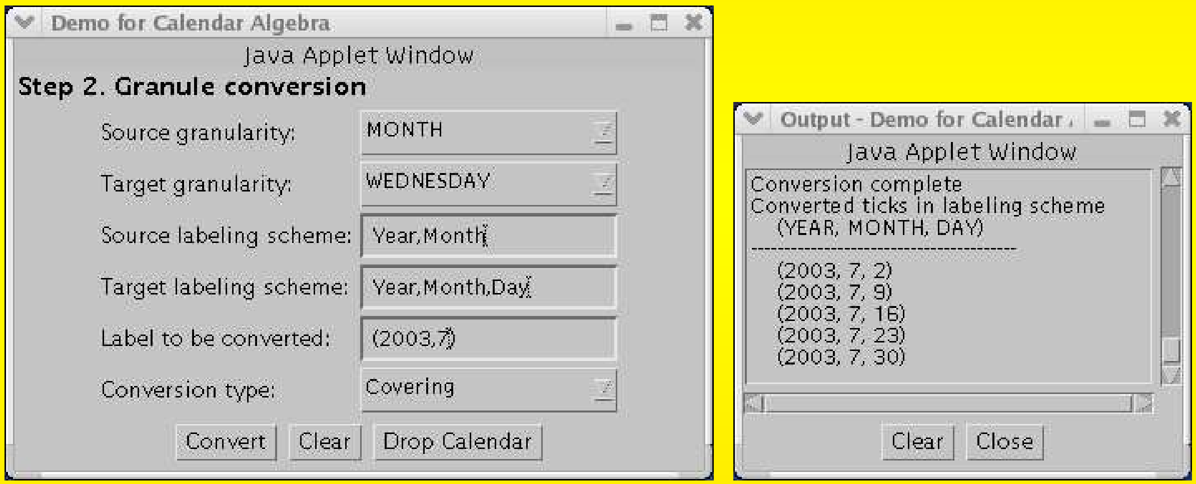
Find the month that contains the 27th week in 2003.



Screenshot of a client-server application developed at GMU.

Granule conversion web services (2)

Find all the Wednesdays in July 2003.

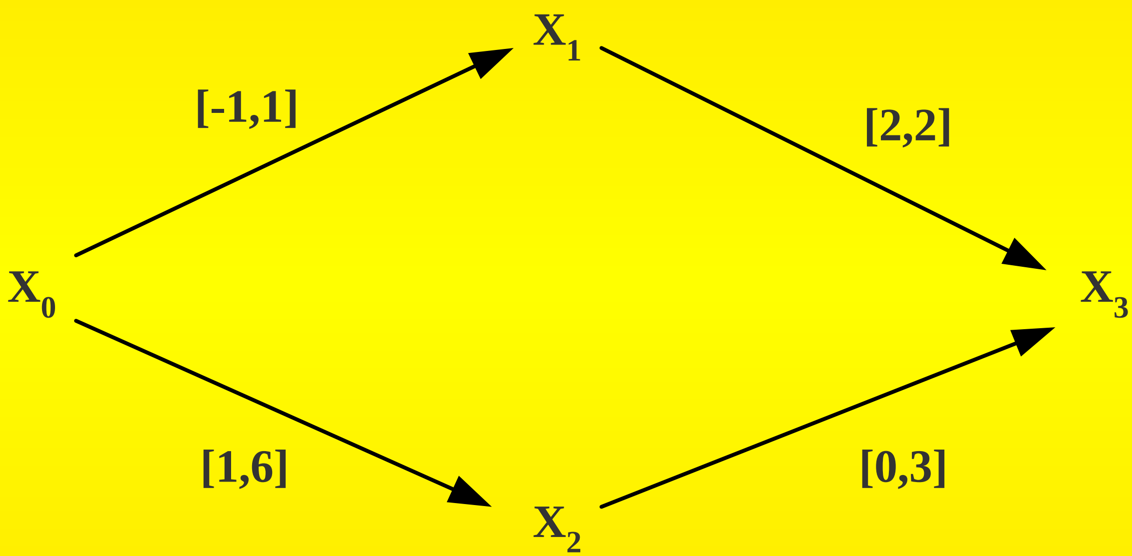


P. Ning, X. Wang, S. Jajodia. An Algebraic Representation of Calendars. *Annals of Mathematics and Artificial Intelligence* 36(1-2): 5-38, 2002.

GSTP: the Granularity Simple Temporal Problem

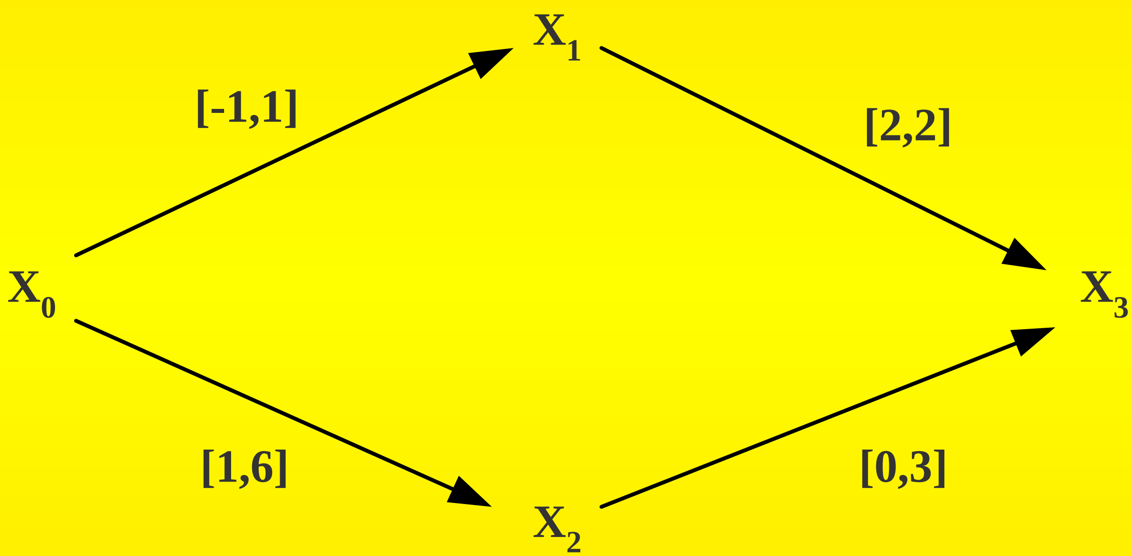
(The theory underlying a constraint processing web service)

Temporal constraint networks



A Simple Temporal Problem (STP).

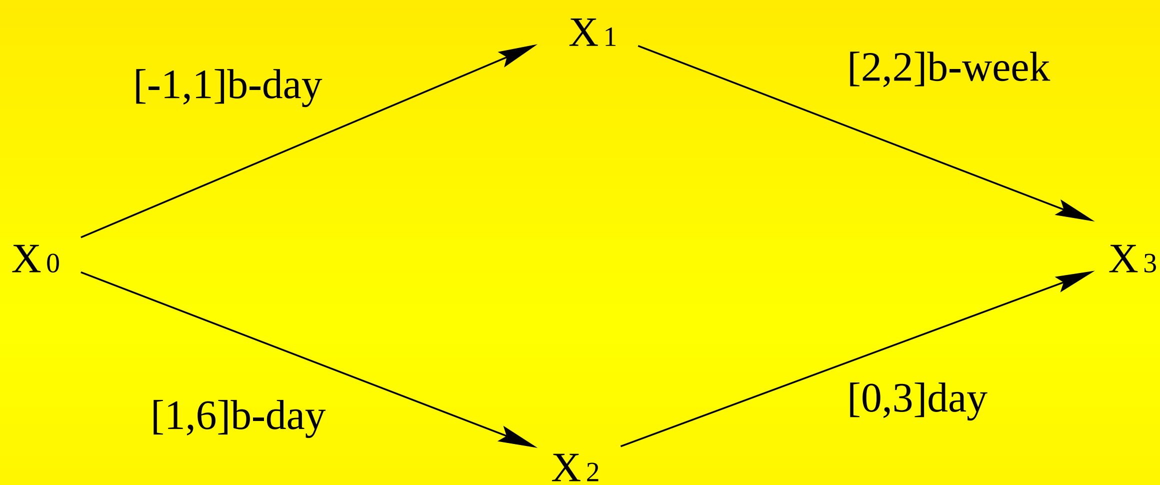
Temporal constraint networks



A Simple Temporal Problem (STP).

Main tasks: consistency, constraint refinement, solution

Temporal constraint networks with granularities



Variables take values in $\mathbb{Z}^+$.

(x_0, x_1) satisfies $[-1, 1]\texttt{bday}$ iff
(1) $\lceil x_0 \rceil^{\texttt{bday}}$ and $\lceil x_1 \rceil^{\texttt{bday}}$ are both defined, and
(2) $-1 \leq (\lceil x_1 \rceil^{\texttt{bday}} - \lceil x_0 \rceil^{\texttt{bday}}) \leq 1$

The intuitive approach

Convert the network constraints in terms of a single granularity, and apply known (polynomial time) algorithms for STP.

The intuitive approach

Convert the network constraints in terms of a single granularity, and apply known (polynomial time) algorithms for STP.

Unfortunately, there is no straightforward reduction. Consistency is NP-hard in terms of the involved granularities [Bettini et al. TIME96 and AMAI98].

An approximate algorithm: Conversion+PC

- Compute the tightest implied constraints in terms of each granularity, generating a set of networks
- Process each network with known algorithms
- Rejoin the networks and repeat from step 1 until a fixpoint is reached

The conversion problem

$[1, 1]\,\texttt{bday} \longrightarrow [?, ?]\,\texttt{day}$

$[1, 1]\,\texttt{bday} \longrightarrow [?, ?]\,\texttt{hour}$

The conversion problem

$[1,1]\,\texttt{bday} \rightarrow [?,?]\,\texttt{day}$ $\qquad$ $[1,1]\,\texttt{bday} \rightarrow [?,?]\,\texttt{hour}$

Why not $[1,1]\,\texttt{day}$?

What about one event on Friday and the other on Monday?

The conversion problem

$[1, 1]\,\texttt{bday} \rightarrow [?, ?]\,\texttt{day}$

$[1, 1]\,\texttt{bday} \rightarrow [?, ?]\,\texttt{hour}$

Why not $[1, 1]\,\texttt{day}$?
What about one event on Friday and the other on Monday?

Why not $[1, 100]\,\texttt{hour}$?
It is *implied* but a tighter implied constraint exists: $[1, 95]\,\texttt{hour}$.

The conversion problem (2)

$$[\mathbf{1},\mathbf{1}]\,\texttt{bday} \quad \rightarrow \quad [\mathbf{1},\mathbf{95}]\,\texttt{hour}$$

Can $[1, 95]\,\texttt{hour}$ *substitute* $[1, 1]\,\texttt{bday}$?
(2003/7/9:14,2003/7/12:14) satisfies $[1, 95]\,\texttt{hour}$
but violates $[1, 1]\,\texttt{bday}$

The conversion problem (2)

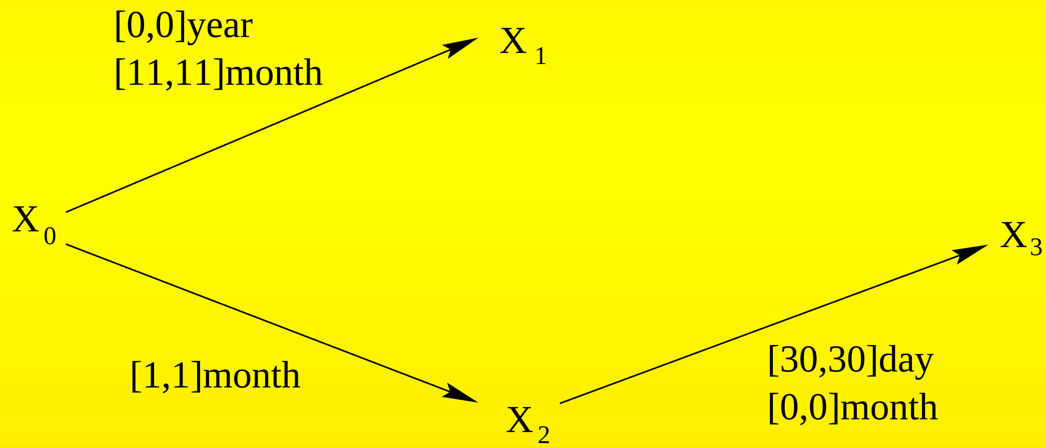
$$[\mathbf{1},\mathbf{1}]\,\texttt{bday} \quad\rightarrow\quad [\mathbf{1},\mathbf{95}]\,\texttt{hour}$$

Can $[1,95]\,\texttt{hour}$ *substitute* $[1,1]\,\texttt{bday}$?
(2003/7/9:14,2003/7/12:14) satisfies $[1,95]\,\texttt{hour}$
but violates $[1,1]\,\texttt{bday}$

Converting constraints is tricky!
New algorithms presented in [Bettini, Ruffini AAAI-WS02 and JUCS].

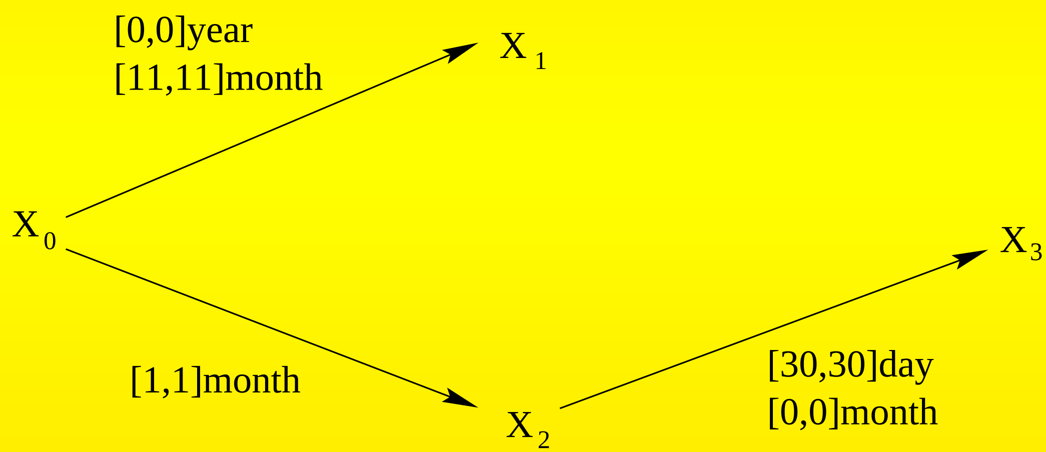
Why incomplete?

Consider this example with only “standard” granularities:



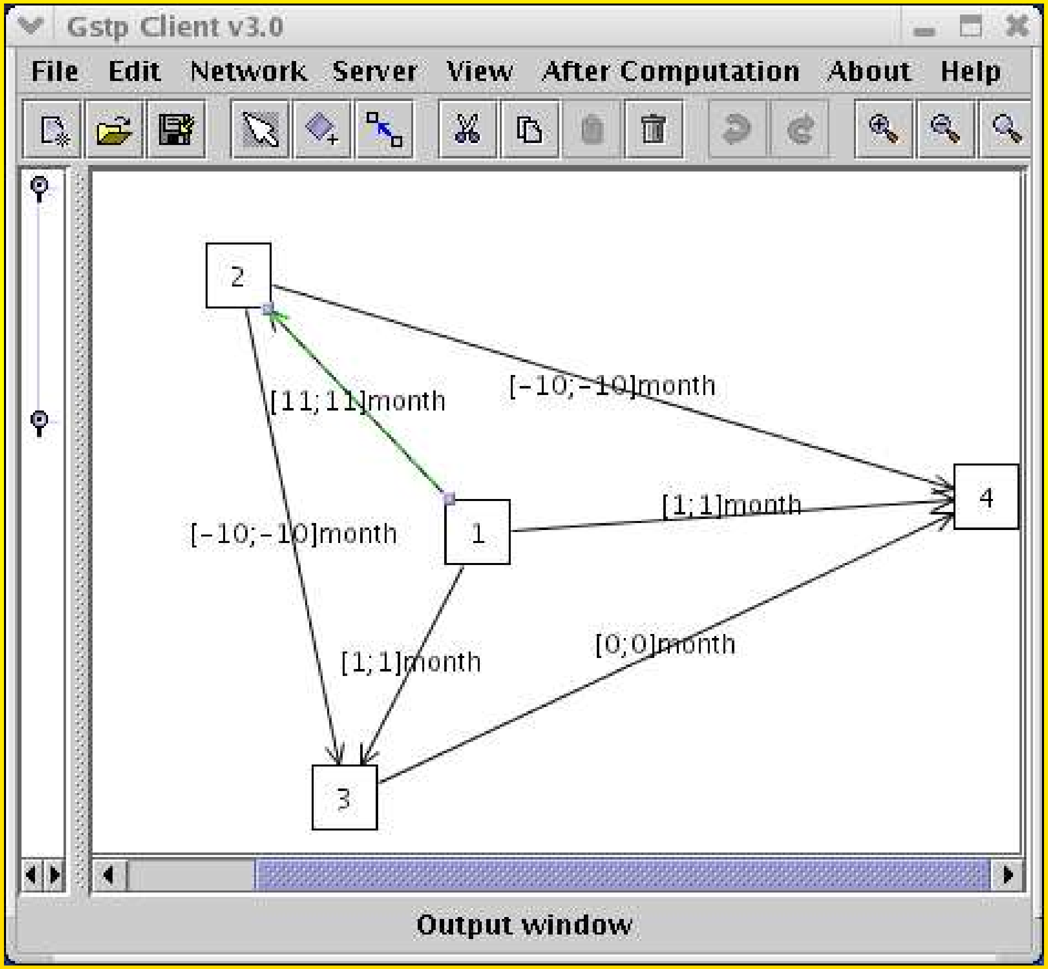
Why incomplete?

Consider this example with only "standard" granularities:

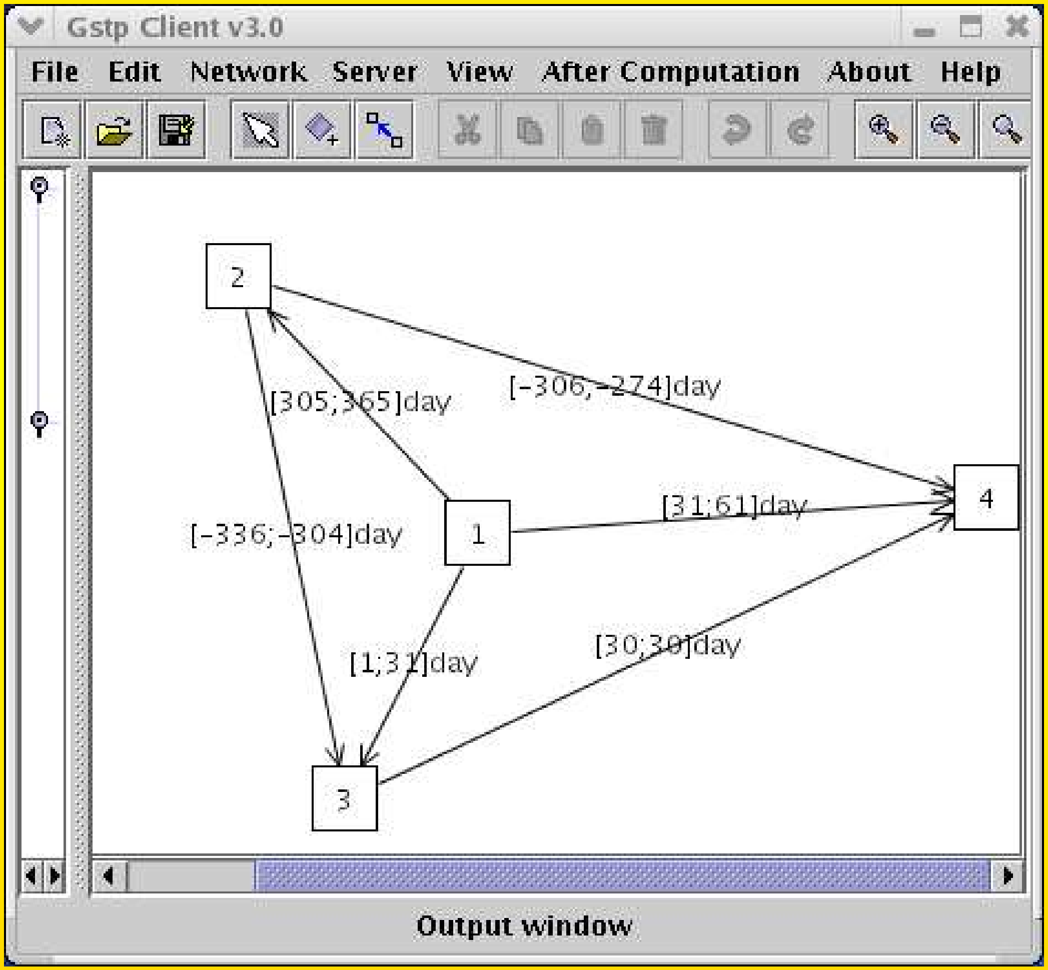


Intuition: We have to take into account the domain of involved variables

A minimal network in terms of `month`



A minimal network in terms of day



A sound and complete algorithm: AC-G

$Q := \{(X_i, X_j) \mid (X_i, X_j) \in A\}$
while $Q \neq \emptyset$ **do**
1. select and delete an arc (X_l, X_k) from Q
2. **if** $Dom(X_l) \neq^{MAX} Dom(X_l) \cap (Dom(X_k) \uplus \Gamma(X_k, X_l))$ **then**
   2.1. $Q := Q \cup \{(X_i, X_l) \mid (X_i, X_l) \in A,\ i \neq k\}$
   2.2. $Dom(X_l) := Dom(X_l) \cap (Dom(X_k) \uplus \Gamma(X_k, X_l))$
3. **if** $Dom(X_l) =^{MAX} \emptyset$ **then** $Q := \emptyset$; $Dom(X_l) := \emptyset$

end while

[Bettini et al. CP97 and AIJ02]. Key theorem: a solution exists iff there is one with all values lower than a network-dependent constant MAX.

The GSTP algorithm

Repeat

1. Conversion+PC
2. AC-G
3. RefineEdgesFromNodes()

Until no change is observed

Return Inconsistent **or** NewNetwork+solution

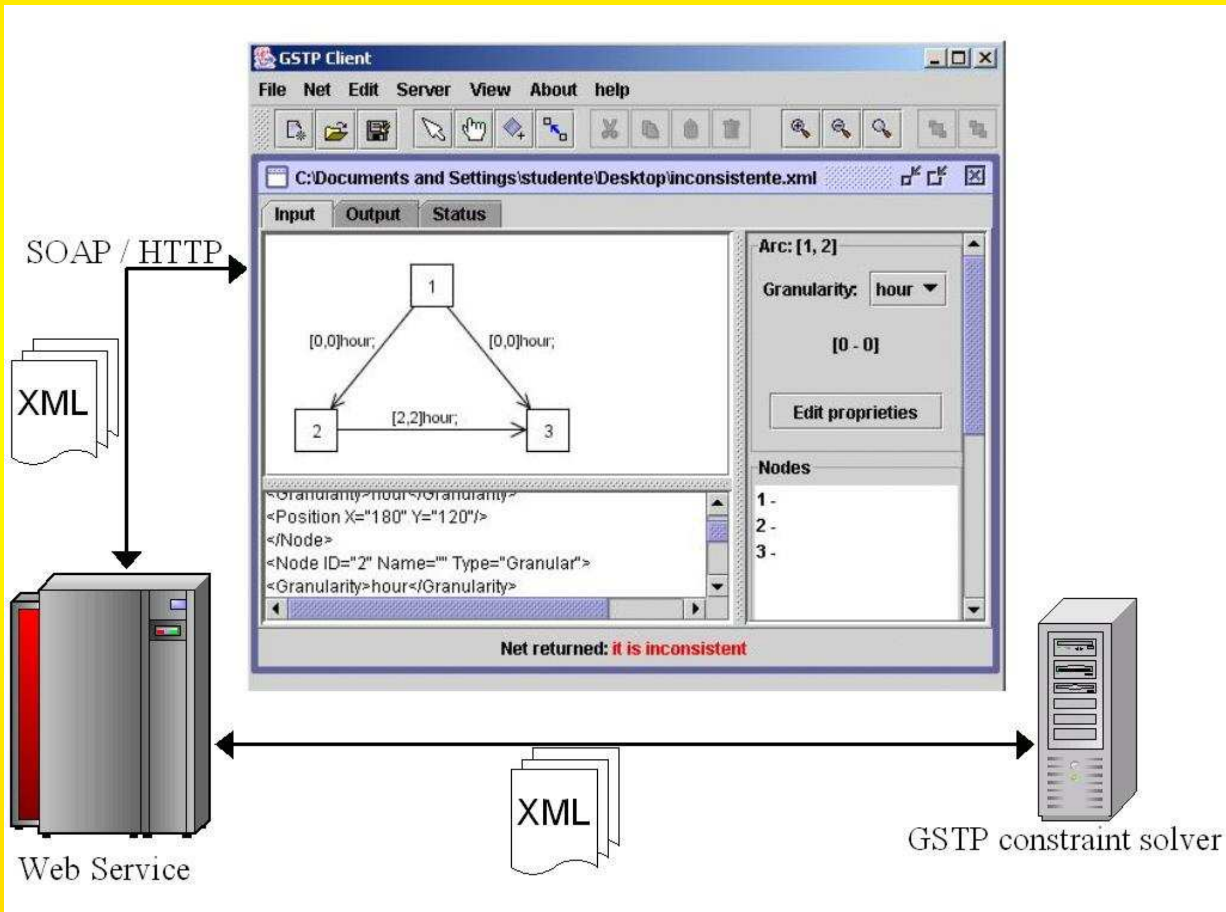
The GSTP web service

- Definition of an XML schema for constraint networks

```
...
<xsd:complexType name = "ArcType">
 <xsd:sequence>
  <xsd:element name="constraint" type="ConstrType" minOccurs="1"/>
 </xsd:sequence>
 <xsd:attribute name="sourceNodeID" use="required" type= .../>
 <xsd:attribute name="targetNodeID" use="required" type= .../>
</xsd:complexType>
...
```

- WSDL service description made available to external applications. (Services: consistency, refinement, solution, ...)

The GSTP web service architecture



References

- C.Bettini, S.Mascetti, V.Pupillo, GSTP: A Temporal Reasoning System Supporting Multi-Granularity Temporal Constraints, In Proc. IJCAI 2003 (Intelligent System Demonstration), Acapulco, Mexico.
- C. Bettini, X. Wang, S. Jajodia, Solving Multi-Granularity Constraint Networks, *Artificial Intelligence*, 140(1-2):107–152, 2002.
- C. Bettini, S. Jajodia, X. Wang, *Time Granularities in Databases, Temporal Reasoning, and Data Mining*. Springer, 2000.

Thank you for your attention

`http://webmind.dico.unimi.it/gstp`