

Calendars and Topologies as Types

—

A Programming Language Approach to Modelling Mobile Applications

Francois Bry, Bernhard Lorenz, Stephanie Spranger

Department of Informatics
Ludwig-Maximilians University of Munich
www.pms.ifi.lmu.de

16. Sept. 2005

1 Motivation

2 Temporal Domain: CaTTS

- Granularities
- Example: The Gregorian Calendar
- Applications

3 Spatial Domain (Future Work)

- Granularities
- Possible Applications

4 Questions / Discussion

Motivation

Developments:

- mobile applications / ubiquitous computing on the rise
- time and space essential for mobile applications

Challenges:

- time and space dependencies
- different time-zones / calendar systems
- requirements of spatial reasoning
- ...

Motivation

Developments:

- mobile applications / ubiquitous computing on the rise
- time and space essential for mobile applications

Challenges:

- time and space dependencies
- different time-zones / calendar systems
- requirements of spatial reasoning
- ...

Motivation

Developments:

- mobile applications / ubiquitous computing on the rise
- time and space essential for mobile applications

Challenges:

- time and space dependencies
- different time-zones / calendar systems
- requirements of spatial reasoning
- ...

Introduction to CaTTS

- *“calendars as types”*
 - declarative modeling
 - program annotation
 - increase of consistency
 - gain in efficiency
- *“theory reasoning”*
 - complements general-purpose “axiomatic reasoning”
 - efficient for specific domain
 - solves specific problems
 - exploits semantics of types

Introduction to CaTTS

- *“calendars as types”*
 - declarative modeling
 - program annotation
 - increase of consistency
 - gain in efficiency
- *“theory reasoning”*
 - complements general-purpose “axiomatic reasoning”
 - efficient for specific domain
 - solves specific problems
 - exploits semantics of types

CaTTS

CaTTS		
CaTTS-DL		CaTTS-CL
-FDL	-TDL	

- CaTTS: type language for calendars and reasoning on calendric data
- CaTTS-DL: calendars and representation formats
 - CaTTS-TDL: calendars and calendric types, e.g. a university calendar with types like teaching term
 - CaTTS-FDL: date formats, e.g. "5.10.2004"
- CaTTS-CL: calendric constraints, e.g. schedule some student's lectures and courses in a teaching term

CaTTS

CaTTS		
CaTTS-DL		CaTTS-CL
-FDL	-TDL	

- CaTTS: type language for calendars and reasoning on calendric data
- CaTTS-DL: calendars and representation formats
 - CaTTS-TDL: calendars and calendric types, e.g. a university calendar with types like teaching term
 - CaTTS-FDL: date formats, e.g. "5.10.2004"
- CaTTS-CL: calendric constraints, e.g. schedule some student's lectures and courses in a teaching term

CaTTS

CaTTS		
CaTTS-DL		CaTTS-CL
-FDL	-TDL	

- CaTTS: type language for calendars and reasoning on calendric data
- CaTTS-DL: calendars and representation formats
 - CaTTS-TDL: calendars and calendric types, e.g. a university calendar with types like teaching term
 - CaTTS-FDL: date formats, e.g. "5.10.2004"
- CaTTS-CL: calendric constraints, e.g. schedule some student's lectures and courses in a teaching term

Using Granularities in CaTTS

- interval-based “discretization” of continuous linear time line in terms of granularities
- e.g. “mapping” of days into continuous time line
 - day-granule “20.04.2005” has duration 1 (day)
 - interval of day-granules from “20.04.2005” to “21.04.2005” has duration 2 (day)
- granule referred to as event
- finite, convex, interval of granules referred to as task
- anchoring with “real time” (e.g. UTC seconds) possible

Using Granularities in CaTTS (cont'd)

- calendric type σ always definable from another type τ
 - aggregation, $\sigma \preceq \tau$ (e.g. *week* \preceq *day*)
 - inclusion, $\sigma \subseteq \tau$ (e.g. *working_day* \subseteq *day*)
- composition $\preceq \otimes \subseteq$ (e.g. *working_week* $\preceq \otimes \subseteq$ *day*): partial order over calendric types
- calendar: finite set of calendric types $\tau_0, \tau_1, \dots, \tau_n$ such that $\tau_i \preceq \otimes \subseteq \tau_0$ for all $i \in \{1, \dots, n\}$

Example: The Gregorian Calendar

```
calendar Gregorian:STANDARD =
cal
import LeapSeconds;
type second = reference;
type minute = aggregate
  alternate minute (i)
  | i == 1051200 (* 1.1.1972*) -> 70 second
  | hasLeapSec?(i)           -> 61 second
  | otherwise                 -> 60 second
end @ second(1);
type hour = aggregate 60 minute @ minute(1);
type day = aggregate 24 hour @ hour(1);
type week = aggregate 7 day @ day(-2);
type month = aggregate
  31 day named january,
  alternate month(i) (* 1972 (index 3) is a leap year *)
  | ((i div 12)-3) mod 4 == 0 &&
    (((i div 12)-3) mod 400 != 100 ||
     (((i div 12)-3) mod 400 == 0)) -> 29 day
  | otherwise                       -> 28 day
end named february,
31 day named march,
30 day named april,
31 day named may,
30 day named june,
31 day named july,
31 day named august,
30 day named september,
31 day named october,
30 day named november,
31 day named december @ day(1);
type year = aggregate 12 month @ month(1) ~@ 1970; (* index 1 refers to 1970*)
group day_of_week =
  with select day(i) where relative i in week == j
  type monday   where j == 5
  type tuesday  where j == 6
  type wednesday where j == 0
  type thursday where j == 1
  type friday   where j == 2
  type saturday where j == 3
  type sunday   where j == 4
end
end
```

Applications

- modeling: e.g. time tables, caledars, scheduling tasks
- query answering: e.g. querying multiple databases (flights, hotels, airport shuttle bus)
- appointment scheduling: e.g. schedule a meeting lasting 3 working days after April 22nd and before May
- ...

Spatial Domain (Future Work)

Broad applicability:

- 1-dimensional structures:
 - train lines, bus lines, ferries
 - intervals: stations, stops, docks
- 2-dimensional structures:
 - grids, coarse distributions
 - areas: parcels, iso-areas
- hierarchical structures:
 - government districts, jurisdiction
 - nodes: continent, country, state, district, city, quarter

Applications

- whenever resources are limited
- performance requirements are crucial
- high resolution spatial reasoning is not necessary

Thank you for your attention.

Further information:

- www.pms.ifi.lmu.de
- `{bry,lorenz,spranger}@pms.ifi.lmu.de`