A New Object-Oriented Model of the Gregorian Calendar

Hernán Wilkinson*
Mercap Development Manager
Tacuarí 202, 7mo Piso
C1071AAF, Buenos Aires, Argentina
54-11-4878-1118 (ext. 120)
h.wilkinson@mercapsoftware.com

Máximo Prieto**
Lifia – Facultad de Informática
Universidad Nacional de La Plata
cc11, 1900, La Plata, Argentina
+54 221 422-8252 (ext. 215)
maximo.prieto@lifia.info.unlp.edu.ar

Luciano Romeo
Mercap Software Architect
Tacuarí 202, 7mo Piso
C1071AAF, Buenos Aires, Argentina
54-11-4878-1118
l.romeo@mercapsoftware.com

* *Also Universidad de Buenos Aires, UBA, Argentina*
** *Also Universidad Nacional de la Patagonia Austral, Unidad Académica Caleta Olivia (UNPA-UACO)*

Problem Presentation

- Time Domain is pervasive
  - Cross Cutting
  - Related with Financial Domain
  - Related with almost every Domain…
- We concentrated our work on the Gregorian Calendar

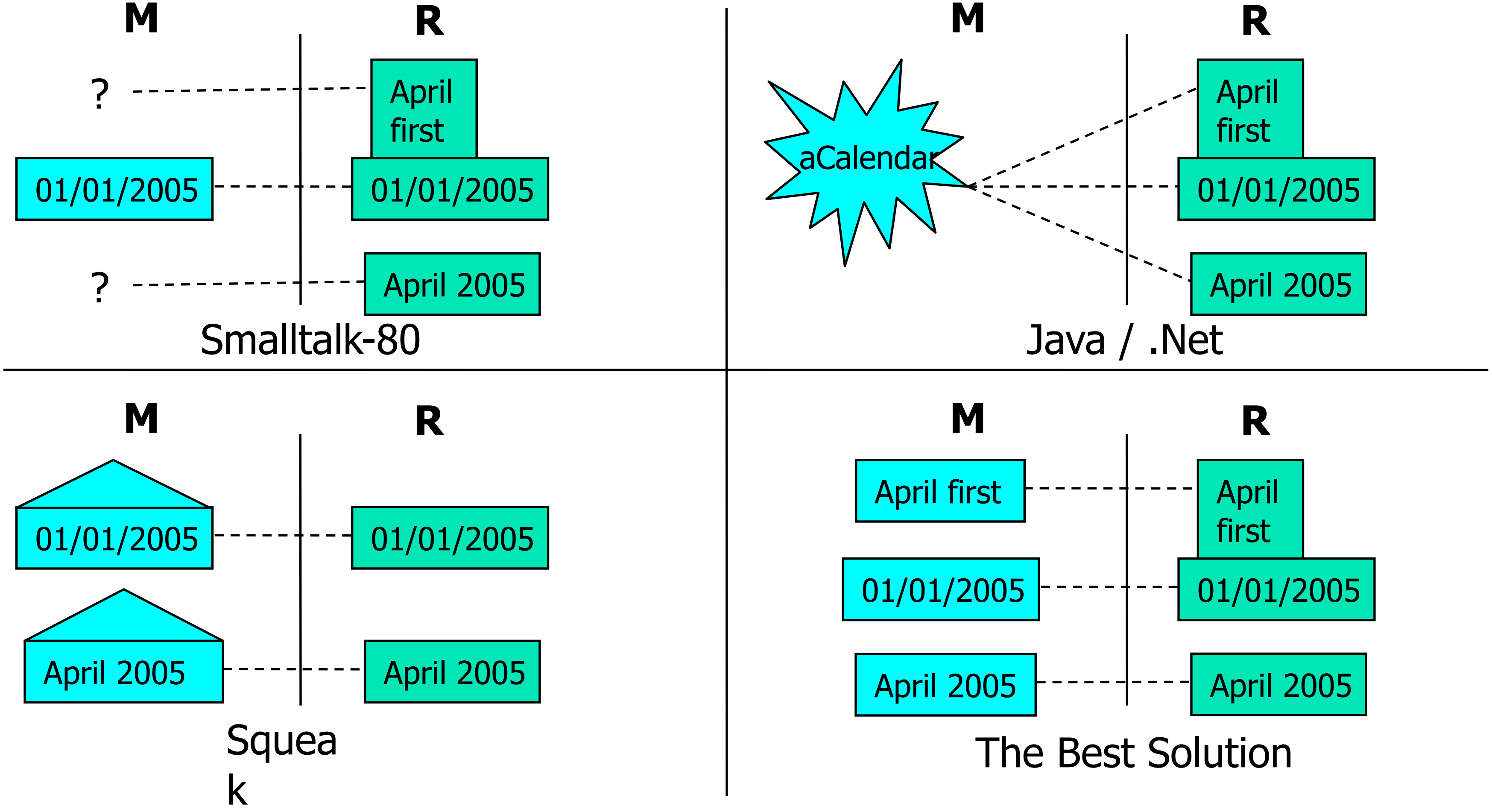
Gregorian Calendar

- Main Design Challenges
  - Irregularity
    - Months have 28,29, 30 or 31 days
    - Leap years
    - Associated with natural events (i.e. day/nigth, seasons, earth movement, etc.)
  - Comparing
    - January < July
    - January first < February twentyninth
    - 3 months < 1 year or 5 days < 1 week
  - Distance
    - (January distanceTo: July) = (January, 2005 distanceTo: July, 2005)
  - Time line filtering
    - workingDays includes: (January first, 2005)
    - workingDays
      daysBetween: (August fifteen, 2005) and: (August twenty, 2005)
    - 14 days from: (April first, 2005) counting: workingDays

Current Models' Limitations

- Lack of abstractions
  - Smalltalk-80: #Monday < #Friday (It returns false)
- Abstractions not matching reality
  - Squeak: Date dates (There are no days in a date…)
- Some models have one, or a few, general purpose abstractions
  - aCalendar.set (Calendar.MONTH, 1) (Does this mean January?)
  - Calendar.getInstance () (Is Calendar a singleton?)
- These problems show lack of understanding of the problem domain, they provide a poor domain language, therefore:
  - It is difficult to express common situations with them
  - They are difficult to learn
  - They offer different possible interpretations
- These problems imply:
  - Ad-hoc implementations
  - Code duplication

Matching Reality



M

R

?

April first

01/01/2005

01/01/2005

?

April 2005

Smalltalk-80

M

R

aCalendar

April first

01/01/2005

April 2005

Java / .Net

M

R

01/01/2005

01/01/2005

April 2005

April 2005

Squeak

M

R

April first

April first

01/01/2005

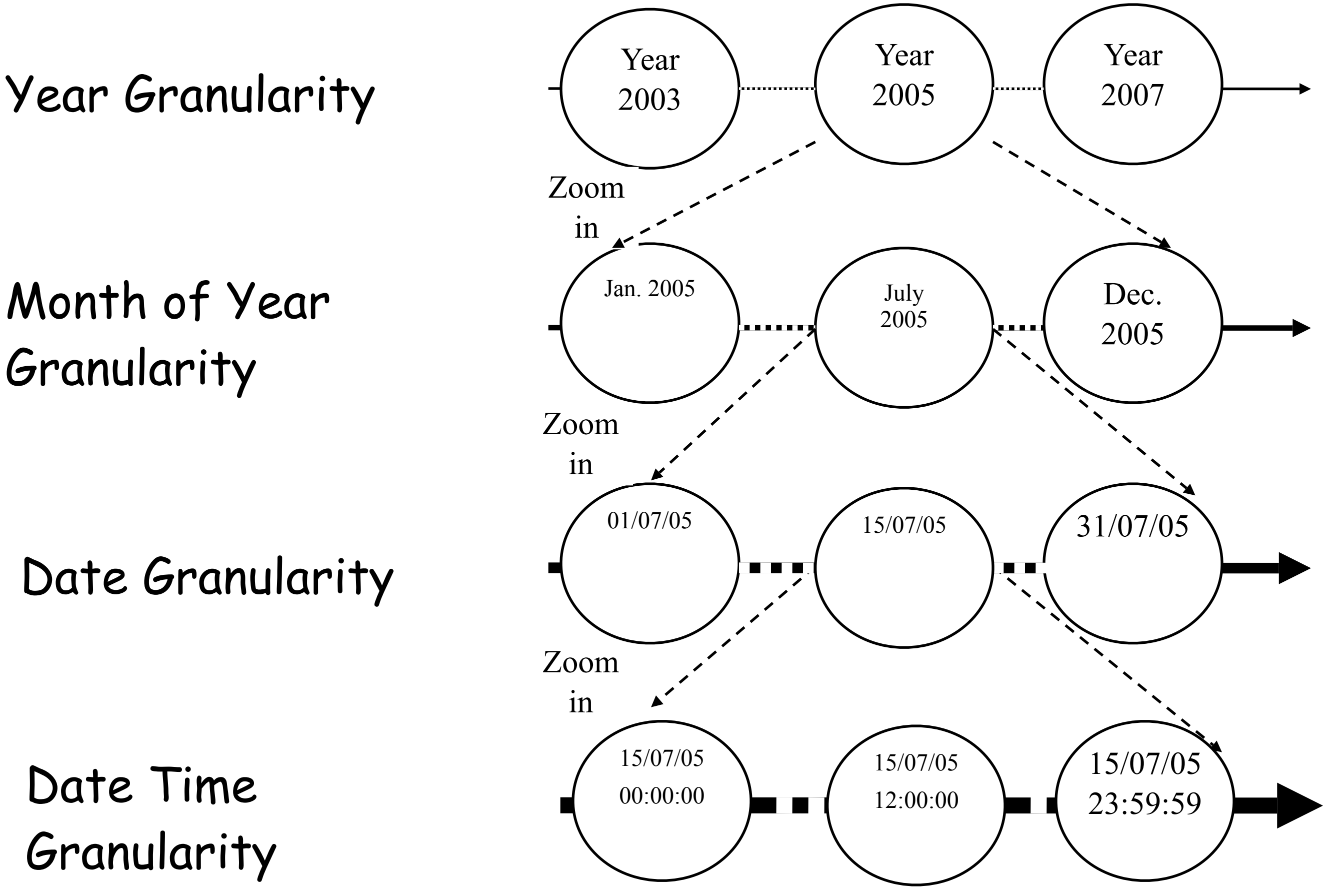
01/01/2005

April 2005

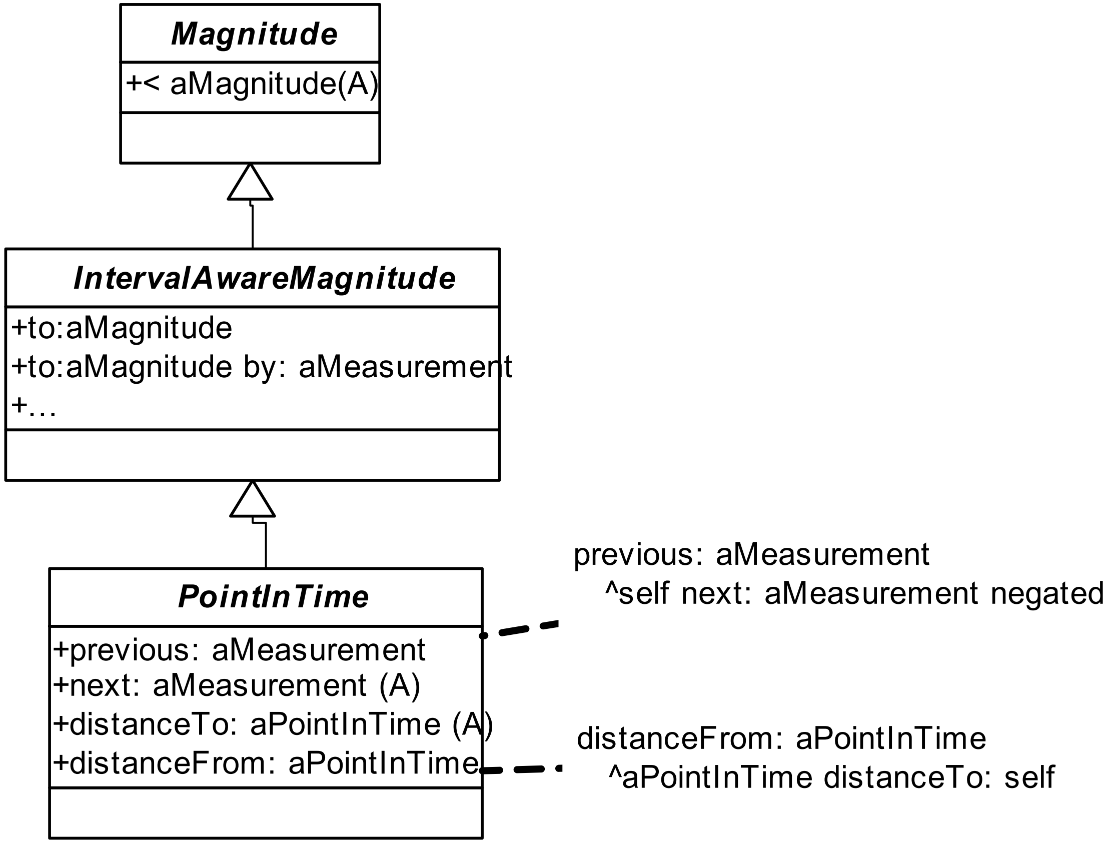
April 2005

The Best Solution

Our Metaphor



Main Abstraction



Magnitude

+< aMagnitude(A)

IntervalAwareMagnitude

+to:aMagnitude
+to:aMagnitude by: aMeasurement
+…

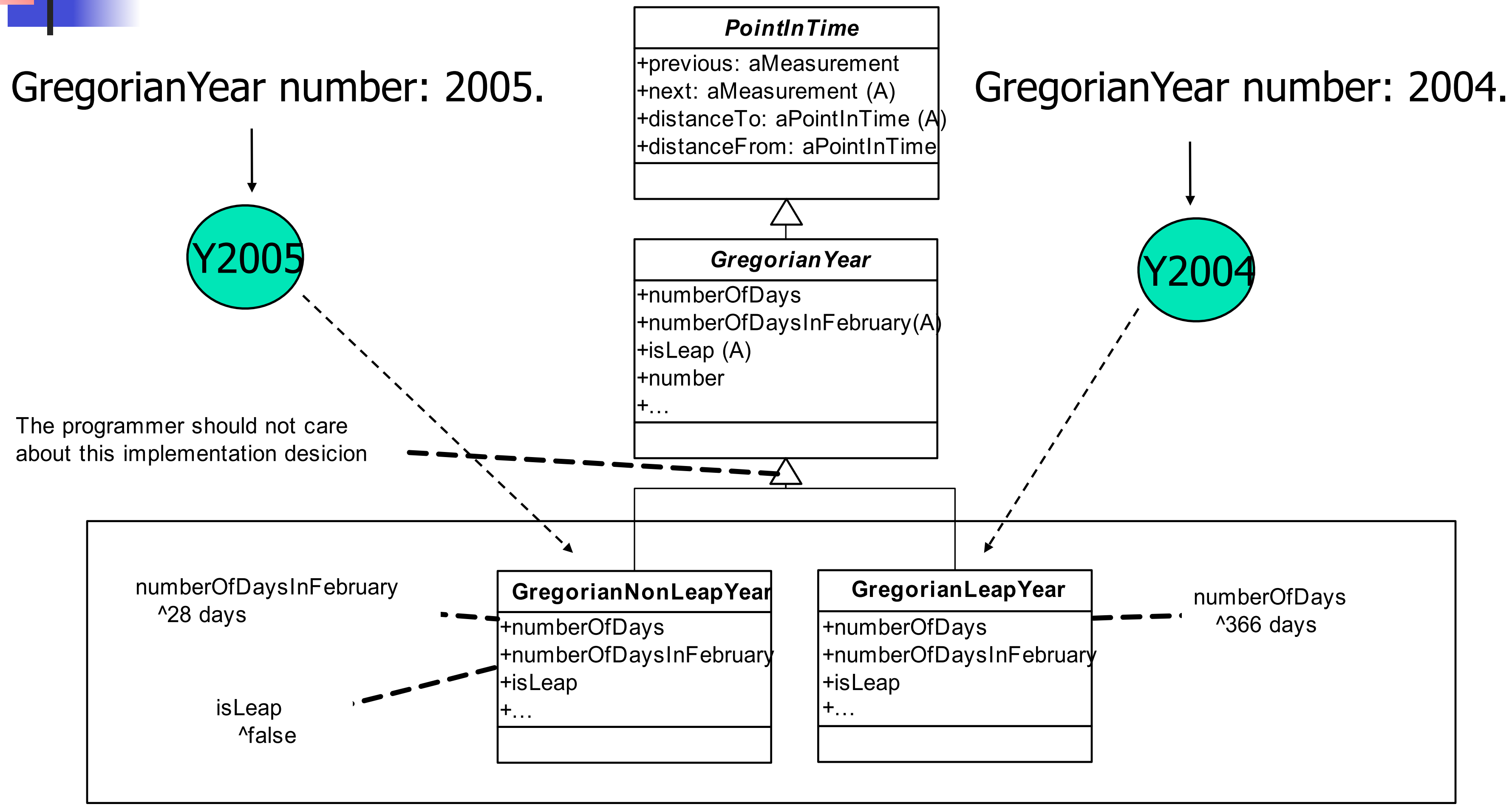
PointInTime

+previous: aMeasurement
+next: aMeasurement (A)
+distanceTo: aPointInTime (A)
+distanceFrom: aPointInTime

```
previous: aMeasurement
   ^self next: aMeasurement negated
```

```
distanceFrom: aPointInTime
   ^aPointInTime distanceTo: self
```

Year Granularity



GregorianYear number: 2005.

Y2005

PointInTime

+previous: aMeasurement
+next: aMeasurement (A)
+distanceTo: aPointInTime (A)
+distanceFrom: aPointInTime

GregorianYear

+numberOfDays
+numberOfDaysInFebruary(A)
+isLeap (A)
+number
+…

GregorianYear number: 2004.

Y2004

The programmer should not care
about this implementation desicion

numberOfDaysInFebruary
^28 days

isLeap
^false

GregorianNonLeapYear

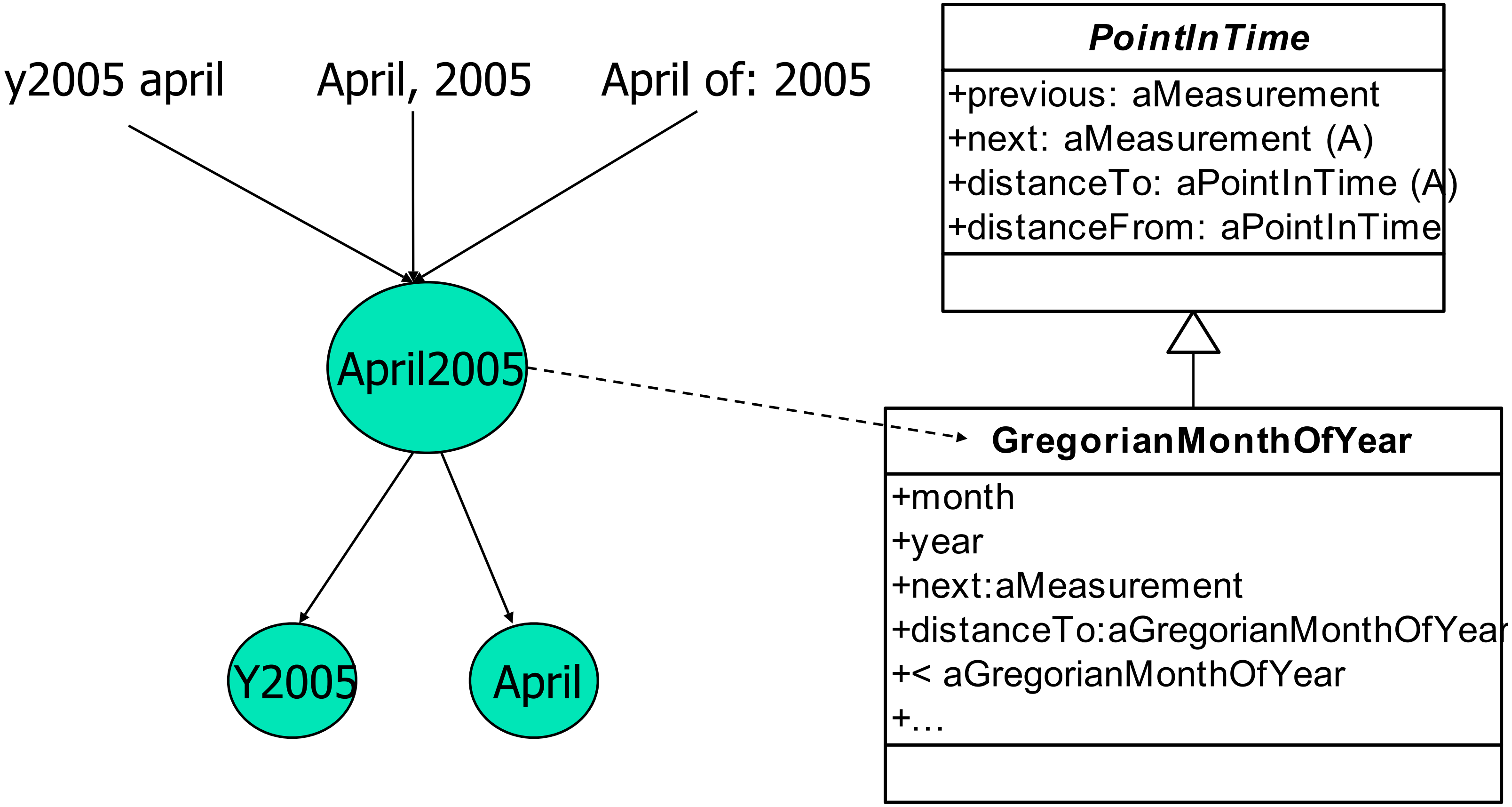
+numberOfDays
+numberOfDaysInFebruary
+isLeap
+…

GregorianLeapYear

+numberOfDays
+numberOfDaysInFebruary
+isLeap
+…

numberOfDays
^366 days

Month of Year Granularity



y2005 april

April, 2005

April of: 2005

April2005

Y2005

April

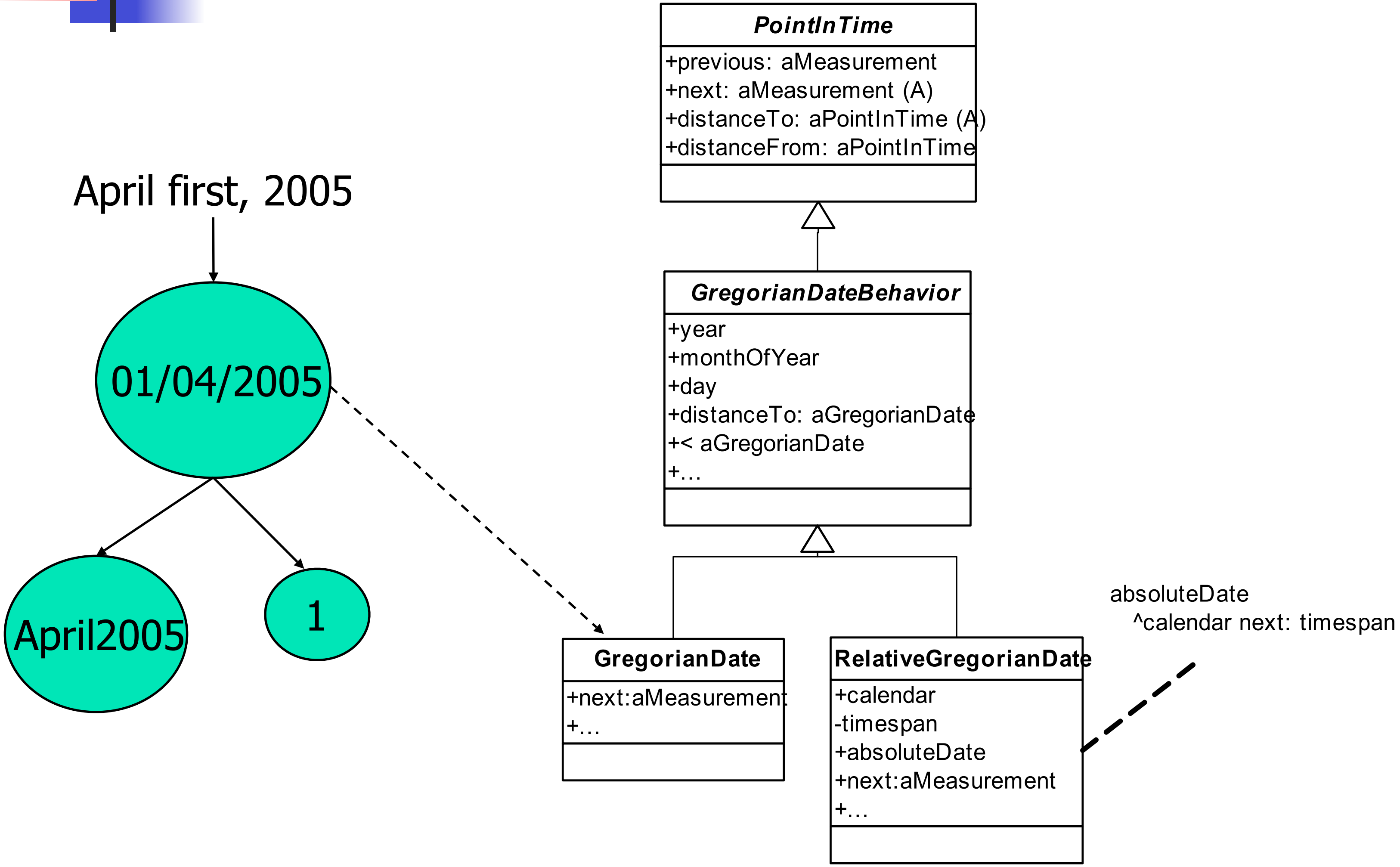
PointInTime

+previous: aMeasurement
+next: aMeasurement (A)
+distanceTo: aPointInTime (A)
+distanceFrom: aPointInTime

GregorianMonthOfYear

+month
+year
+next:aMeasurement
+distanceTo:aGregorianMonthOfYear
+< aGregorianMonthOfYear
+…

Date Granularity



April first, 2005

01/04/2005

April2005

1

PointInTime

+previous: aMeasurement
+next: aMeasurement (A)
+distanceTo: aPointInTime (A)
+distanceFrom: aPointInTime

GregorianDateBehavior

+year
+monthOfYear
+day
+distanceTo: aGregorianDate
+< aGregorianDate
+…

GregorianDate

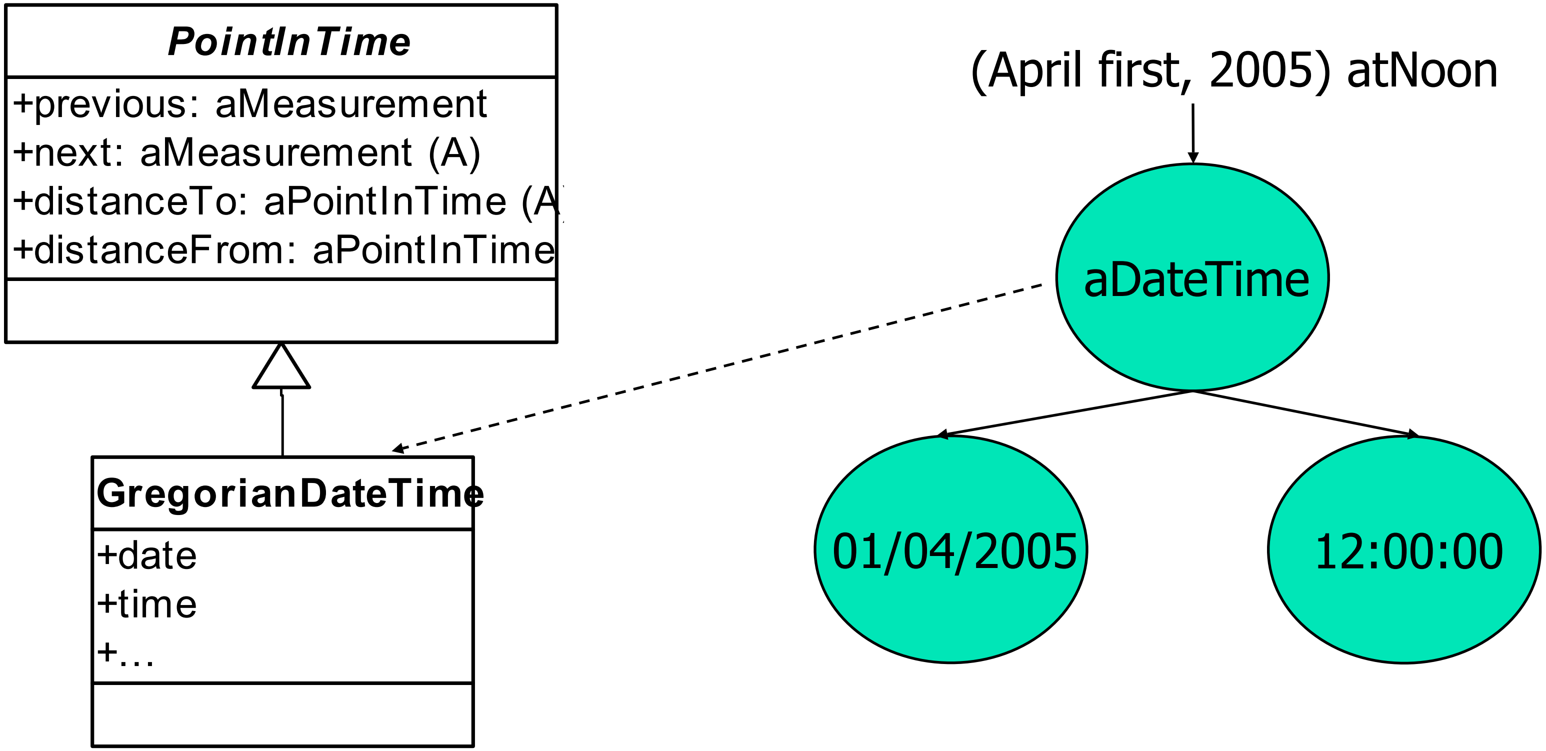
+next:aMeasurement
+…

RelativeGregorianDate

+calendar
-timespan
+absoluteDate
+next:aMeasurement
+…

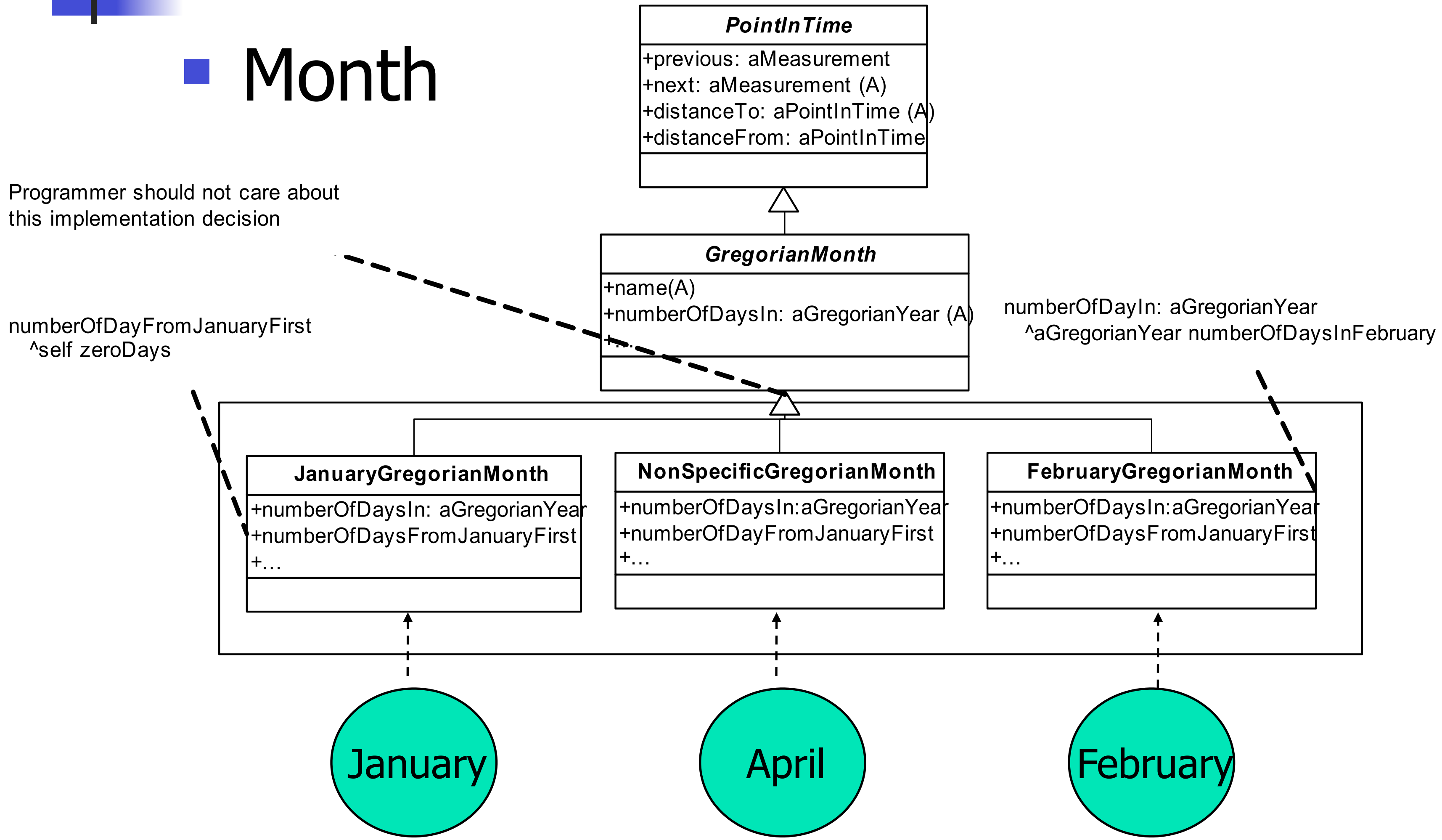
absoluteDate
^calendar next: timespan

Date Time Granularity



Recurrent Time Entities

- Month



Programmer should not care about this implementation decision

numberOfDayFromJanuaryFirst
^self zeroDays

PointInTime

+previous: aMeasurement
+next: aMeasurement (A)
+distanceTo: aPointInTime (A)
+distanceFrom: aPointInTime

GregorianMonth

+name(A)
+numberOfDaysIn: aGregorianYear (A)
+...

numberOfDayIn: aGregorianYear
^aGregorianYear numberOfDaysInFebruary

JanuaryGregorianMonth

+numberOfDaysIn: aGregorianYear
+numberOfDaysFromJanuaryFirst
+...

NonSpecificGregorianMonth

+numberOfDaysIn:aGregorianYear
+numberOfDayFromJanuaryFirst
+...

FebruaryGregorianMonth

+numberOfDaysIn:aGregorianYear
+numberOfDaysFromJanuaryFirst
+...

January

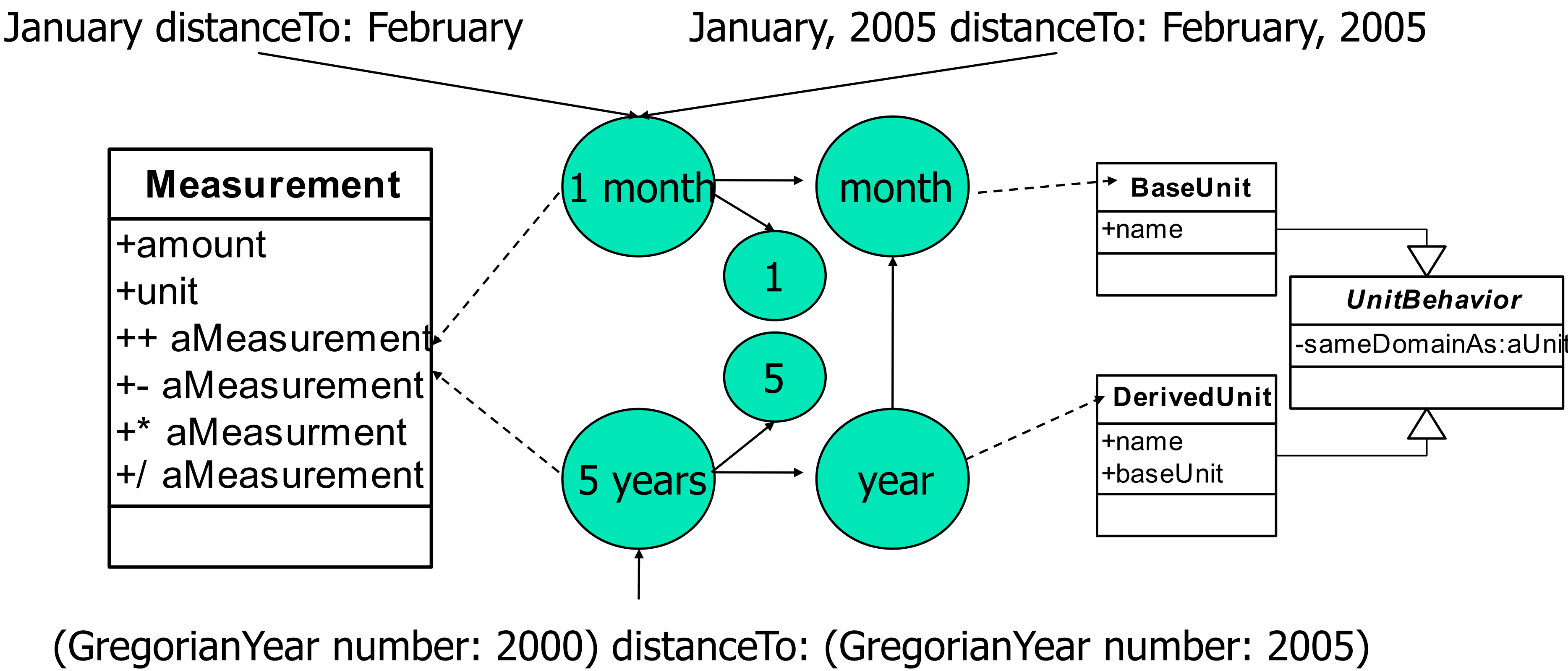
April

February

Recurrent Time Entities

- Day of Month
  - January first
  - December twentyFifth
- Day of Week
  - Monday
  - Tuesday
- Time of Day
  - TimeOfDay noon
  - TimeOfDay hours: 10 minutes: 11

Time measurements and Their Relevance



January distanceTo: February

January, 2005 distanceTo: February, 2005

Measurement

+amount
+unit
++ aMeasurement
+- aMeasurement
+* aMeasurment
+/ aMeasurement

1 month

month

1

5

5 years

year

BaseUnit

+name

UnitBehavior

-sameDomainAs:aUnit

DerivedUnit

+name
+baseUnit

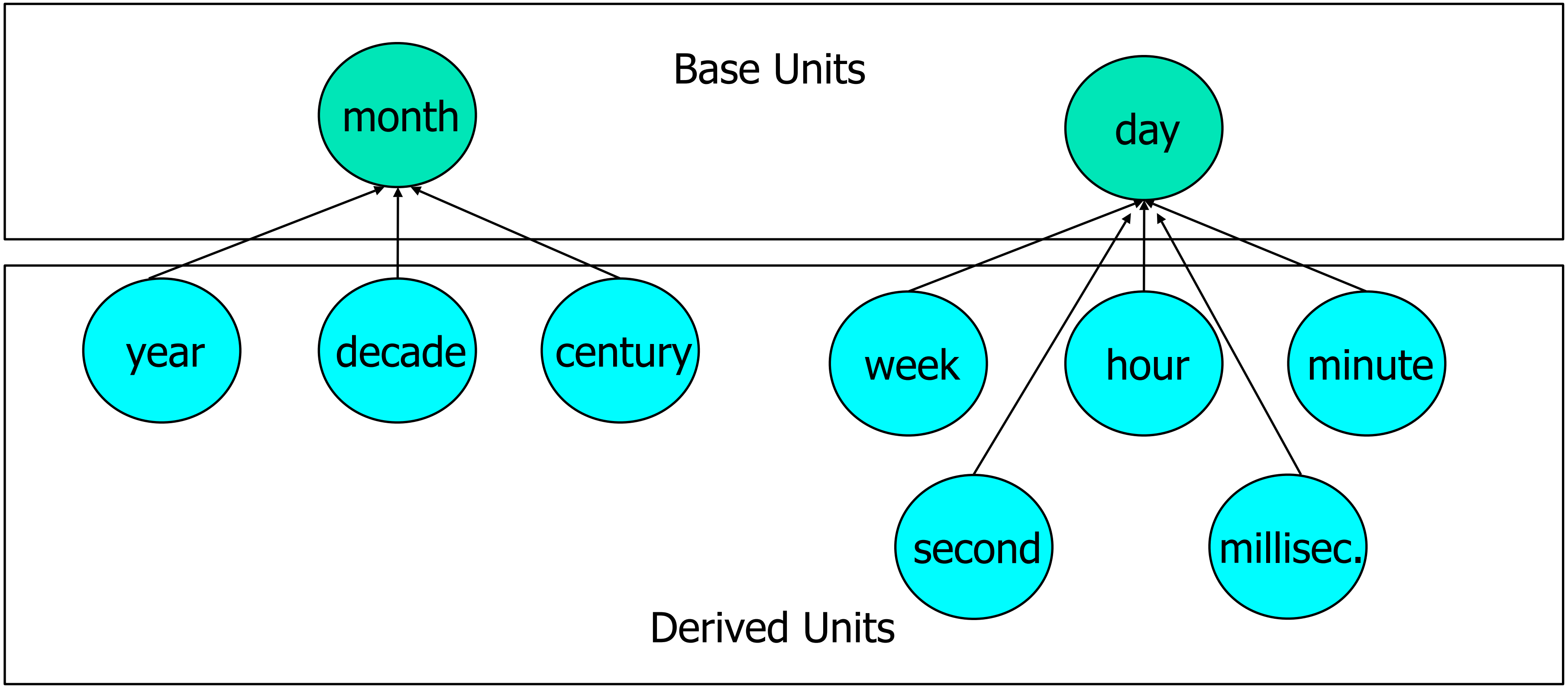
(GregorianYear number: 2000) distanceTo: (GregorianYear number: 2005)

Measurements

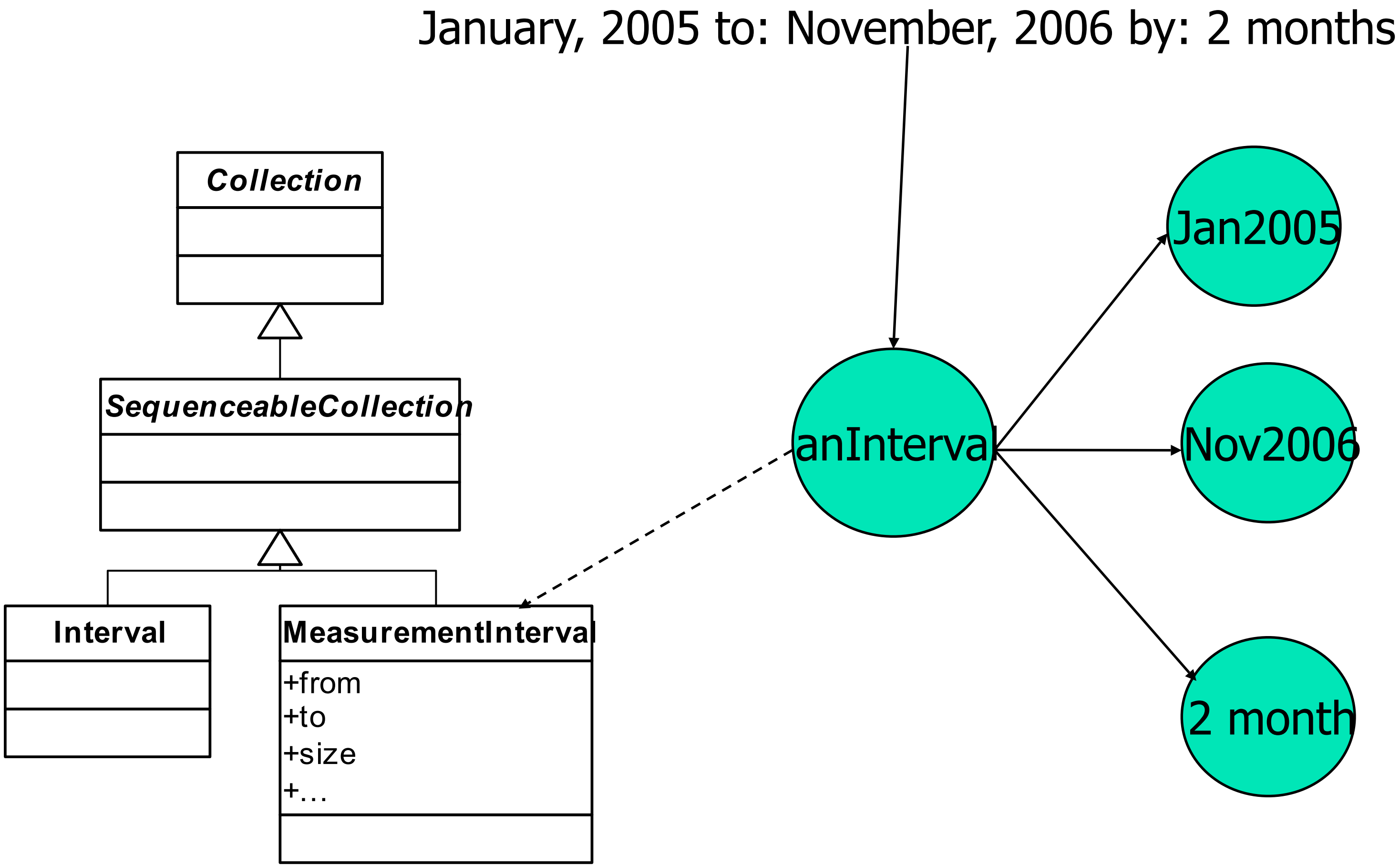
- Generic Model
- Will be presented at OOPSLA 2005
- Examples:
  - 1 year + 6 months → 18 months
  - 3 months + 5 days → 3 months + 5 days (A “Bag”)
  - 5 days + 3 weeks → 26 days
  - 1 day + 1 hour → 25 hours
  - 0.10 / 1 year → Yearly Interest Rate of 10 %
  - 40 km / 1 hour → Speed
  - 40 km / 1 hour * 2 hours → 80 km

Time Units

- Gregorian calendar irregularity

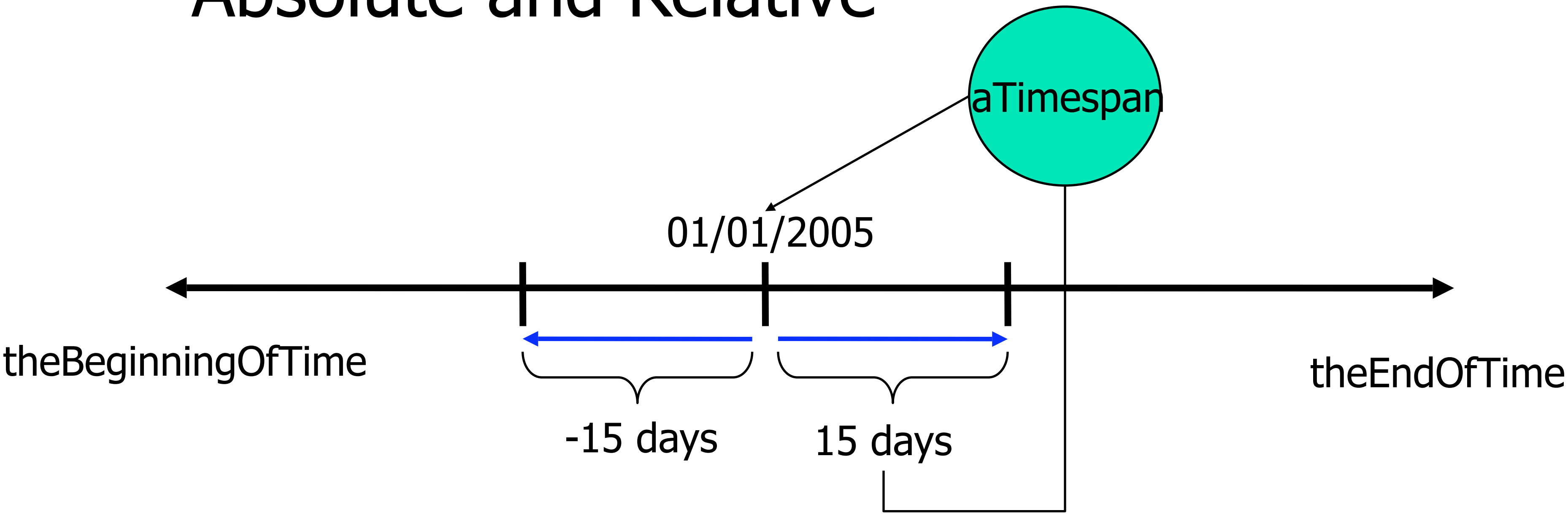


Intervals

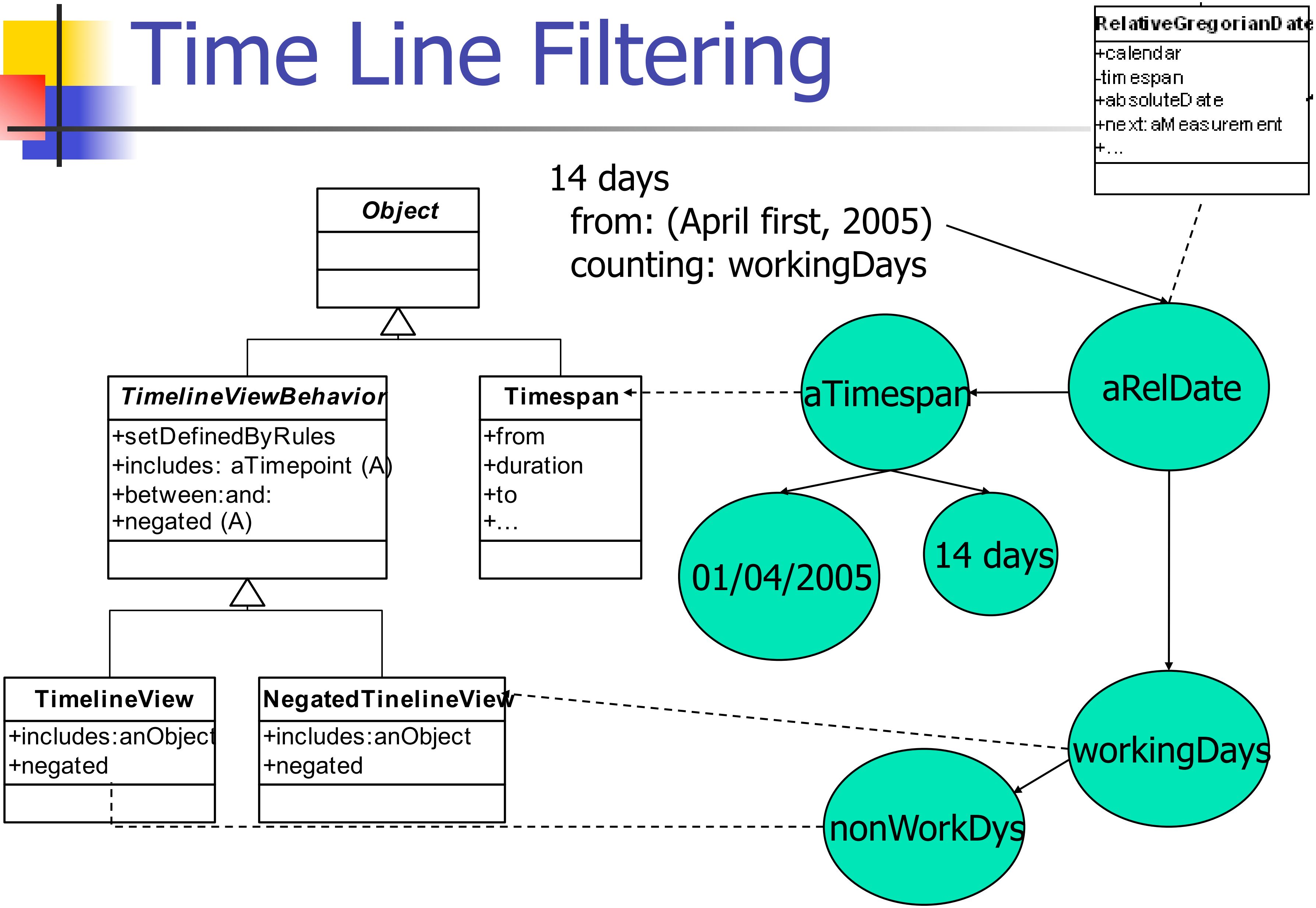

January, 2005 to: November, 2006 by: 2 months
Collection
SequenceableCollection
Interval
MeasurementInterval
+from
+to
+size
+…
anInterval
Jan2005
Nov2006
2 month

Segments

- Absolute and Relative



Time Line Filtering



14 days
from: (April first, 2005)
counting: workingDays

RelativeGregorianDate
+calendar
-timespan
+absoluteDate
+next: aMeasurement
+...

Object

TimelineViewBehavior
+setDefinedByRules
+includes: aTimepoint (A)
+between:and:
+negated (A)

Timespan
+from
+duration
+to
+...

TimelineView
+includes:anObject
+negated

NegatedTinelineView
+includes:anObject
+negated

aTimespan

aRelDate

01/04/2005

14 days

workingDays

nonWorkDys

Conclusions

- Object model of the Gregorian calendar
- Metaphor: Different resolution points of the time line
  - Total order between time points
  - Distance
  - Move from one point to another
  - Move between points of different resolution
- Representation of time line segments and intervals
- Generic Measurement Model to reify Time Measurements
- Time line filtering allows Relative points in time
- Abstractions for time entities such as a day, a day of a month and months.

Future Work

- Time zone support
- Expand **Timespan** protocol
- New abstractions like **Hour**, **Minute**, etc. (not units)
- Reify time lines
- Allow relative points of any granularity

Questions