Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre - RS - BRAZIL

Adding Time to an Object-Oriented Versions Model

Mirella Moura Moro
Nina Edelweiss
Silvia Maria Saggiorato
Clesio Saraiva dos Santos

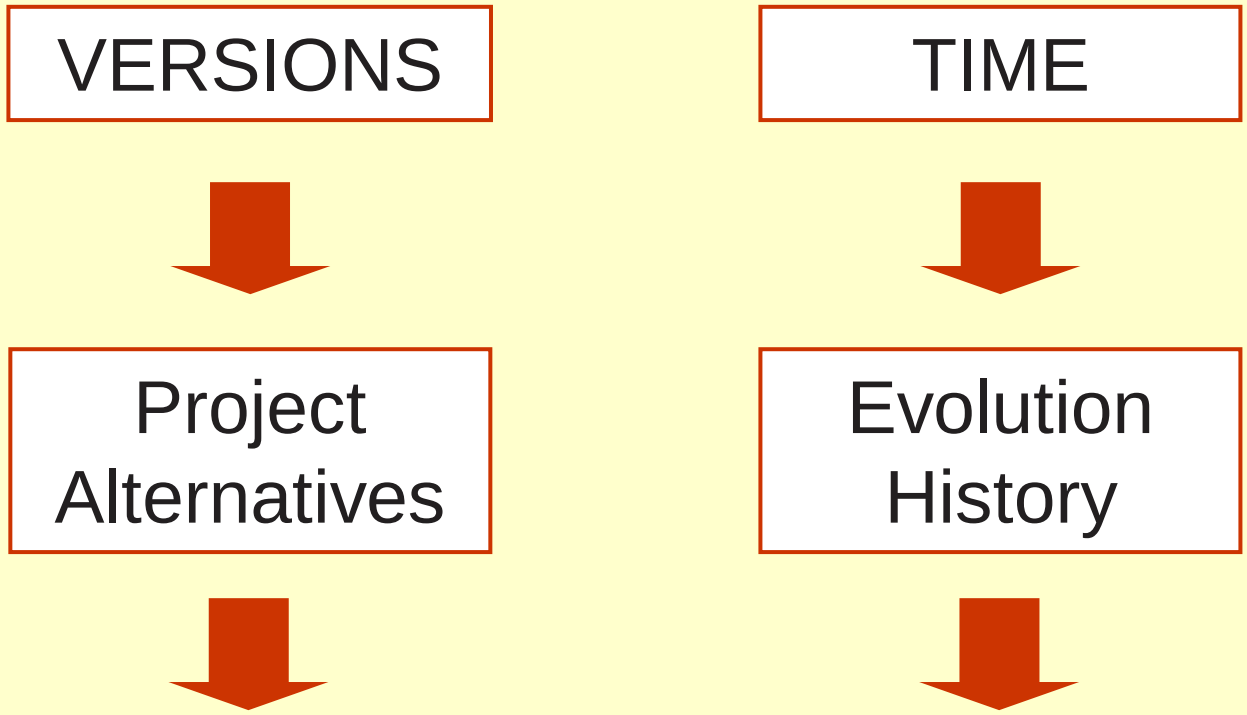
Outline

- Introduction
- Objective
- Basic Concepts
- Temporal Versions Model
- Example
- Implementing TVM
- Concluding Remarks

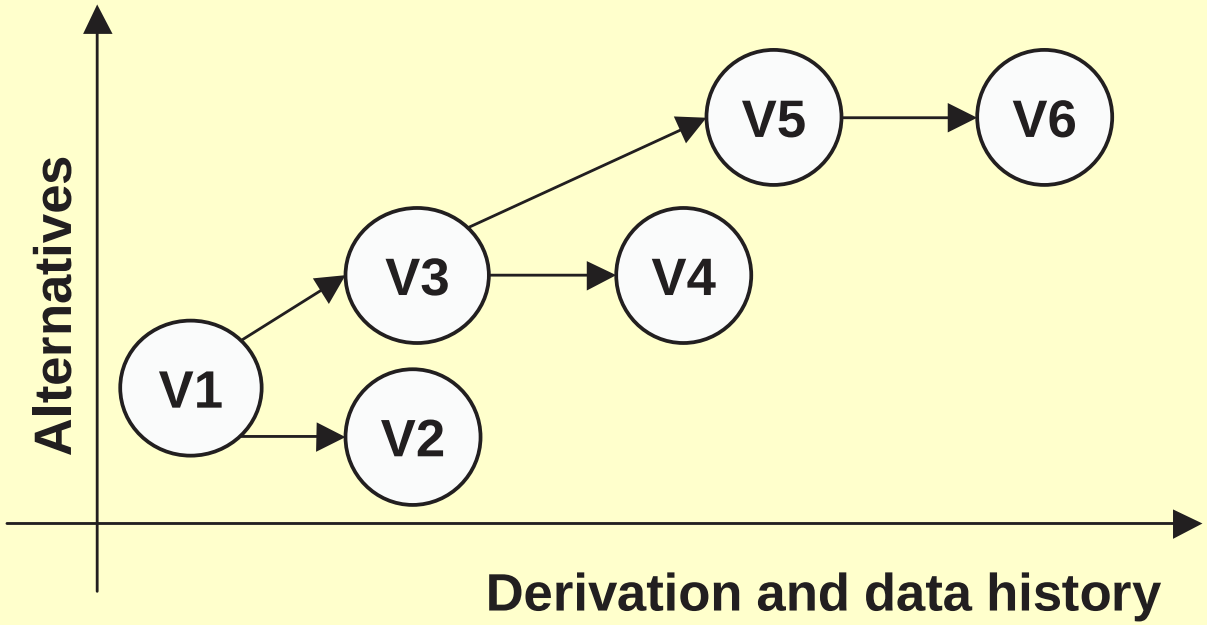
Introduction

- Version: describes an object in a period of time or from a certain point of view
- Temporal Model: specifies both static and dynamic aspects of the application by associating temporal labels

Objective


VERSIONS
TIME
Project Alternatives
Evolution History
Temporal Versions Model

Objective



Basic Concepts

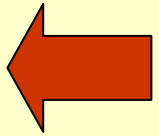
- Versions Model
- Temporal information
  - order: linear, branched, circular
  - timestamp: instant, interval, element
  - time dimension: transaction time, valid time, bitemporal

Temporal Versions Model

TEMPORAL FEATURES

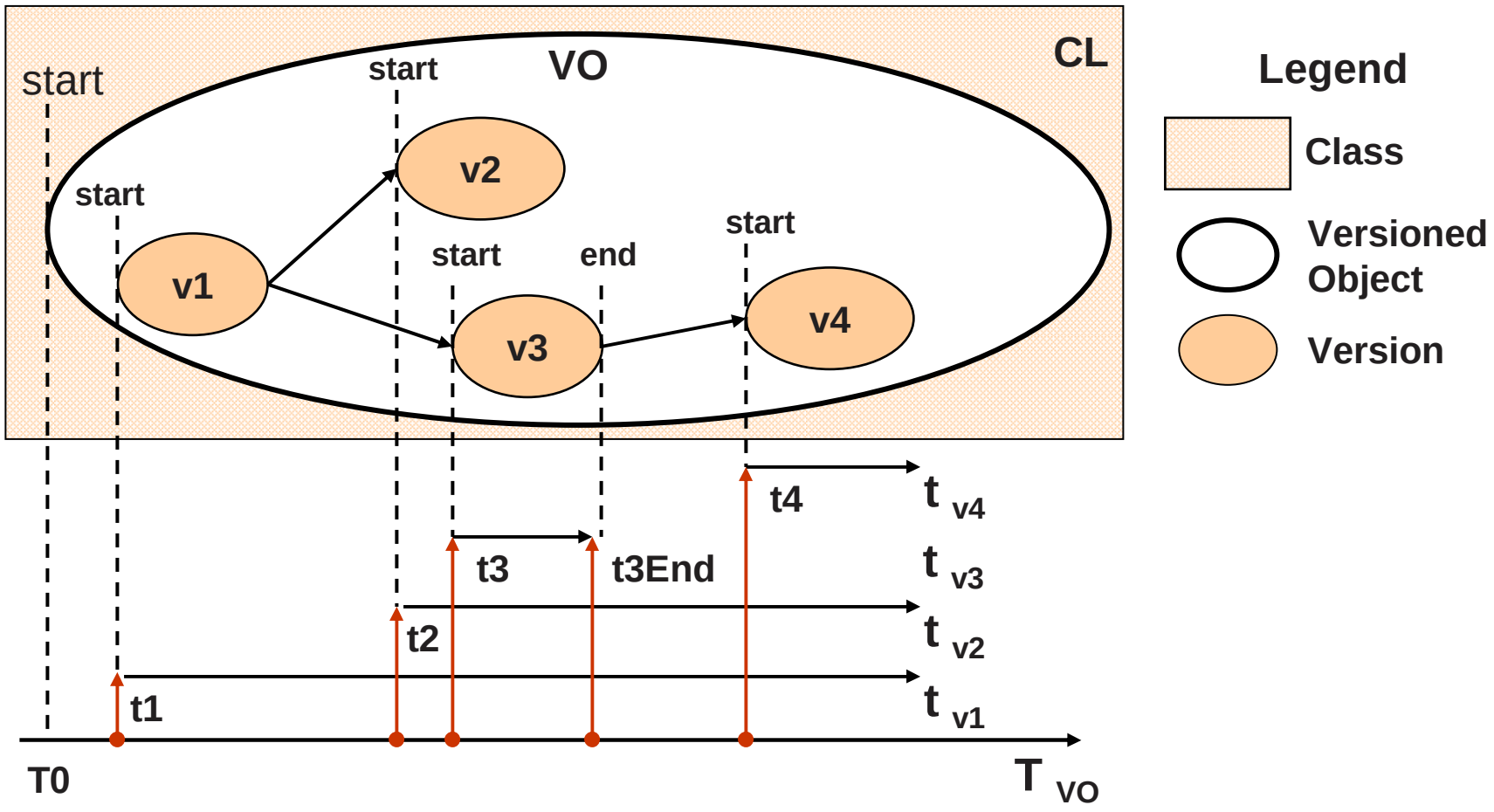
- object, version, attribute, relationship
- temporal label: interval
- bitemporal (valid and transaction times)
- object and versions: *start & end*
- temporal label:

  vTimei, *vTimef*, *tTimei*, *tTimef*
- version: a lifetime line
- object: branched time (each version)



Temporal Versions Model

TEMPORAL FEATURES - ATTRIBUTES *start*



Temporal Versions Model
LOGICAL AND PHYSICAL REMOVAL

- LOGIC
  - version ⇒ status *deactivated*
  - lifetime finished
    - *end* = exclusion instant
    - *vTimef*, *tTimef* = *end*
- PHYSICAL
  - reduce the database size
  - lost information = lost historic
- all exclusions are logical

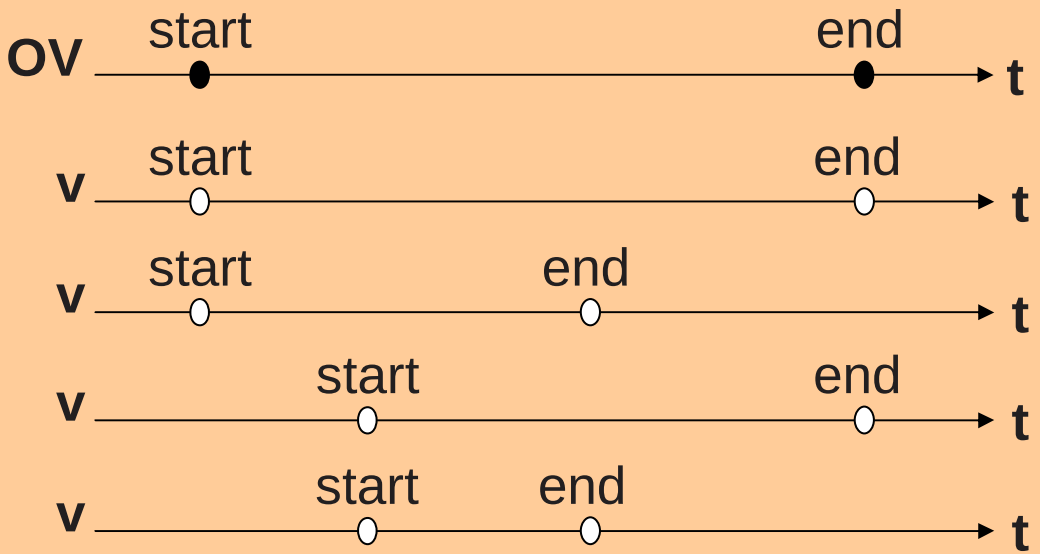
Temporal Versions Model

TEMPORAL INTEGRITY RULES

Valid Time

Transaction Time

Lifetime



Temporal Versions Model

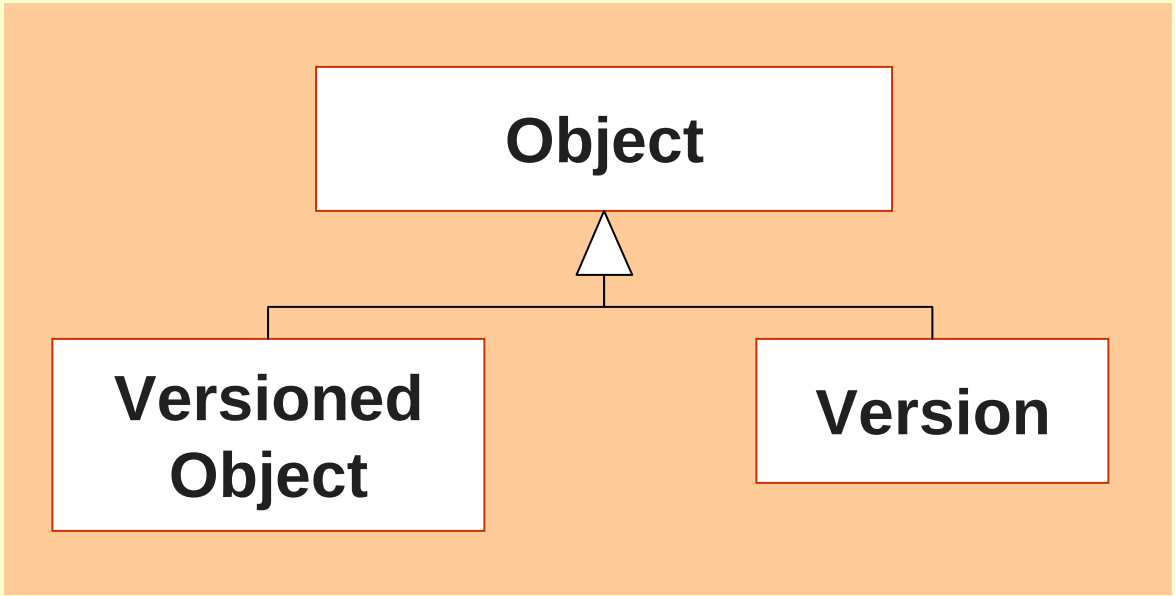
VERSION FEATURES

- versioned objects and versions
- derivation among versions: directed acyclic graph
- tvOID structure

    <entity id, class id, version number>

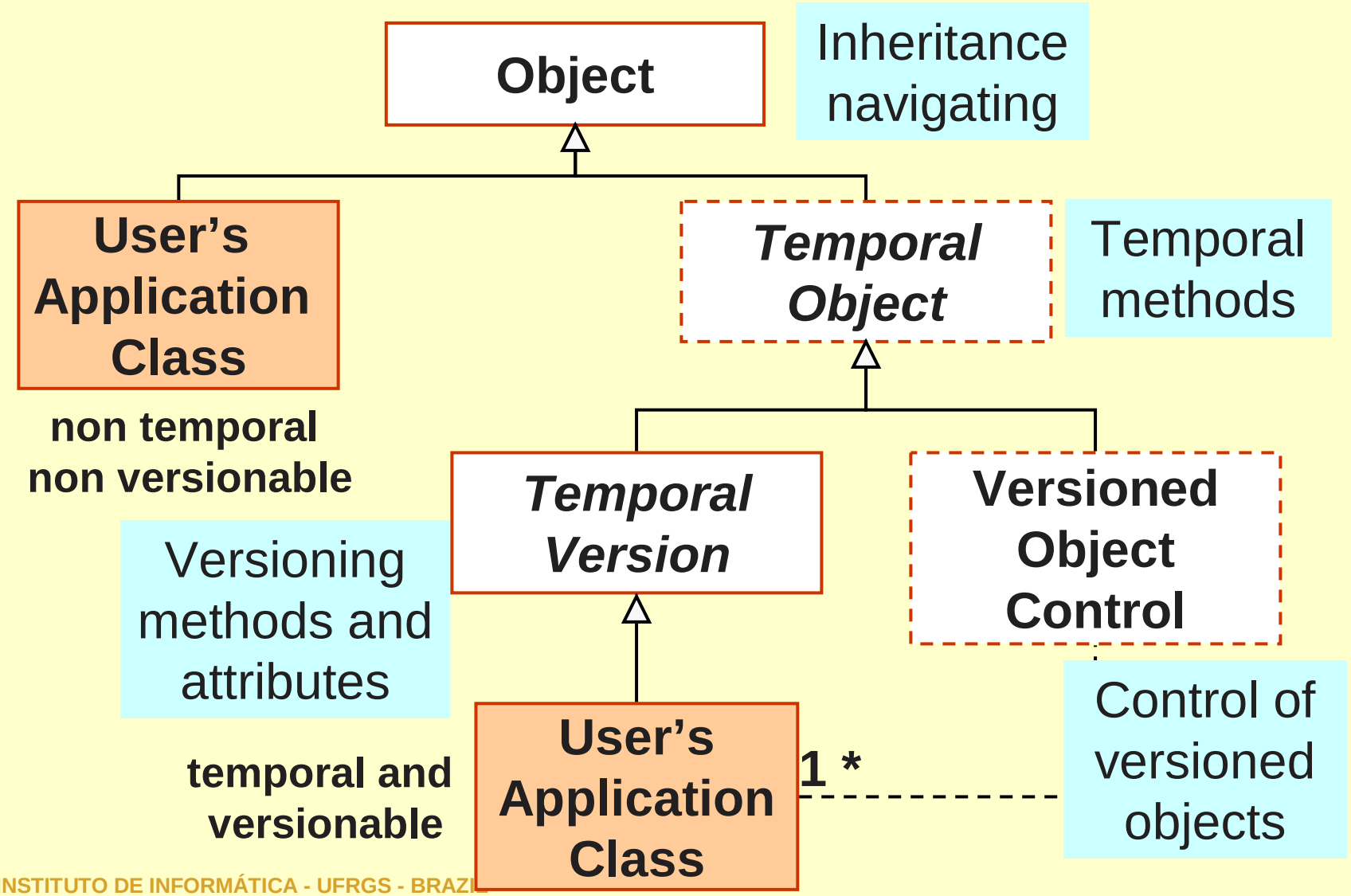
Temporal Versions Model

CLASS DIAGRAM (Golendziner)



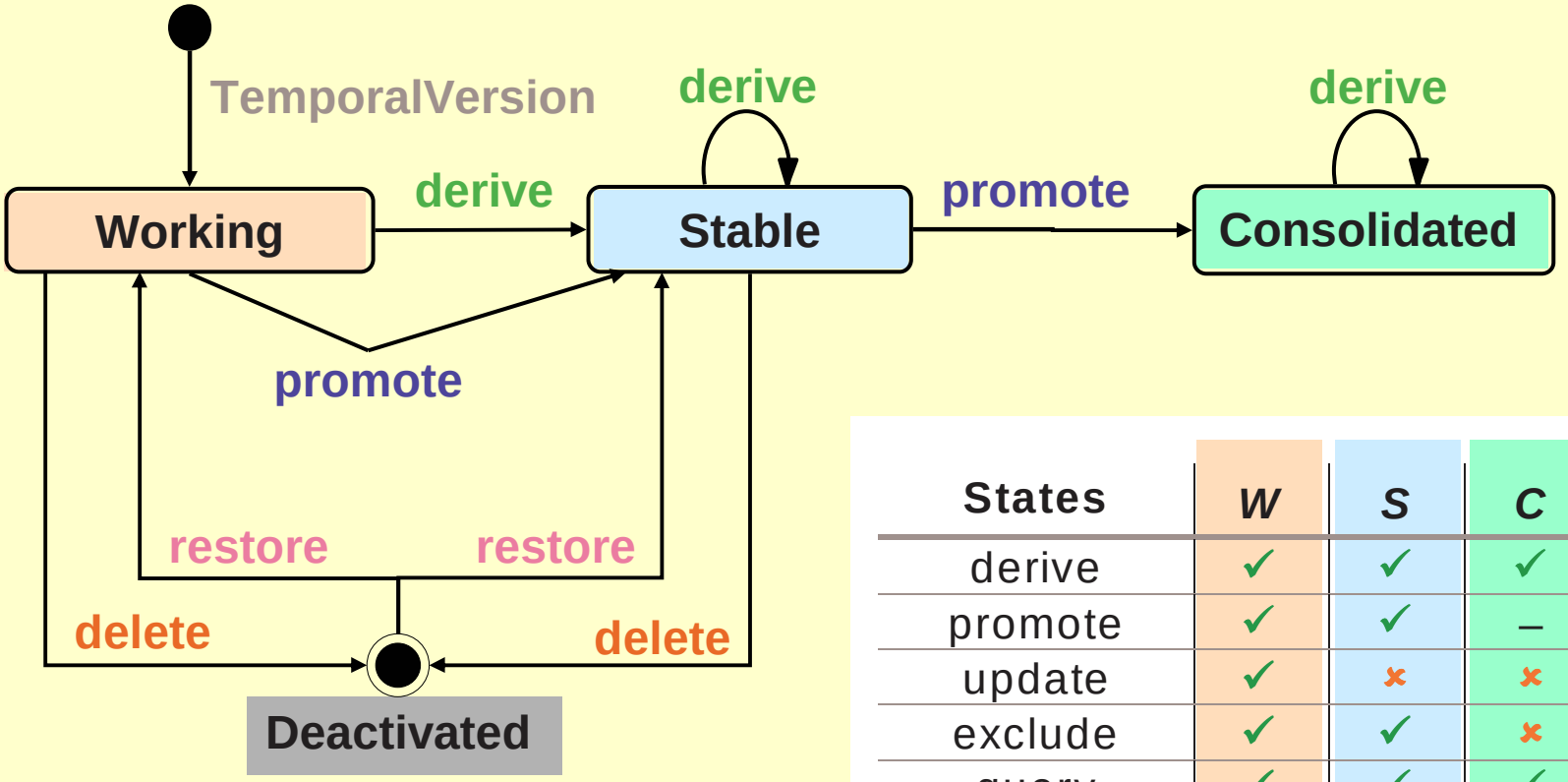
Temporal Versions Model

CLASS DIAGRAM



Modelo Temporal de Versões

STATE DIAGRAM



States	*W*	*S*	*C*	*D*
derive	✓	✓	✓	–
promote	✓	✓	–	–
update	✓	✗	✗	✗
exclude	✓	✓	✗	✗
query	✓	✓	✓	✓
share	✗	✓	✓	✗
restore	–	–	–	✓

Temporal Versions Model
CLASS DEFINITION LANGUAGE

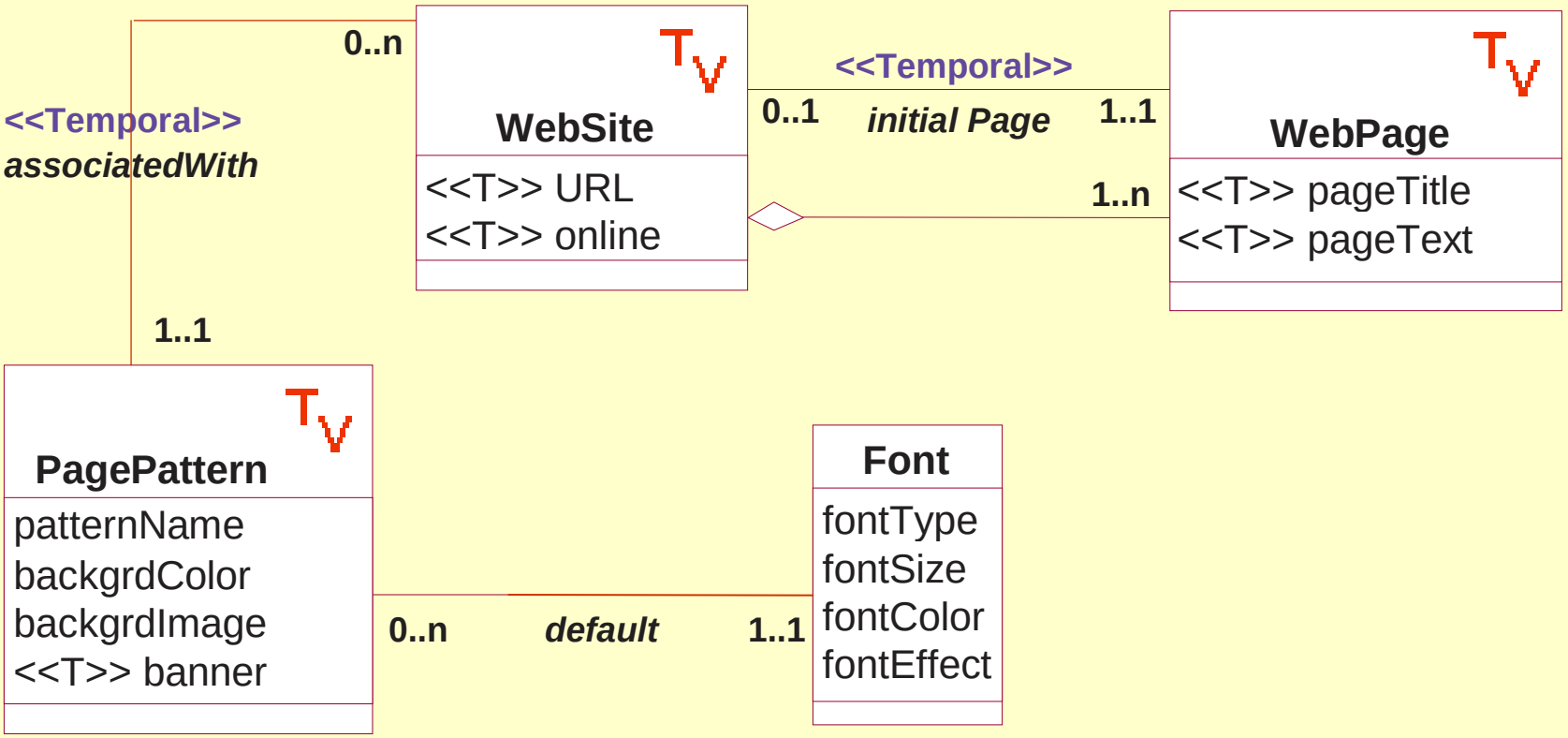
classDef ::= [**public**][**abstract** | **final**]
class className [**hasVersions**] [**inherit**
[**byExtension**] className [**correspondence** (**1:1** | **1:n** | **n:1** | **n:m**)]]
[[**temporal**] **aggregateOf** [**n**] className (**byValue** | **byReference**)
{, [**temporal**] **aggregateOf** [**n**] className (**byValue** | **byReference**)}]
([**Properties**:
{ [**public** | **private** | **protected**] [**static**]
[**temporal**] attributeName : attributeDomain [**default** value] **;** }+]
[**Relationships**:
{ [**temporal**] relationshipName (**0:1** | **0:n** | **1:1** | **1:n** | **n:m**)
[**inverse** inverseRelationshipName] relatedClassName **;** }+]
[**Operations**:
{ [**public** | **private** | **protected**] [**static**]
[**abstract** | **final**] operationDefinitions }+] **)** ;

Example

- Website design company
- Pages with pattern associated to
- Pattern varies according to the seasons of the year and commemorative dates

Example

CLASS DIAGRAM

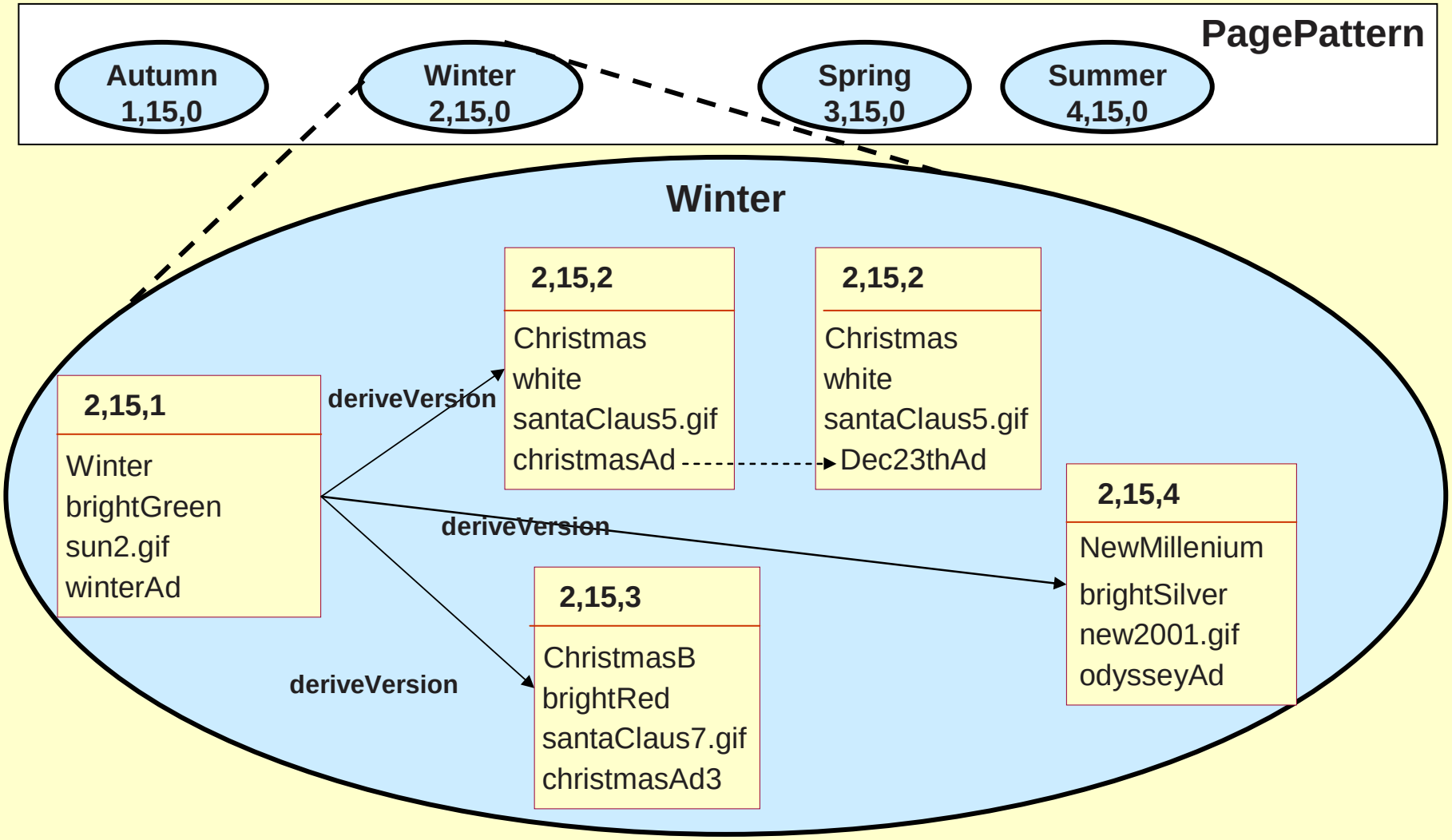


Example
DEFINITION LANGUAGE

```
Class WebSite HasVersions
      aggregate_of n WebPage byReference
   ( Properties:
            temporal URL : string ;
            temporal online : boolean;
        Relationship:
            temporal initialPage 1:1 WebPage;
            temporal associatedWith 1:1 PagePattern;
      ) ;
Class WebPage HasVersions
      ( Properties:
            temporal pageTitle : string;
            temporal pageText : text;
      ) ;
...
```

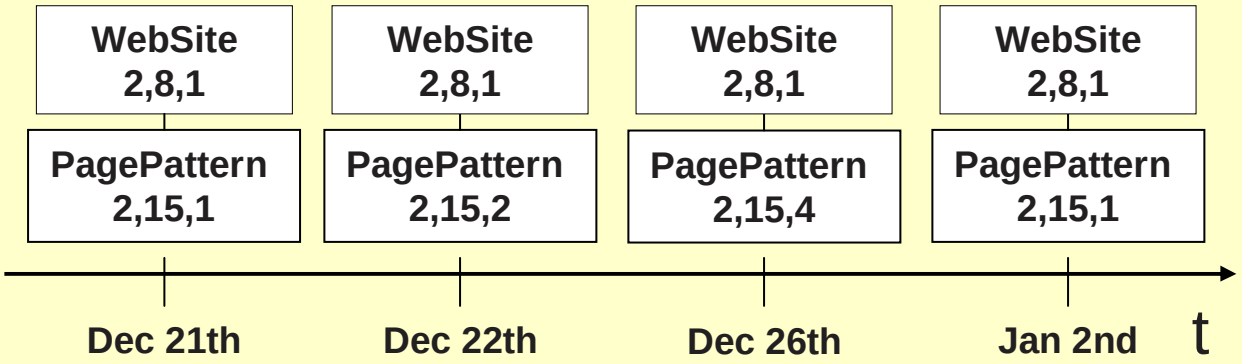
Example

GRAPHIC REPRESENTATION



Exemplo

RELATIONSHIP *AssociatedWith*



PagePattern associatedWith	Initial Transaction Time	Final Transaction Time	Initial Valid Time	Final Valid Time
2,15,1	12/01/2000	>>	12/21/2000	12/21/2000
2,15,2	12/01/2000	>>	12/22/2000	12/25/2000
2,15,4	12/01/2000	>>	12/26/2000	01/01/2001
2,15,1	12/01/2000	>>	01/02/2001	>>

Example
ADVANTAGES

- Evolution of the clients' website is stored
- Company has the employee historical information
- Discover patterns and clients' profiles by using data mining techniques

Implementing TVM

MODEL HIERARCHY MAPPING

- Existing commercial DBMS
- Mapping from TVM to DB2
- Class hierarchy > type hierarchy

  methods: SQL commands + queries
- *Object*, *TemporalObject*, *TemporalVersion*: NOT intantiable structured type
- *VersionedObject* (control): instantiable by the system
  - table VersionedObjects: store all instances

Implementing TVM
COMPLEMENTARY MAPPING

- OID: elements > varchar attribute separated by commas (‘101,20,5’)
- METADATA
  - CONTROL_CLASS:
    - ID_CLASS
    - CLASS
    - VERSIONED
    - IS_ROOT
    - ASCENDANT
- Temporal labes: TEMPORALSTAMP

Implementing TVM
APPLICATION CLASSES MAPPING

- NORMAL CLASS
  1. a structured type: class structure
  2. a table: stores class instances
- TV CLASS
  1. a structured type
  2. a table
  3. a table for each temporal property
  4. triggers: temporal and version updates

Implementing TVM

APPLICATION CLASSES MAPPING

PagePattern
tvOID
start
end
ascendant
configuration
descendant
predecessor
status
successor
refVersObjects
patternName
backgrdColor
backgrdImage
banner
refPagePatternBanner

PagePattern_Banner
tvOID
vTimei
vTimef
tTimei
tTimef
refPagePattern
banner

Concluding remarks

- Temporal Versions Model as an application specification technique for dynamic systems (time-evolving systems)
- Extension of the Versions Model by adding time features
- Design alternatives and data evolution

Concluding remarks

- For each version, every update in attributes and relationships values defined as temporal are stored
- Different versions coexist: branched time order
- Integration with existing specifications: the Model does not require all classes to be temporal and versionable

Concluding remarks

- On-going work:
  - Environment for class specification, object versioning, versions management, query, and visualization
  - Schema evolution model
  - Extender for DB2

Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre - RS - BRAZIL

Adding Time to an Object-Oriented Versions Model

nina@inf.ufrgs.br