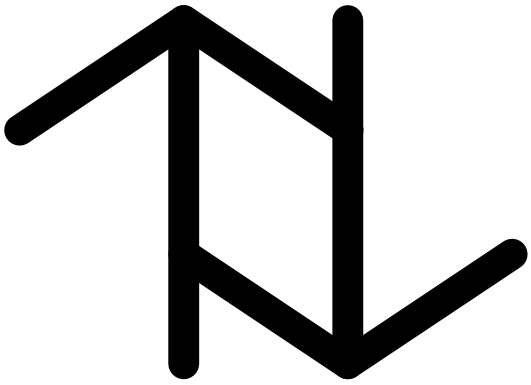

τ*DOM*: A Time-Aware API for Managing Temporal XML Documents

Haitao Liu

October 14, 2003

TR-74

A TIMECENTER Technical Report

Title	*τDOM*: A Time-Aware API for Managing Temporal XML Documents
Author(s)	Haitao Liu
Publication History	October 2003, A TIMECENTER Technical Report

TIMECENTER Participants

Aalborg University, Denmark
Christian S. Jensen (codirector), Michael H. Böhlen, Renato Busatto, Curtis E. Dyreson, Heidi Gregersen, Dieter Pfoser, Simonas Šaltenis, Janne Skyt, Giedrius Slivinskas, Kristian Torp

University of Arizona, USA
Richard T. Snodgrass (codirector), Sudha Ram

Individual participants
Anindya Datta, Georgia Institute of Technology, USA
Kwang W. Nam, Chungbuk National University, Korea
Mario A. Nascimento, State University of Campinas and EMBRAPA, Brazil
Keun H. Ryu, Chungbuk National University, Korea
Michael D. Soo, University of South Florida, USA
Andreas Steiner, TimeConsult, Switzerland
Vassilis Tsotras, Polytechnic University, USA
Jef Wijsen, Vrije Universiteit Brussel, Belgium

For additional information, see The TIMECENTER Homepage:
URL: <http://www.cs.auc.dk/research/DBS/tdb/TimeCenter/>

The TIMECENTER icon on the cover combines two "arrows." These "arrows" are letters in the so-called *Rune* alphabet used one millennium ago by the Vikings, as well as by their precedessors and successors. The Rune alphabet (second phase) has 16 letters, all of which have angular shapes and lack horizontal lines because the primary storage medium was wood. Runes may also be found on jewelry, tools, and weapons and were perceived by many as having magic, hidden powers.

The two Rune arrows in the icon denote "T" and "C," respectively.

Contents

1 Introduction

Extensible Markup Language (XML) [1] is a simple, very flexible text format derived from SGML [7]. Although originally designed to meet the requirement of large-scale electronic publishing, XML is rapidly becoming an important medium for representing and exchanging data. XML documents are suitable to describe both structured and semi-structured data. An XML document has a hierarchical structure. Pairs of tags carrying semantic meanings are used to delimit data, but whether some data would appear or the sequence that data appears can be very flexible.

As data changes over time, the corresponding XML document needs to be updated to reflect the change, and it is often preferable to preserve the history of the data for analyzing or avoiding loss of critical information. For example, stores want to keep the sale records in different periods to analyze customers' shopping patterns. Every document that reflects the stable status of the data during a period of time is called a version. How to effectively manage multi-version XML documents is important in many application domains, such as data warehousing and document archives. Some effort has been put in developing systems to manage multi-version XML documents. A simple approach is to store each version as a separate document. But this is not desirable because it exhibits too much redundancy and is difficult to keep the data consistency between versions. In the edit-based approach [5], one complete version is kept and the successive versions can be represented by applying edit scripts to that version. This approach effectively reduces redundancy, but it doesn't support general queries very well because additional computation is needed to retrieve actual data at the moment.

On the other side, τXSchema [2] provides another solution for version control. It defines two types of annotations, *logical annotation* and *physical annotation*, as supplements to an XML schema to make an XML document time-sensitive. Logical annotation denotes which parts of the document can evolve over time. Physical annotation denotes the places where the temporal information is added to the document. An XML document described by these two annotations is called a *temporal XML document*. A temporal XML document is physically a single document, but conceptually it can be seen as the merger of multiple documents. Each of those documents describes the state of the data at one moment. Temporal XML documents use timestamps to distinguish data that is valid at different periods by timestamping the data. Two significant enhancements in a temporal XML document brought by τXSchema are *time-varying elements* and *time-varying attributes*. A time-varying element is an element that has a child element of `timestamp` which states the valid period of the element. A time-varying attribute takes the form of an element with the fixed name `timevarying-attr`, but it is an attribute in concept. The attribute's name, value and valid period are given as the attributes of the `timevarying-attr` element.

The following is an example of temporal XML document. Figures 1, 2 and 3 provide three XML documents that contain the information about a university employee. They show the person received a promotion on 1/1/1997 and two wage raises on 1/1/1997 and 1/1/1998. The same information is depicted in the temporal XML document in Figure 4. Note attribute `Title` is converted to time-varying attribute and timestamps are added to element `Salary` to denote their valid periods.

```
<Bob Title="Assistant Provost">
      <DateofBirth>1945-05-09</DateofBirth>
      <Salary>40000</Salary>
  </Bob>
```

Figure 1: Snapshot 1 is valid from 1/1/1995 to 12/31/1996

```
<Bob Title="Provost">
       <DateofBirth>1945-05-09</DateofBirth>
       <Salary>50000</Salary>
   </Bob>
```

Figure 2: Snapshot 2 is valid from 1/1/1997 to 12/31/1997

```
<Bob Title="Provost">
       <DateofBirth>1945-05-09</DateofBirth>
       <Salary>55000</Salary>
   </Bob>
```

Figure 3: Snapshot 3 is valid from 1/1/1998 to 12/31/2002

```
<Bob>
       <timestamp vtbegin="1995-1-1" vtend="2003-1-1"/>
       <timevarying-attr name="Title" value="Assistant Provost"
           vtbegin="1995-1-1" vtend="1997-1-1"/>
       <timevarying-attr name="Title" value="Provost"
           vtbegin="1997-1-1" vtend="2003-1-1"/>
       <DateofBirth>1945-05-09</DateofBirth>
       <Salary>
           <timestamp vtbegin="1995-1-1" vtend="1997-1-1"/>
           40000
       </Salary>
       <Salary>
           <timestamp vtbegin="1997-1-1" vtend="1998-1-1"/>
           50000
       </Salary>
       <Salary>
           <timestamp vtbegin="1998-1-1" vtend="2003-1-1"/>
           55000
       </Salary>
   </Bob>
```

Figure 4: Temporal XML document: valid from 1/1/1995 to 12/31/2002

Temporal XML document is suitable to store time-varying information. Temporal XML document is self-explaining in that it stores time information along with data. Time-varying element and time-varying attribute are capable of recording the evolution of the data. Data redundancy is avoided because only changed data is kept.

Most of the work on multi-version XML document focuses on detecting and representing changes between versions and querying multi-version XML documents. However, we believe a well-defined set of APIs specifically designed for accessing temporal XML document is equally important. Such APIs could make it much easier for user to manage time-varying data stored in temporal XML document because user can focus on data while leaving the APIs to deal with temporal information. Those APIs should be capable of fulfilling common tasks that could be applied to multi-version XML documents. They should be general enough to be able to work with different models, too.

A widely adopted programming API for non-temporal XML is the Document Object Model (DOM) [3]. DOM regards an XML document as a tree. The root is the document. Internal nodes are elements that have children. Leaf nodes are elements that don't have children and other components of XML document, such as text. DOM defines an interface to access every kind of component of XML document. It allows reading and writing valid and well-formed XML documents through those interfaces. It defines the logical structure of documents and the way a document is accessed and managed.

Programs based on standard DOM can access, change, delete, or add to the data encoded in an XML document, but this is not sufficient for a temporal XML document. Temporal XML documents contain the history of the data, so it can provide multiple views of data to user if different temporal constraints are specified. Operations on temporal XML documents should take the temporal constrains into account. Standard DOM needs an extension to provide such function. We add some new methods to the DOM-defined Java interfaces and create some new Java interfaces to make DOM time-sensitive. We call the augmented DOM API τDOM.

The rest of this report is arranged as follows. We describe our considerations in designing τDOM in Section 2 and the extensions of τDOM on DOM in Section 3. In Section 4 we discuss some implementation issues of τDOM. In Section 5 we outline future work. In the appendices we give a detailed description of τDOM API as well as examples of using τDOM API to manage temporal XML documents.

2 τDOM Design Considerations

Because temporal XML documents contains both data and the temporal information of data, the effect of temporal information on operations must be taken into account during τDOM's design. Here we describe some of our considerations when designing τDOM.

2.1 Modes

A temporal XML document contains the complete history of data in that document, but users may only hope to extract part of the history that contains the data they need. Users should be able to set the time range of data they want to retrieve by giving a temporal condition, such as a sub-period of the whole period, with τDOM exposing only the data that meet the temporal condition to users. Borrowing from work in temporal databases [6], we envision there are three possible scenarios in which the user may work: (i) catch a snapshot of data at one moment, (ii) examine the mutation of data during a period, or (iii) work directly with document as represented. Correspondingly we devise three working modes in τDOM. These modes are *current mode*, *sequenced mode* and *representational mode*. Let's see how different the temporal XML in Figure 4 looks like in different modes.

In current mode, the user chooses a time instant, then only the data valid at that instant is visible to the user. We call this instant *the current instant* in τDOM. Those data that is not valid at the current instant is screened from the user automatically by τDOM. So if the user chooses to see the data valid on 3/1/1996 in Figure 4, he will see that element `Bob` has one attribute `Title`, one child element `DateofBirth`, and another child element `Salary`. The `Salary` element has a child text node with the value of `40000`. In this case, the user feels that he is reading the traditional XML document in Figure 5.

```
<Bob Title="Assistant Provost">
       <DateofBirth>1945-05-09</DateofBirth>
       <Salary>40000</Salary>
  </Bob>
```

Figure 5: Snapshot of Figure 4 on 3/1/1996

In sequenced mode, users chooses a period of time by giving two instants, then τDOM only presents to users that data that is valid within that period. We call the period *the period of applicability* in τDOM. The period of applicability begins from the earlier instant and ends just before the later instant. Data that becomes valid after the period is not considered valid within the period of applicability. If the user choose a period of applicability from 6/1/1995 to 6/1/1998 in the document in Figure 4, he will still see element `Bob` has a child element `DateofBirth` and an attribute `Title`, but this attribute had value `Assistant Provost` from 6/1/1995 to 12/31/1996 and value of `Provost` from 1/1/1997 to 6/1/1998. This is a significant difference between τDOM and DOM. In DOM, an attribute can only have one value, while in sequenced mode in τDOM, an attribute can have different values in different periods. Another difference is with the child element `Salary`. It has a child text node with two different values valid at different periods in sequenced mode. We describe our solution to these differences in the next section.

When τDOM works in representational mode, it functions the same as DOM. In representational mode, all the content of the temporal XML document is presented to the user in its original form. So the user will see element `Bob` has six child elements `timestamp`, `timevarying-attr`, `timevarying-attr`, `DateofBirth` and `Salary`, but no attributes, in Figure 4.

2.2 Logical Structure versus Physical Structure

As we mentioned before, DOM regards an XML document as a tree. Temporal XML documents should also be presented to the user as a tree under τDOM. τDOM gives the user the logical structure of the document, while DOM provides the physical structure. τDOM constructs the logical structure of a temporal XML document from its physical structure. The physical structure contains all data in the temporal XML document, but τDOM only extracts, transforms and exposes the valid data as specified by the mode under which that data is accessed. In addition, τDOM should present data in an appropriate form to users. This form may be different from the form in which it is recorded in the document. For example, time-varying attributes are recorded using `timevarying-attr` elements in temporal XML documents, but users should see them as attributes. The τDOM tree and The DOM tree of the same temporal XML document can be significantly different, especially in sequenced mode. Figure 6 and Figure 7 show DOM's and τDOM's trees for the temporal XML document in Figure 4 when τDOM works in sequenced mode and the period of applicability is from 1995-01-01 to 1999-01-01. The attribute nodes in the physical structure are omitted to save space.

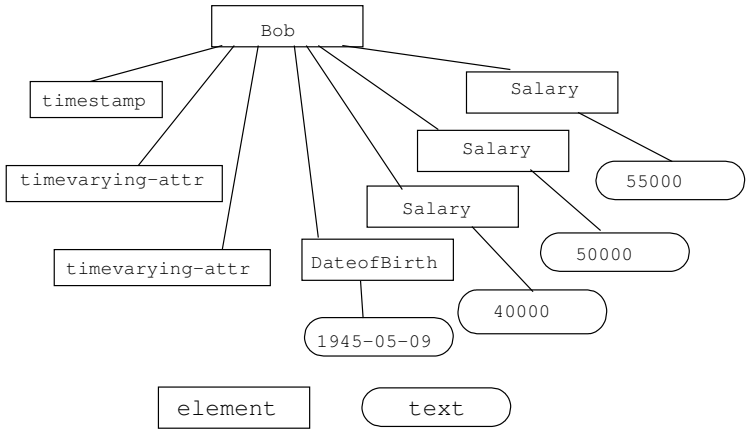

Figure 6: Physical structure of Figure 4

Comparing these two trees, we can see there are four reasons for τDOM to remodel the DOM tree to adapt to users' need. First, components in the DOM tree that keep the temporal information are not presented as such to users in the τDOM tree. For example, the `timestamp` element in the document is only used by τDOM to judge if some data is valid under the temporal constraint and so should be present to the user. Second, τDOM

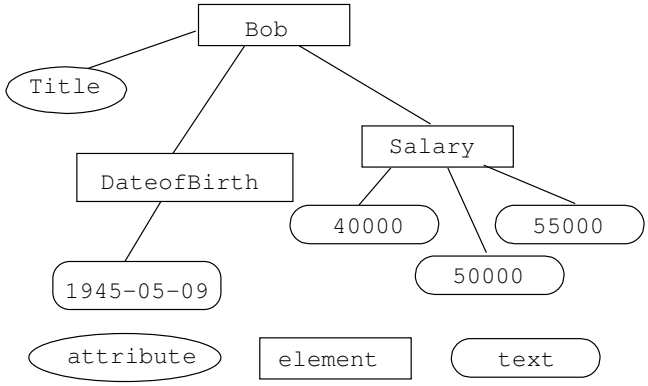

Figure 7: Logical structure of Figure 4

only allows the user to see the data that is valid under the stated temporal constraint, so the components of the DOM tree that are not valid under the temporal constraint have to be discarded in the τDOM tree. In the above example, the salary information after 1/1/1999 is removed from the tree since it is out of the period of applicability. Next, temporal XML documents have time-varying attributes. Time-varying attributes are recorded as elements in temporal XML documents, but they are actually attributes. τDOM needs to wrap up time-varying attribute elements to make them look like attributes to users. For example, in the above τDOM tree the attribute node `Title` represent two `timevarying-attr` element nodes in the DOM tree. Last, temporal XML document keeps data in a long period. If some data changes, a new node is allocated in the DOM tree to represent the new value. But in τDOM, changed data should be grouped because they represent the same entity. So in the τDOM tree above, there is only one `salary` element instead of three in the DOM tree.

τDOM is designed in such way that the physical structure is invisible to the user who don't want to see this structure. First, when the user wants to retrieve data from the document, τDOM compares timestamps of data with the temporal constraint. Only valid data is imported into the τDOM tree. Second, τDOM automatically identifies related components in the document and groups them together. Through τDOM, the user only see an aggregate entity representing those related component. Operations on the aggregate entity are mapped to appropriate operations on the corresponding components in the document by τDOM. For example, if an entity is removed in sequenced mode, τDOM removes all its components in the document during the period of applicability. Third, τDOM unifies time-varying attribute elements and regular attributes so that they have the same behavior to the user. The user uses attributes without needing to care about the actual form of the attributes in the document. For example, if user changes the value of an attribute in sequenced mode, τDOM finds the corresponding nodes in the document and changes its value in different ways according to the type of nodes. If node is a regular attribute node, it is given a new value. If node is `timevarying-attr` element, new value is given to its `value` attribute. In addition, if the change only applies to part of a regular attribute node's lifespan, that attribute node is transformed into several time-varying attribute elements so that some time-varying attribute elements keep old value of the period out of the period of applicability and the others have the new value within the period of applicability.

2.3 Modes Reexamined

In representational mode, the raw content of the document is presented to user and τDOM just directs the user's calls to DOM's methods. In current mode, the user operates on a virtual document that contains data valid at a given instant. Data valid at an instant only exhibits a single content, so DOM's methods are capable of handling tasks under current mode. But in sequenced mode, we need to deal with multiple values at different times. Sequenced mode exhibits two remarkable aspects that cannot be handled only with DOM's methods. First, a time-varying attribute might have multiple values in sequenced mode. Second, different snapshots might exhibit different data content. We address the first problem by extending DOM's `Attr` interface to `TAttr`. `TAttr` has a new method to return multiple values of time-varying attributes. We describe that method in next section. Our

solution to the second problem involves the concept of *versioned element* in τDOM.

Because τDOM deals with data during a period in sequenced mode and the data may change in that period, τDOM needs to inform the user of the change. This requirement is beyond the capability of DOM's `Element` methods, which assumes the content of an entity is static. τDOM includes a new kind of element, *versioned element*, in addition to the element inherited from DOM. A versioned element represents an entity during the whole period of applicability, while an element in τDOM represents a consistent view of the entity in some sub-periods of the period of applicability. A versioned element can exhibit multiple views in the period of applicability because its content changes during the period of applicability.

```
<People>
      <timestamp vtbegin="1999-01-01" vtend="2000-01-01"/>
      <Bob>
      ......
      </Bob>
      <Jane>
      ......
      </Jane>
  </People>
  <People>
      <timestamp vtbegin="2000-01-01" vtend="2001-01-01"/>
      <Jane>
      ......
      </Jane>
      <Bob>
      ......
      </Bob>
  </People>
```

Figure 8: One `People` entity in two periods

Figure 8 illustrates why versioned element is necessary in τDOM. In this document there are two `People` elements representing the same `People` entity in different periods. In sequenced mode, the user should only see one `People` element, which contains the content of those two `People` elements in Figure 8. To distinguish two kinds of `People` elements here, we call the element we see in sequenced mode the *sequenced People element* and the element actually appearing in the document, the *representational People element*.

Some methods that τDOM inherits from DOM, such as `getChildNodes()`, `getFirstChild()`, etc, require that the returned child nodes are arranged in a determined order. In DOM, one node corresponds to one item in the document, so DOM just returns the child nodes in the same order as they appear in the document. τDOM tries to group related item in the document and presents users only a single node, so sometimes the ordering of nodes is not apparent. For example, in Figure 8, although both representational `People` elements have two child elements: `Bob` and `Jane`, they appear in different orders. τDOM can't determine the ordering of `People`'s children and it is better to inform the user of the different children orderings and allow the user to choose which one is preferred. In addition to the ordering of children, an entity may have some attribute during one period, but doesn't have that attribute in another period. To accommodating such structure evolution of entities, we introduce *version* in τDOM. A version of an entity represents a stable structure of that entity during a period. Any DOM method on a version returns a consistent result. A sequenced element can have one or more versions. One or several representational elements determine a version if and only if (i) they have the same number of child nodes, (ii) their child nodes at the same positions represent the same entities, and (iii) they have the same attributes. In Figure 8, the sequenced `People` element has two versions. In the first version, `Bob` is its first child and is followed by `Jane`. In the second version, the sequence of `Bob` and `Jane` is reversed.

3 API Extension

In this section we introduce τDOM's extensions to the DOM API.

3.1 The `TDocument` interface

The `TDocument` interface contains three distinct methods for setting representational mode, current mode and sequenced mode. When setting current mode or sequenced mode, the user needs to provide the time format information so that τDOM can parse the time string in timestamps, time-varying attributes, etc. The three methods to set mode are as follows.

`setModeCurrent()` The user calls this method to make τDOM work in current mode. It takes two arguments. The first argument is a format string describing the format of the time information used in the document and by the user. τDOM needs to know the format because the time information in a temporal XML document is expressed using strings. The format string should adhere to the rules set in JAVA [8] for the `SimpleDateFormat` class. Figure 9 shows the legal letters that can appear in format string and their meanings:

Letter	Date or Time Component	Presentation	Examples
`G`	Era designator	Text	`AD`
`y`	Year	Year	`1996`; `96`
`M`	Month in year	Month	`July`; `Jul`; `07`
`w`	Week in year	Number	`27`
`W`	Week in month	Number	`2`
`D`	Day in year	Number	`189`
`d`	Day in month	Number	`10`
`F`	Day of week in month	Number	`2`
`E`	Day in week	Text	`Tuesday`; `Tue`
`a`	Am/pm marker	Text	`PM`
`H`	Hour in day (0-23)	Number	`0`
`k`	Hour in day (1-24)	Number	`24`
`K`	Hour in am/pm (0-11)	Number	`0`
`h`	Hour in am/pm (1-12)	Number	`12`
`m`	Minute in hour	Number	`30`
`s`	Second in minute	Number	`55`
`S`	Millisecond	Number	`978`
`z`	Time zone	General time zone	`Pacific Standard Time`; `PST`; `GMT-08:00`
`Z`	Time zone	RFC 822 time zone	`-0800`

Figure 9: Pattern letters of format string

The second argument is also a string. It can be interpreted as a time instant given its format declared in the first argument. By this the user requires τDOM to extract a snapshot of the data at that instant. After setting the mode to current mode, the user can use DOM's API to read the document. All the information returned to the user is valid at that instant. Ineligible information is automatically suppressed by τDOM. The user won't see the existence of the `Time-Stamp` element and he can access a time-varying attribute like a normal attribute even it has the form of element in the document.

`setModeSequenced()` The user calls this method to make τDOM work in sequenced mode. This method takes three arguments. The first argument gives the date format as in `setModeCurrent()`. The second argument and the third argument specify the starting instant and the ending instant of a time interval, respectively. The user wants to retrieve data that is valid in that interval. The starting instant is included, and

the ending instant is excluded, so if the user wants to retrieve data of year 1996, the starting instant and the ending instant can be `1-1-1996` and `1-1-1997` respectively. Sequenced mode is more complex to deal with because snapshots at different moment during an interval may have different contents. It is challenging to combine information from multiple snapshots and provide an effective way to user to manage it. To support this functionality, DOM's `Element`, `Attr` and some other interfaces are enhanced with methods used only in sequenced mode.

`setModeRepresentational()` The user calls this method to make τDOM work in representational mode. It has no arguments. In representational mode, τDOM works exactly like DOM and the temporal XML document is just treated as a normal XML document. Timestamps and time-varying attributes, which have special meaning in temporal XML, are just common elements to the user in this mode. This mode gives the user maximal freedom to manipulate the temporal XML document, but it also brings the potential danger of destroying the integrity of the data because the user may erroneously violate the temporal dependencies among elements and attributes.

3.2 The `VersionedElement` and `Element` interfaces

In sequenced mode, an instance of `VersionedElement` denotes multiple representational elements that represent a single sequenced element and multiple versions in those representational elements. `VersionedElement` has methods for getting the number of its versions and a specified version, which is an `Element` instance. One version of sequenced element is an instance of DOM's `Element` interface. It denotes an child nodes ordering exhibited by some representational elements. When working in sequenced mode, the user always gets a `VersionedElement` instance first. To access an element's child nodes, the user calls `getVersion()` of `VersiondedElement` and specifies the index of the desired version. `getVersion()` returns an instance of `Element` whose child nodes has a fixed ordering. Then users can access child nodes by calling `getChildNodes()`, `getFirstChild()` or `getLastChild()`. The relations between `VersionedElement` and `Element` in Figure 8 is illustrated in Figure 10.

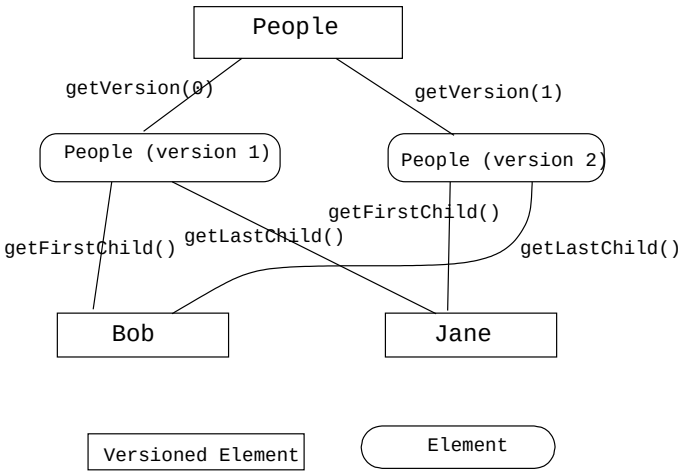

Figure 10: Diagram of `VersionedElement` and `Element` in sequenced mode

Two additional new methods of `VersionedElement` are `getPreviousSiblingInParentVersion()` and `getNextSiblingInParentVersion()`. These methods return the sibling of a `VersionedElement` in a specified version of its `VersionedElement` parent. These methods are needed because an `VersionedElement` sits at different positions in different versions of its parent. For example, in Figure 8, the next sibling of `Bob` in the first version of its parent `People` is `Jane`, while in the second version of `People`, `Bob` doesn't have a next sibling.

3.3 The `TAttr` and `SequencedValue` interface

We extend DOM's `Attr` interface to `TAttr` to allow retrieval of a time-varying attribute's multiple values in sequenced mode. `TAttr` has a new method `getSequencedValue()`. `getSquencedValue()` returns an instance of `SequencedValue`, which provides the value and the temporal information about the attribute during the period of applicability. `getSequencedValue()` replaces `Attr`'s method `getValue()` in sequenced mode. The user should always call `getSequencedValue()` and then use the `SequencedValue` to get values of attributes in sequenced mode. The `getValue()` method is still kept in `TAttr` interface to keep compatibility with DOM's `Attr` interface, but it always returns `null`.

The values of a time-varying attribute are kept in a `SequencedValue` in sequenced mode in time order. Using `SequencedValue`, the user can call `getLength()` to find out how many values the attribute has during the period of applicability, call `getValue()` to get a value by its index, or call `getTimestamp()` to get the timestamp of a value by its index.

3.4 Other Extensions

We also add the following new interfaces in τDOM.

`Period` This class defines a contiguous time. It remembers its starting instant and its ending instant. The period is defined from the starting instant to the instant just before the ending instant. `Period` is used to save timestamps of entities in τDOM. For example, a `Period` can denote during what time an attribute has a specific value.

`PeriodSet` This class contains multiple `Period`s. These `Period`s are not contiguous and are ordered by time. `PeriodSet` is used to save timestamps of time-varying elements in sequenced mode if the element are valid in several disjoint periods.

`TDOMException` τDOM throws `TDOMException` if it sees an illegal operation. For example, `TDOMException` is thrown if the user gives an ending instant that is before the starting instant when calling `setModeSequenced`.

4 τDOM Implementation

Developing a τDOM implementation is a challenging task. Although there are many mature DOM implementations on which τDOM can be developed, the implementation of τDOM still involves more than 6,000 lines of code. Among the available DOM implementations, we chose Apache's Xerces [10] and developed our τDOM implementation on it. We now discuss some implementation issues.

4.1 Implementation of Representational Mode

The implementation of representational mode is trivial because τDOM is equivalent to DOM in representational mode, so τDOM can completely reply on DOM. The only difference between τDOM and DOM in representational mode is that at the document level, users use τDOM's `TDocument` instance instead of DOM's `Document` instance. `TDocument` instance embeds a DOM's `Document` instance that contains all the content of a temporal XML document. Calls to `TDocument` are directed to the corresponding `Document`'s methods and the results are returned to users without any changes, so under the document level, the user interacts directly with the DOM implementation. Calls to `TDocument`'s methods that are specifically for current or sequenced modes would make τDOM throw a `TDOMException` to remind the user that τDOM is working in representational mode.

4.2 Implementation of Sequenced Mode

In the implementation of sequenced mode and current mode, the temporal constraint must be taken into account and be compared with the lifespan of components in a document. All the components in a temporal XML document have their *lifespan*. The instant when a document becomes valid is denoted by an attribute of the document element. The document has a lifespan from that instant to forever. The lifespan of an element is denoted

by its `timestamp` child element. If it doesn't have `timestamp` child element, it has the same lifespan as its closest ancestor element that has a `timestamp` child element. If no such ancestor exists, the element has the same lifespan as the document. The lifespan of an attribute in the form of `timevarying-attr` is given explicitly by `vebegin` and `vtend` attributes. Normal attributes have the same lifespan as its owner elements. Other components have the same lifespan as their parent elements. In τDOM operations, lives of components must be considered in order to provide users only valid data and keep temporal consistency. For example, a child can not have a lifespan that is beyond the lifespan of its parent.

Our implementation of sequenced mode uses three classes that implement the interfaces `VersionedElement-Impl`, `SequencedElementImpl` and `SequencedAttrImpl`.

Class `VersionedElementImpl` implements interface `VersionedElement`. It represents element entities in sequenced mode. Inside a `VersionedElementImpl`, there is a list of representational elements in the document that reflect the status of the same entity in the period of applicability. When constructing a `VersionedElement`, τDOM identifies related representational elements, discards those that are not valid in the period of applicability and puts valid ones in the list in the order of their lives. In sequenced mode, some operations need to be considered very carefully to limit the change to the period of applicability and to keep temporal consistency. For example, Figure 11 illustrates the process of an appending operation. Here the valid content of child in the period of applicability is moved from its old parent to the new parent. The rectangles denote the lifespan of old parent, child and new parent.

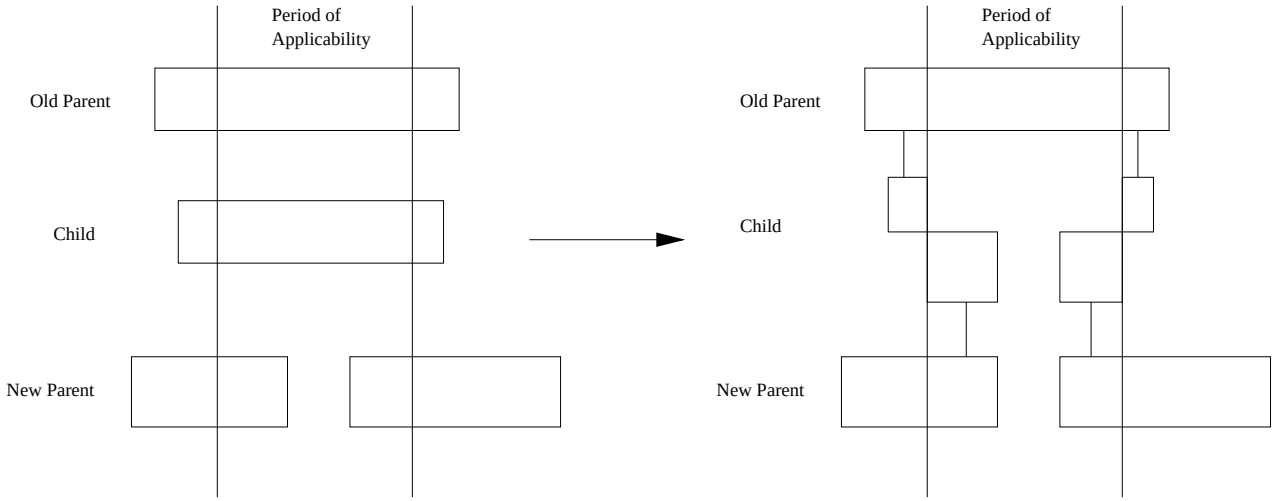

Figure 11: Appending child in sequenced mode

When temporal information is considered, the process involves the following steps,

1. First, the child is examined. Its content that is valid in the period of applicability is extracted and detached from the old parent. In Figure 11, because child's lifespan crosses the beginning and the end of the period of applicability, the child is split into three parts which keep content before, within and after the period of applicability respectively. The parts before and after period of applicability are left with the old parent, while the part within the period of applicability is cut off from the old parent. In this step, τDOM ensures that only data valid in the period of applicability is touched.

2. Next, the extracted content of the child is checked against new parent's lifespan. The content that is valid within the lifespan of new parent's representational element is appended to that representational element. Content that is not valid within any lifespan of new parent's representational elements is discarded. In this step, τDOM ensures that the temporal consistency is not violated, that is, the lifespan of children is always within the lifespan of their parent.

Class `SequencedElementImpl` implements interface `Element`. In sequenced mode, `SequencedElementImpl` is always affiliated with a `VersionedElementImpl`. An `SequencedElementImpl` contains a list of some representational elements of its `VersionedElementImpl`. Those representational elements belong to the same version of the `VersionedElementImpl`. Most calls to `SequencedElementImpl`, such as `getAttributeNode()`, `appendChild()`, etc, are directed to the `VersionedElementImpl`, so such calls have the same effect on `VersionedElementImpl` and its `SequencedElementImpl`. `SequencedElementImpl` only take care of those operations that depend on the ordering of children, such as `getChildNodes()`. When `getChildNodes()` is called, `SequencedElementImpl` retrieves the children of the `VersionedElementImpl`, arranged them in a list in the order defined by the version, and return the list.

Class `SequencedAttrImpl` implements interface `TAttr`. It represents an attribute in sequenced mode. A `SequencedAttrImpl` remembers its owner, which is a `VersionedElementImpl`, and the name of the attribute. `SequencedAttrImpl` accesses the attribute dynamically, that is, when a `SequencedAttrImpl` is created, it doesn't know where the attribute appears in the owner. When `SequencedAttrImpl`'s methods are called, `SequencedAttrImpl` searches all the representational elements of its owner for the appearances of the attribute, which can be a normal attribute or a time-varying attribute element, and applies the operation on the fly.

4.3 Implementation of Current Mode

At first glance, current mode seems quite different from sequenced mode and easier to implement because current mode only provides a snapshot of data to the application. Since all the data is valid at that current instant, we don't need to consider the complex temporal dependency between entities in operations that are present in sequenced mode. But one thing we try to maintain in current mode is *temporal upward compatibility* (TUC).

Current mode looks like sequenced mode under TUC. We consider current mode as a special case of sequenced mode where the period of applicability is from the current instant to forever. An additional requirement is that the accessible entities in current mode must be valid within a contiguous period beginning from the current instant, so current mode actually needs more check on the validity of data than sequenced mode.

Changes to the status at the current instant have a lasting effect on the status in the future. Figure 12 shows the effect of TUC in changing the value of an attribute in current mode. The lifespan of the attribute is from `t1` to `t3` and then continues from `t4`. The current instant is set to `t2`. In TUC, the attribute should have the new value from `t2` to `t3`, while the old value is kept before `t2` and after `t4`.

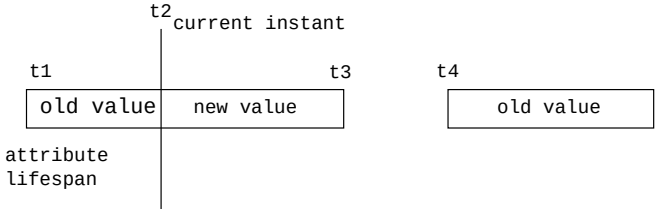

Figure 12: TUC in current mode

The implementation of current mode depends on two classes, `CurrentElementImpl` and `CurrentAttrImpl`. `CurrentElementImpl` contains a list of related representational elements ordered by their lifetimes. The first representational element is valid at the current instant. There is no gap between the lifetime of one representational element and the lifetime of its successor in the list. When reading the child nodes of a `CurrentElementImpl`, its children are arranged in the order as they appear in the first representational element. When modifying a `CurrentElementImpl`, changes are applied to all the representational elements in the list to keep TUC.

`CurrentAttrImpl` represents an attribute in current mode. It remembers its owner, which is a `CurrentElementImpl`, and the name of the attribute it represents. When reading the value of an attribute, `CurrentAttrImpl` returns the value of the attribute of the first representational element of its owner. When setting the value of an attribute, `CurrentElementImpl` changes all the appearances of the attribute, which can be time-varying attribute elements or regular attributes.

4.4 Use of DOM's *Traversal Interface*

When dealing with a temporal XML document, we need to differentiate `timestamp` elements and `time-varying-attr` elements from other elements because these two kinds of elements have to be transformed before being presented to users. To do this, we have to extract them from a representational element's children. We found DOM Level 2 *Traversal and Range interface* (TS) [4] is suitable for this work. TS allows users to choose a node as root and provide a customized filter. When using TS to traverse the sub-tree rooted at the node, the sub-nodes in the tree are sent to the customized filter first. The filter can decide if the sub-node should be seen by users. By implementing different filters, we can retrieve only the information we need from the document. The example in Figure 13 shows how to extract an element's timestamp.

```
TreeWalker tw;
   tw = ts_doc.create
   TreeWalker(root,
              NodeFilter.SHOW_ELEMENT,
              new TSFilterIn(),
              false);
   Element ts = (Element)tw.firstChild();
```

Figure 13: TreeWalker Usage

The customized filter here is `TSFilterIn`. `TSFilterIn` checks the name of the node given to it and only accepts `timestamp` elements. `TSFilterIn` eases the work of retrieving `timestamp` element. We developed several classes used as customized filters.

- `CurInstFilter` Retain the nodes that are valid at the instant in current mode.
- `SeqExtFilter` Retain the nodes that are valid during the period in sequenced mode.
- `TS_TAttrFilterOut` Retain the nodes other than those representing timestamps or time-varying attributes.
- `TS_FilterIn` Retain the nodes that represent timestamps.
- `TVAttrFilterIn` Retain the nodes that represent time-varying attributes.

4.5 Recognizing the Representational Components of Entity

As we saw before, multiple representational elements can be related to the same entity in a temporal XML document. One question is how to correctly group such representational elements. τDOM maintains a special attribute `verid` of representational element for this purpose. Representational elements of the same entity has the same and unique value of `verid` attribute. The attribute `verid` is under the control of τDOM and is invisible to user. Suppose that initially an entity has only one representational element in the document; no `verid` attribute is given to the representational element. Later, the content of the entity is changed. Two representational elements are then needed for the entity: one holds the content before the moment when the change happens, and the other keeps the content after the moment. The splitting is done by τDOM when doing the modification. During the splitting process, τDOM utilize the `verid` attribute to indicate that those two representational element should be grouped together. τDOM adds the `verid` attribute to those two representational elements and gives it a value that is unique in the document.

4.6 Code Statistics

The implementation of τDOM consists of more than 7,000 lines of Java code. The implementation is done on the DOM implementation of Xerces, which consists of about 28,000 lines of Java code. Many of the methods in DOM API are given new semantics in current and sequenced mode and they are completely rewritten in τDOM.

5 Conclusion and Future Work

The report summarizes our work in developing the τDOM API. τDOM provides a common language that can be used to access temporal data stored in XML documents. Our design focus on making minimum extensions to DOM API to support temporal operations. τDOM add only a few new methods to interfaces in DOM, but these methods are capable of fulfilling most of users' tasks with temporal XML documents. A lot of effort was put into implementing τDOM APIs. The most difficult aspect of the implementation is to correctly understand the effects of operations under the three modes and to ensure consistency in all situations.

τDOM implements most of DOM's methods. However, one method of DOM conflicts with the nature of temporal XML document and is still under consideration. Specifically, method `getElementById()` of `Document` interface assumes that a unique id is possessed by only one element, but in temporal XML document, an entity can have multiple representational elements and they share the same id. This problem should be addressed in τSchema [2] first and later in τDOM. We are also considering adding more functions to τDOM implementation, such as making `SequencedValue`, `PeriodSet` and `PeriodSet` serializable and clonable, allowing multiple disjointed periods as the period of applicability so that the user can manage the data of different periods at the same time.

In the current τDOM design, we only provide a basic set of methods for managing temporal XML documents. Those methods can be combined to fulfill complex tasks. After we have a better understanding of usage patterns in accessing and modifying temporal data, we can devise some new APIs which can address users' needs more directly.

We are considering improving τDOM's performance in our implementation. DOM is resource-consuming because it needs to read the whole document to construct its inner document tree. τDOM has the potential to use less resources than DOM because in current mode and sequenced mode, not all content in document are accessible, so some of resources can be freed. We plan to experiment further with our implementation.

Finally, we are considering new representations of temporal XML documents. The current structure of a temporal XML document cannot completely remove content redundancy because a child lifespan is strictly within its parent's lifespan, so if some content changes and a new representational component is needed, other content has to be duplicated in the new component although it is not changed. Another way to organizing temporal data is to store data by entity. Each item contains all the data of an entity. The relationship between entities can be denoted by links. In this way, when creating new representational components, only links, not data itself, need to be duplicated.

References

[1] T. Bray, J. Paoli, C. M. Sperberg-McQueen and E. Maler, "Extensible Markup Language (XML) 1.0 Second Edition (W3C Recommendation)," 6 October 2000.

[2] F. Currim, S. Currim, C. E. Dyreson and R. T. Snodgrass, "Effecting Data Independence for Temporal XML Schemas," in preparation.

[3] A. L. Hors, P. L. Hgaret, L. Wood, G. Nicol, J. Robie, M. Champion and S. Byrne, "Document Object Model Level 2 Core (W3C Recommendation)," 13 November 2000.

[4] J. Kesselman, J. Robie, M. Champion, P. Sharpe, V. Apparao and L. Wood, “Document Object Model Level 2 Traversal and Range (W3C Recommendation),” 13 November 2000.

[5] A. Marian, S. Abiteboul, G. Cobéna and L. Mignet, “Change-Centric Management of Versions in an XML Warehouse”, The *VLDB Journal*, p.581–p.590, 2001.

[6] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner, “Transitioning Temporal Support in TSQL2 to SQL3”, Temporal Databases: Research and Practice, O. Etzion, S. Jajodia, and S. Sripada (eds.), Springer, pp. 150–194, 1998.

[7] ISO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML).

[8] SUN JavaTM 2 SDK, Standard Edition Documentation, Version 1.4.1.

[9] XBench – A Family of Benchmarks for XML DBMSs, `http://db.uwaterloo.ca/ ddbms/projects/xbench/`

[10] Xerces2 Java Parser, `http://xml.apache.org/xerces2-j/index.html`

Appendices

A τDOM API

Here we give the detailed descriptions of τDOM API. In representational mode, almost all the calls are directed to the corresponding methods of the underlying DOM implementation, except for some new methods in `TDocument`. We feel that not all methods are applicable to all interfaces. For example, `TDocument` doesn't have siblings, so it is meaningless to call `getNextSibling()` on `TDocument`. When those inapplicable methods are called, they return `NULL` directly.

A.1 Extensions to the `Document` Interface

Methods specific to `TDocument`:

Methods	Representational Mode	Current Mode	Sequenced Mode
`void SetModeRepresentational()`	Set mode to `MODE_REPRESENTATIONAL`.		
`void setModeCurrent (String pattern, String current)`	Set mode to `MODE_CURRENT`. Take date format and a string denoting the current instant as arguments. Throw `TDOMException` with code `INVALID_DATE_FORMAT` if the string cannot be parsed using the date format.		
`void setModeCurrent (String pattern, Date current)`	Set mode to `MODE_CURRENT`. Take date format and current instant as arguments.		
`void setModeCurrent (String pattern)`	Set mode to `MODE_CURRENT`. Take date format as argument. The current instant is the time of the system.		
`void setModeSequenced (String pattern, String begin, String end)`	Set mode to `MODE_SEQUENCED`. Take date format and two strings denoting the beginning and ending instants of the period of applicability as arguments. Throw `TDOMException` with code `INVALID_DATE_FORMAT` if the given strings cannot be parsed using the date format.		
`void setModeSequenced (String pattern, Period seq_period)`	Set mode to `MODE_SEQUENCED`. Take date format and a period denoting the period of applicability as arguments.		
`void setModeSequenced (String pattern)`	Set mode to `MODE_SEQUENCED`. Take date format as argument. The sequential applicability is from system's current time to forever.		
`int getMode()`	`Return MODE_REPRESENTATIONAL.`	`Return MODE_CURRENT.`	`Return MODE_SEQUENCED.`
`String getDateFormat()`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Return the date format description.	
`Date getCurrentInstant()`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Return the current instant of current mode.	Throw `TDOMException` with code `INAPPRO_TIME_MODE`
`Period getSequencedExtent()`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Return the period of applicability of sequenced mode.
`Period createVersionedElement (String name)`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Throw `TDOMException` with code `INAPPRO_TIME_MODE`	Create a `VersionedElement` valid in the period of applicability.

Methods inherited from `Document`:

Methods	Representational Mode	Current Mode	Sequenced Mode
`getDocType()`	Return the document type declaration associated with this document.		
`getDocumentElement()`	Call `getDocumentElement()` of `Document`	Support time-varying root. Return `Element`.	Support time-varying root. Return `VersionedElement`.
`getElementById()`	Call `getDocumentById()` of `Document`	To be considered.	To be considered.
`getElementsByTagName()`	Call `getElementsByTagName()` of `Document`.	Return a `NodeList` that contains elements valid at the current instant.	Return a `NodeList` that contains elements valid within the period of applicability.
`getImplementation()`	Call `getImplementation()` of `Document`	Return `DOMImplementation` that supports creating a temporal XML document.	
`createElement()`	Call `createElement()` of `Document`	Create an `Element` valid from the current instant.	Create an `Element` (version) valid in the period of applicability.
`createAttribute()`	Call `createAttribute()` of `Document`	Create an `Attr` that is valid from the current instant.	Create a `TAttr` that is valid in the period of applicability.

Methods inherited from `Node`.

Methods	Representational Mode	Current Mode	Sequenced Mode
`getAttributes()`	Call `getAttributes()` of `Document`	Return `NULL`.	
`getChildNodes()`	Call `getChildNodes()` of `Document`	Return a `NodeList` that contains the children that are valid under the temporal constraint.	
`getFirstChild()`	Call `getFirstChild()` of `Document`	Return the first child.	
`getLastChild()`	Call `getLastChild()` of `Document`	If last child is a document element, return an `Element`, otherwise return what DOM returns.	If last child is a document element, return a `VersionedElement`, otherwise return what DOM returns.
`hasChildNodes()`	Call `hasChildNodes()` of `Document`	If the only child node is document element and the root is time-varying, return `true` if the root is valid at the instant or in the extent, otherwise return `false`.	
`getPreviousSibling()`	Call `getPreviodSibling()` of `Document`	Return `NULL`.	
`getNextSibling()`	Call `getNextSibling()` of `Document`	Return `NULL`.	
`getNodeName()`	Call `getNodeName()` of `Document`	Return "`#tdocument`".	
`getNodeType()`	Call `getNodeType()` of `Document`	Return `TDOCUMENT_NODE`.	
`getNodeValue()`	Call `getNodeValue()` of `Document`	Return `NULL`.	
`getOwnerDocument()`	Call `getOwnerDocument()` of `Document`	Return `NULL`.	
`getParentNode()`	Call `getParentNode()` of `Document`	Return `NULL`.	
`hasAttributes()`	Call `hasAttributes()` of `Document`	Return `false`.	
`isSupported`	Support the "temporal" feature in addition to the features supported by underlying DOM implementation.		

A.2 Extensions to the `Element` Interface

New methods of `VersionedElement`.

Methods	Sequenced Mode
`int getNumVersion()`	Return the number of versions in this `VersionedElement`.
`Element getVersion (int idx)`	Return specified version. The range of index is from 0 to the number of versions minus 1.
`Period getValidTimeStamp()`	Return `PeriodSet` which contains merged timestamps of the representational elements.
`Node getPreviousSiblingOfParentVersion (int idx)`	Return the previous sibling in the parent's specified version.
`Node getNextSiblingOfParentVersion (int idx)`	Return the next sibling in the parent's specified version.
`void addVersion(Element version, Period ts)`	Add a version to this versioned element in the specified period.
`boolean hasSeqAttribute(String name)`	Test if the element has an attribute with the given name at sometime within the period of applicability.

Methods from `Element`.

Methods	Current Mode	Sequenced Mode
`getAttribute()`	Return the value of specified attribute.	Return `NULL` because an attribute may have multiple values in different periods. Call `getAttributeNode()` instead to access attribute.
`getAttributeNode()`	Return `Attr`	Return `TAttr`.
`getElementsByTagName()`	Return the descendants with the given name.	
`getTagName()`	Return the element's name.	
`hasAttribute()`	Test if the element has an attribute with the given name at the current instant.	Test if the element has an attribute with the given name during the entire period of the intersection of the period of applicability and the lifespan of the element.
`removeAttribute()`	Remove the attribute in the contiguous period beginning from the current instant.	Remove the attribute during the period of applicability.
`removeAttributeNode()`	Remove the attribute in the contiguous period beginning from the current instant.	Remove the attribute during the period of applicability.
`setAttribute()`	Add an attribute valid from the current instant.	Add an attribute valid within the period of applicability.
`setAttributeNode()`	Add an attribute valid from the current instant.	Add an attribute valid within the period of applicability.

Methods from `Node`.

Functions	Current Mode	Sequenced Mode	
	`Element`	`VerionedElement`	`Element`
`getAttributes()`	Return `NamedNodeMap`.		
`getChildNodes()`	Return the child nodes valid at the instant.	Return `NULL` because different versions have different ordering of children.	Return the child nodes valid in the period.
`getFirstChild()`	Return first child valid at the instant.	Return `NULL`.	Return first child valid in the period.
`getLastChild()`	Return last child valid at the instant.	Return `NULL`.	Return last child valid in the period.
`getPreviousSibling()`	Return previous sibling valid at the instant.	Return `NULL`. Call `getPreviousSiblingInParentVersion()` instead.	Return `NULL` because of multiple versions..
`getNextSibling()`	Return next sibling valid at the instant.	Return `NULL`. Call `getNextSiblingInParentVersion()` instead.	Return `NULL` because of multiple versions..
`getNodeName()`	Return element's name.		
`getNodeType()`	Return `ELEMENT_NODE`.		
`getNodeValue()`	Return `NULL`		
`getOwnerDocument()`	Return a `TDocument`		
`getParentNode()`	Return parent `Element`.	Return parent `VersionedElement`.	
`hasAttributes()`	Return true if the element has attributes (including time-varying attribute), otherwise return false.		
`isSupported()`	Support the "temporal" feature in addition to the features supported by underlying DOM implementation.		
`hasChildNodes()`	Return `true` if it has child nodes valid at the instant or in the period, otherwise return false.		
`appendChild()`	Use `Element`'s method.	Append a child that is valid from the current instant.	Append a child that is valid during the period of applicability.
`insertBefore()`	Use `Element`'s method.	insert a new child that is valid from the current instant before the given child.	insert a new child that is valid during the period of applicability before the given child.
`removeChild()`	Use `Element`'s method.	Remove the specified child beginning from the current instant.	Remove the specified child during the period of applicability.
`replaceChild()`	Use `Element`'s method.	Replace the specified child with a new child beginning from the current instant.	Remove the specified child with a new child during the period of applicability.

A.3 Extensions to the `Attr` Interface

Method specific to `TAttr`.

Methods	Representational Mode	Current Mode	Sequenced Mode
`getSequencedValue()`	Not applicable	Not applicable	Return `SequencedValue`.

Methods from `Attr`.

Methods	Representational Mode	Current Mode	Sequenced Mode
Methods of `Attr`			
`getName()`	Return attribute's name.		
`getOwnerElement()`	Return owner `Element`.		Return owner `VersionedElement`.
`getSpecified()`	Return `true`.		
`getValue()`	Use `Attr`'s method.	Return the valid valid at the current instant.	Return `NULL` because multiple values may exist in different periods. Call `getSequencedValue()` instead.
`setValue()`	Use `Attr`'s method.	Set the value in a contagious period beginning from the current instant.	Set the value in the period of applicability.

A.4 Other Extensions

These classes and interfaces provide temporal information. They are immutable.

A.4.1 The `SequencedValue` Interface

Methods	Semantics
`int getLength()`	Return the number of values contained.
`SequencedValue getValue(int idx)`	Return the specified value given its index.
`PeriodSet getTimeStamp(int idx)`	Return the timestamp of a specified value given its index.

A.4.2 The `PeriodSet` Interface

Methods	Semantics
`int getLength()`	Return the number of `Period` contained.
`Period getPeriod(int idx)`	Return the specified `Period` given its index.

A.4.3 The `Period` Class

Methods	Semantics
`Date getBegin()`	Return the beginning instant, which is contained in the period.
`Date getEnd()`	Return the ending instant, which is excluded from the period.

B Temporal XML Document Example

B.1 Example of Reading

Temporal XML document used as the input to the example programs.

```
<?xml version="1.0" standalone="yes"?>
   <People>
       <Person>
           <Time-Stamp vtbegin="1999-01-01" vtend="2001-12-31"/>
           <Time-VaryingAttr name="Title" value="Director"
               vtbegin="1999-01-01" vtend="2001-12-31"/>
           <Name>Shery</Name>
           <DateofBirth>1969-01-10</DateofBirth>
       </Person>
       <Person>
           <Time-Stamp vtbegin="1992-01-01" vtend="1994-12-31" id="17"/>
           <Time-VaryingAttr name="Title" value="Lecture"
               vtbegin="1994-01-01" vtend="1994-12-31"/>
           <Name>Bob</Name>
           <DateofBirth>1945-05-09</DateofBirth>
           <YearlySalary>
               <Time-Stamp vtbegin="1994-01-01" vtend="1994-12-31"/>
               30000
           </YearlySalary>
       </Person>
       <Person>
           <Time-Stamp vtbegin="1995-01-01" vtend="2002-12-31" id="17"/>
           <Time-VaryingAttr name="Title" value="Assistant Provost"
               vtbegin="1995-01-01" vtend="1996-12-31"/>
           <Time-VaryingAttr name="Title" value="Provost"
               vtbegin="1997-01-01" vtend="1998-12-31"/>
           <Time-VaryingAttr name="Title" value="Professor"
               vtbegin="1999-01-01" vtend="2002-12-31"/>
           <Name>Bob</Name>
           <DateofBirth>1945-05-09</DateofBirth>
           <YearlySalary>
               <Time-Stamp vtbegin="1995-01-01" vtend="1996-12-31"/>
               60000
           </YearlySalary>
           <YearlySalary>
               <Time-Stamp vtbegin="1997-01-01" vtend="1998-12-31"/>
               70000
           </YearlySalary>
           <YearlySalary>
               <Time-Stamp vtbegin="1999-01-01" vtend="2002-12-31"/>
               80000
           </YearlySalary>
       </Person>
   </People>
```

B.1.1 Reading Program in Representational Mode

This program outputs the content of the temporal XML document in representational mode.

```
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.IOException;
import java.io.*;
import org.xml.sax.*;

public class RepDemo {
    public static void main(String[] args) throws Exception{
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder parser = factory.newDocumentBuilder();
        TDocument doc = new TDocumentImpl(
                    parser.parse(new InputSource("people.xml")));

        // Representational Mode
        System.out.println("Representational Mode:");
        System.out.println();

        NodeList personList = doc.getElementsByTagName("Person");

        for (int i = 0; i < personList.getLength(); i++)
            output_element(personList.item(i), "");
    }

    static void output_attrs(Element element, String prefix){
        NamedNodeMap attrs = element.getAttributes();

        if (attrs.getLength() > 0)
            System.out.println(prefix + "ATTRIBUTES:");
        else
            return;

        String new_prefix = prefix + "   ";

        for (int i = 0; i < attrs.getLength(); i++){
            Attr attr = (Attr)attrs.item(i);
            System.out.println(new_prefix + attr.getName() + " = " +
                               attr.getValue());
        }
    }

    static void output_element(Node node, String prefix){
        String new_prefix = prefix + "   ";

        if (node instanceof Text){
            System.out.println(prefix + node.getNodeValue());
            return;
        }

        Element element = (Element)node;
        System.out.println(prefix + element.getNodeName() + ":");

        output_attrs(element, new_prefix);

        NodeList children = element.getChildNodes();
```

```
        if (children.getLength() > 0){
            System.out.println(new_prefix + "CHILDREN:");
            for (int i = 0; i < children.getLength(); i++){
                Node child = children.item(i);
                output_element(child, new_prefix + "   ");
            }
        }
    }
}
```

B.1.2 Output of Representational Mode

Output of the program in Appendix C given Appendix A as input.

```
Representational Mode:

Person:
   CHILDREN:
      Time-Stamp:
         ATTRIBUTES:
            vtbegin = 1999-01-01
            vtend = 2001-12-31
      Time-VaryingAttr:
         ATTRIBUTES:
            name = Title
            value = Director
            vtbegin = 1999-01-01
            vtend = 2001-12-31
      Name:
         CHILDREN:
            Shery
      DateofBirth:
         CHILDREN:
            1969-01-10
Person:
   CHILDREN:
      Time-Stamp:
         ATTRIBUTES:
            id = 17
            vtbegin = 1992-01-01
            vtend = 1994-12-31
      Time-VaryingAttr:
         ATTRIBUTES:
            name = Title
            value = Lecture
            vtbegin = 1994-01-01
            vtend = 1994-12-31
      Name:
         CHILDREN:
            Bob
      DateofBirth:
         CHILDREN:
            1945-05-09
      YearlySalary:
```

```
         CHILDREN:
            Time-Stamp:
               ATTRIBUTES:
                  vtbegin = 1994-01-01
                  vtend = 1994-12-31
            Amount:
               CHILDREN:
                  30000
Person:
   CHILDREN:
      Time-Stamp:
         ATTRIBUTES:
            id = 17
            vtbegin = 1995-01-01
            vtend = 2002-12-31
      Time-VaryingAttr:
         ATTRIBUTES:
            name = Title
            value = Assistant Provost
            vtbegin = 1995-01-01
            vtend = 1996-12-31
      Time-VaryingAttr:
         ATTRIBUTES:
            name = Title
            value = Provost
            vtbegin = 1997-01-01
            vtend = 1998-12-31
      Time-VaryingAttr:
         ATTRIBUTES:
            name = Title
            value = Professor
            vtbegin = 1999-01-01
            vtend = 2002-12-31
      Name:
         CHILDREN:
            Bob
      DateofBirth:
         CHILDREN:
            1945-05-09
      YearlySalary:
         CHILDREN:
            Time-Stamp:
               ATTRIBUTES:
                  vtbegin = 1995-01-01
                  vtend = 1996-12-31
            Amount:
               CHILDREN:
                  60000
      YearlySalary:
         CHILDREN:
            Time-Stamp:
               ATTRIBUTES:
                  vtbegin = 1997-01-01
                  vtend = 1998-12-31
```

```
            Amount:
               CHILDREN:
                  70000
      YearlySalary:
         CHILDREN:
            Time-Stamp:
               ATTRIBUTES:
                  vtbegin = 1999-01-01
                  vtend = 2002-12-31
            Amount:
               CHILDREN:
                  80000
```

B.1.3 Reading Program in Current Mode

This program outputs the content of the temporal XML document in current mode.

```
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.*;
import java.text.*;
import java.util.*;

public class CurDemo {
    static SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    public static void main(String[] args) throws Exception{
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder parser = factory.newDocumentBuilder();
        TDocument doc = new TDocumentImpl(parser.parse("people.xml"));

    //Current Mode
    doc.setModeCurrent("yyyy-MM-dd", "2000-08-27");
    System.out.println("Current Mode: " + sdf.format(doc.getCurrentInstant()));
    System.out.println();

        NodeList personList = doc.getElementsByTagName("Person");

        for (int i = 0; i < personList.getLength(); i++)
            output_element(personList.item(i), "");
    }

    static void output_attrs(Element element, String prefix){
        NamedNodeMap attrs = element.getAttributes();

        if (attrs.getLength() > 0)
            System.out.println(prefix + "ATTRIBUTES:");
        else
            return;

        String new_prefix = prefix + "   ";

        for (int i = 0; i < attrs.getLength(); i++){
```

```
            Attr attr = (Attr)attrs.item(i);
            System.out.println(new_prefix + attr.getName() + " = "
                               + attr.getValue());
        }
    }

    static void output_element(Node node, String prefix){
        String new_prefix = prefix + "   ";

        if (!(node instanceof Element)){
            System.out.println(prefix + node.getNodeValue());
            return;
        }

        Element element = (Element)node;
        System.out.println(prefix + element.getNodeName() + ":");

        output_attrs(element, new_prefix);

        NodeList children = element.getChildNodes();
        if (children.getLength() > 0){
            System.out.println(new_prefix + "CHILDREN:");
            for (int i = 0; i < children.getLength(); i++){
                Node child = children.item(i);
                output_element(child, new_prefix + "   ");
            }
        }
    }
}
```

B.1.4 Output of Current Mode

In current mode, τDOM outputs data that is valid at the current instant in its appropriate form.

```
Current Mode: 2000-08-27

Person:
   ATTRIBUTES:
      Title = Director
   CHILDREN:
      Name:
         CHILDREN:
            Shery
      DateofBirth:
         CHILDREN:
            1969-01-10
Person:
   ATTRIBUTES:
      Title = Professor
   CHILDREN:
      Name:
         CHILDREN:
            Bob
      DateofBirth:
```

```
        CHILDREN:
           1945-05-09
      YearlySalary:
         CHILDREN:
           80000
```

B.1.5 Program of Reading in Sequenced Mode

This program outputs the content of the temporal XML document in sequenced mode.

```
import javax.xml.parsers.*;
import org.w3c.dom.*;
import java.io.*;
import java.text.*;
import java.util.*;

public class SeqDemo {
    static SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

    public static void main(String[] args) throws Exception{
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
        DocumentBuilder parser = factory.newDocumentBuilder();
        TDocument doc = new TDocumentImpl(parser.parse(new File("people.xml")));

        doc.setModeSequenced("yyyy-MM-dd", "1992-01-01", "1997-12-31");
        System.out.println("Sequenced Mode: " +
                           sdf.format(doc.getSequencedExtent().getValidBegin()) +
                          " --> " +
                          sdf.format(doc.getSequencedExtent().getValidEnd()));
        System.out.println();

        NodeList personList = doc.getElementsByTagName("Person");

        for (int i = 0; i < personList.getLength(); i++)
           output_element(personList.item(i), "");
    }

    static void output_element(Node node, String prefix){
        String new_prefix = prefix + "   ";

        if (node.getNodeType() == Node.TEXT_NODE){
            System.out.println(prefix + node.getNodeValue());
            return;
        }

        VersionedElement element = (VersionedElement)node;
        System.out.println(prefix + element.getNodeName() + ": ");

        PeriodSet ps = element.getValidTimeStamp();
        System.out.println(new_prefix + "Time-stamp: ");
        for (int i = 0; i < ps.getLength(); i++){
            Period p = ps.getPeriod(i);

            System.out.println(new_prefix + "   " +
```

```
                            sdf.format(p.getValidBegin()) + " --> " +
                            sdf.format(p.getValidEnd()));
        }

        output_attrs(element, new_prefix);

        NodeList children = element.getVersion(0).getChildNodes();
        if (children.getLength() > 0){
            System.out.println(new_prefix + "CHILDREN:");
            for (int i = 0; i < children.getLength(); i++){
                Node child = children.item(i);
                output_element(child, new_prefix + "   ");
            }
        }
    }

    static void output_attrs(Element element, String prefix){
        String new_prefix = prefix + "   ";

        Attr attr = element.getAttributeNode("Title");
        if (attr == null)
            return;

        System.out.println(prefix + "ATTRIBUTES:");

        if (attr instanceof TAttr){
            TAttr tattr = (TAttr)attr;
            SequencedValue seq_value = tattr.getSequencedValue();
            System.out.println(new_prefix + tattr.getName());

            for (int j = 0; j < seq_value.getLength(); j++){
                Period p = seq_value.getTimeStamp(j);
                System.out.println(new_prefix + "   " +
                           seq_value.getValue(j) + ": " +
                           sdf.format(seq_value.getTimeStamp(j).getValidBegin()) +
                           " --> " +
                           sdf.format(seq_value.getTimeStamp(j).getValidEnd()));
            }
        }
        else{
            System.out.println(new_prefix + attr.getName() +
                            " = " + attr.getValue());
        }
    }
}
```

B.1.6 Output of Sequenced Mode

In sequenced mode, τDOM outputs data that is valid within the period of applicability in its appropriate form.

```
Sequenced Mode: 1992-01-01 --> 1997-12-31

Person:
   Time-stamp:
```

```
      1992-01-01 --> 1997-12-31
   ATTRIBUTES:
      Title
         Lecture: 1994-01-01 --> 1994-12-31
         Assistant Provost: 1995-01-01 --> 1996-12-31
         Provost: 1997-01-01 --> 1997-12-31
   CHILDREN:
      Name:
         Time-stamp:
            1992-01-01 --> 1997-12-31
         CHILDREN:
            Bob
      DateofBirth:
         Time-stamp:
            1992-01-01 --> 1997-12-31
         CHILDREN:
            1945-05-09
      YearlySalary:
         Time-stamp:
            1994-01-01 --> 1997-12-31
         CHILDREN:
            30000
            60000
            70000
```

B.2 Examples of Modification

Here we demonstrate how to modify a temporal XML document. Assume we have a snapshot. We modify it twice in current mode and once in sequenced mode. The snapshot is as below. The data in the document comes from XBench [9].

```
<?xml version="1.0"?>
<catalog cur_date="2002-01-01" cur_verid="232">
    <item id="I1959" subject="LITERATURE" verid="0">
        <title>Unable,newusersBABABABAALNGOGmake.Annual,particular</title>
        <authors>
            <author>
                <name>
                    <first_name>Alla</first_name>
                    <middle_name>fu</middle_name>
                    <last_name>BABABABABAOGINXEbH</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="207">
                        <street_information name_of_country="Chile">
                            <street_address verid="10">Cs:Q}:_}Z=zI++@}J/dUJ(].lU[P
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="145">+71(81)96767976</phone_number>
                    <email_address>BABABABABAOGINXEbH@uni-sb.de</email_address>
                </contact_information>
            </author>
```

```
            <author>
                <name>
                    <first_name>Jianan</first_name>
                    <middle_name>sentiments would d</middle_name>
                    <last_name>BABABABABAULRI</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="208">
                        <street_information name_of_country="Venezuela">
                            <street_address verid="11">S5!U+uhLGI;g{y$4
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="146">+41(149)33135070</phone_number>
                    <email_address>BABABABABAULRI@concordia.ca</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Anind</first_name>
                    <middle_name>careful a</middle_name>
                    <last_name>BABABABAALBAIN</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="209">
                        <street_information name_of_country="Croatia">
                            <street_address verid="12">OnFn+M,GjimXh#R?ujJ2|}
                            </street_address>
                            <street_address verid="13">k7q;4?*2t8;3$d9,#mAfcydE@,|e,
                                ^6ER!oU6p</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="147">+6(889)98249888</phone_number>
                    <email_address>BABABABAALBAIN@sun.com</email_address>
                </contact_information>
            </author>
        </authors>
        <related_items>
            <related_item verid="170">
                <item_id>I886</item_id>
            </related_item>
            <related_item verid="171">
                <item_id>I172</item_id>
            </related_item>
        </related_items>
        <pricing quantity_in_stock="25">
            <cost currency="Dollars">2836.94</cost>
            <when_is_available>1968-02-18</when_is_available>
        </pricing>
    </item>
    <item id="I886" subject="BUSINESS" verid="1">
        <title>Social,possibleBABABABAALATULchildrenfor</title>
        <authors>
            <author>
```

```
            <name>
                <first_name>Madhura</first_name>
                <middle_name>multip</middle_name>
                <last_name>BABABABAOGSEBA</last_name>
            </name>
            <contact_information>
                <mailing_address verid="210">
                    <street_information name_of_country="Eastern Caribbean">
                        <street_address verid="14">]o=,4sXBˆIe6x4xKkTRpH6Htˆ
                        Ikw0p*PmH3M7 n-</street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="148">+69(444)48546076</phone_number>
                <email_address>BABABABAOGSEBA@inria.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Debbie</first_name>
                <middle_name>permane</middle_name>
                <last_name>BABABABAOGATAL</last_name>
            </name>
            <contact_information>
                <mailing_address verid="211">
                    <street_information name_of_country="Armenia">
                        <street_address verid="15">zc/RyT+zt42_d?ˆˆqX/P]9B
                        </street_address>
                        <street_address verid="16">].b/RrRr(k.S(GsomgMp]L
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="149">+87(781)19242914</phone_number>
                <email_address>BABABABAOGATAL@ubs.com</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items>
        <related_item verid="176">
            <item_id>I172</item_id>
        </related_item>
    </related_items>
    <pricing quantity_in_stock="16">
        <cost currency="Dollars">4944.09</cost>
        <when_is_available>1982-08-28</when_is_available>
    </pricing>
</item>
<item id="I172" subject="COOKING" verid="2">
    <title>BA</title>
    <authors>
        <author>
            <name>
                <first_name>Gigliola</first_name>
                <middle_name>furious, </middle_name>
                <last_name>BABABABABAOGIN</last_name>
```

```
            </name>
            <contact_information>
                <mailing_address verid="212">
                    <street_information name_of_country="Ireland">
                        <street_address verid="17">3?61)gphl 8H1UEY,t80hNA
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="150">+73(450)67471136</phone_number>
                <email_address>BABABABABAOGIN@ogi.edu</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Akeo</first_name>
                <middle_name>st</middle_name>
                <last_name>BABABABAOGULBAXWSl4</last_name>
            </name>
            <contact_information>
                <mailing_address verid="213">
                    <street_information name_of_country="Poland">
                        <street_address verid="18">Zx8pvbWT5e1dP4^xbv@^6
                        G~0uzl 23}A:7Ih/</street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="151">+45(307)102286.5</phone_number>
                <email_address>BABABABAOGULBAXWSl4@imag.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Kyung</first_name>
                <middle_name>rea</middle_name>
                <last_name>BABABABABAOGOG</last_name>
            </name>
            <contact_information>
                <mailing_address verid="214">
                    <street_information name_of_country="Bangla Desh">
                        <street_address verid="19">XecJy!COgBwf4#I7p
                        </street_address>
                        <street_address verid="20">]3^DiSjZMQK}hkpxlwB9f|rJj?M+2,N@
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="152">+10(783)41845260</phone_number>
                <email_address>BABABABABAOGOG@monmouth.edu</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items/>
    <pricing quantity_in_stock="11">
        <cost currency="Dollars">2704.13</cost>
        <when_is_available>1963-11-04</when_is_available>
    </pricing>
```

```
    </item>
</catalog>
```

B.2.1 Program of First Modification in Current Mode

Here we test the following modification functions of τDOM in current mode.

1. Modify the content of an element by changing one author's address and another author's phone number.
2. Append a child by adding a related item.
3. Add an attribute to the address of an author.

```
/*
 * The TDOM API Sample, Version 1.0
 *
 * Copyright (c) 1999-2002
 * The Computer Science Department of University of Arizona.
 * All rights reserved.
 *
 * TDOM APIs extend DOM API to support operations on temporal XML document.
 * For more information about TDOM API and temporal XML document,
 * visit the homepage of TAU Project <http://www.cs.arizona.edu/tau/>.
 */

import org.w3c.dom.*;
import org.apache.xerces.parsers.*;
import javax.xml.parsers.*;
import org.w3c.dom.ls.*;

import java.util.*;
import java.text.*;
import java.io.*;

/**
 *  Evolve1to2 reads the first snapshot document, set the current instant
 *  to a time later, makes some changes and writes out the document.
 */
public class Evolve1to2
{
    static TDocumentImpl tdoc;

    public static void main(String[] args) throws Exception{
        /**
         *  parse the snapshot document
         */
        DOMParser parser = new DOMParser();
        parser.parse("catalog1.xml");

        Document snap_doc = parser.getDocument();

        tdoc = new TDocumentImpl(snap_doc);

        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
```

```
        /**
         *  set the current instant
         */
        tdoc.setModeCurrent("yyyy-MM-dd", "2002-02-01");

        /**
         *  change the addresss of the first author of the first item
         */
        change_address();

        /**
         *  change the addresss of the second author of the second item
         */
        change_phone_number();

        /**
         *  add a related item to the second item
         */
        add_related_item();

        /**
         *  add confirmed attribute to the second author of the third item
         */
        confirm_address();

        tdoc.writeTree();
    }

    static void change_address(){
        NodeList item_nl = tdoc.getElementsByTagName("item");

        Element item0 = (Element)item_nl.item(0);

        NodeList children = item0.getChildNodes();

        Element authors = null;
        for (int i = 0; i < children.getLength(); i++)
            if (children.item(i).getNodeName().equals("authors")){
                authors = (Element)children.item(i);
                break;
            }

        children = authors.getChildNodes();
        Element author0 = (Element)children.item(0);

        Element street = (Element)author0.getElementsByTagName
                                        ("street_address").item(0);
        Text t = (Text)street.getFirstChild();

        t.setData("1234 1st ave, New York");
    }

    static void change_phone_number(){
```

```
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item1 = (Element)item_nl.item(1);

    NodeList children = item1.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    children = authors.getChildNodes();
    Element author1 = (Element)children.item(1);

    Element phone = (Element)author1.getElementsByTagName
                                ("phone_number").item(0);
    Text t = (Text)phone.getFirstChild();

    t.setData("+56(344)19242914");
}

static void add_related_item(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item1 = (Element)item_nl.item(1);

    NodeList children = item1.getChildNodes();

    Element related_items = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("related_items")){
            related_items = (Element)children.item(i);
            break;
        }

    Element new_related_item = tdoc.createElement("related_item");
    Element item_id = tdoc.createElement("item_id");

    item_id.appendChild(tdoc.createTextNode("I1959"));
    new_related_item.appendChild(item_id);

    related_items.appendChild(new_related_item);
}

static void confirm_address(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(2);

    NodeList children = item2.getChildNodes();

    Element authors = null;
```

```
        for (int i = 0; i < children.getLength(); i++)
            if (children.item(i).getNodeName().equals("authors")){
                authors = (Element)children.item(i);
                break;
            }

        children = authors.getChildNodes();
        Element author1 = (Element)children.item(1);

        Element mailing_address = (Element)author1.getElementsByTagName
                                        ("mailing_address").item(0);

        mailing_address.setAttribute("confirmed", "yes");
    }
}
```

B.2.2 The Document after First Modification

The changes are enclosed in boxes.

```
<?xml version="1.0"?>
<catalog cur_date="2002-01-01" cur_verid="234">
    <item id="I1959" subject="LITERATURE" verid="0">
        <title>Unable,newusersBABABABAALNGOGmake.Annual,particular</title>
        <authors>
            <author>
                <name>
                    <first_name>Alla</first_name>
                    <middle_name>fu</middle_name>
                    <last_name>BABABABABAOGINXEbH</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="207">
                        <street_information name_of_country="Chile">
```

```
<street_address verid="10">
    <timestamp vtbegin="2002-01-01" vtend="2002-02-01"/>
        Cs:Q}:_}Z=zI++@}J/dUJ(].lU[P
    </street_address>
<street_address verid="10">
    <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
    1234 1st ave, New York
    </street_address>
```

```
                        </street_information>
                    </mailing_address>
                    <phone_number verid="145">+71(81)96767976</phone_number>
                    <email_address>BABABABABAOGINXEbH@uni-sb.de</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Jianan</first_name>
                    <middle_name>sentiments would d</middle_name>
                    <last_name>BABABABABAULRI</last_name>
```

```
            </name>
            <contact_information>
                <mailing_address verid="208">
                    <street_information name_of_country="Venezuela">
                        <street_address verid="11">S5!U+uhLGI;g{y$4
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="146">+41(149)33135070</phone_number>
                <email_address>BABABABABAULRI@concordia.ca</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Anind</first_name>
                <middle_name>careful a</middle_name>
                <last_name>BABABABAALBAIN</last_name>
            </name>
            <contact_information>
                <mailing_address verid="209">
                    <street_information name_of_country="Croatia">
                        <street_address verid="12">OnFn+M,GjimXh#R?ujJ2|}
                        </street_address>
                        <street_address verid="13">
                        k7q;4?*2t8;3$d9,#mAfcydE@,|e,^6ER!oU6p
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="147">+6(889)98249888</phone_number>
                <email_address>BABABABAALBAIN@sun.com</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items>
        <related_item verid="170">
            <item_id>I886</item_id>
        </related_item>
        <related_item verid="171">
            <item_id>I172</item_id>
        </related_item>
    </related_items>
    <pricing quantity_in_stock="25">
        <cost currency="Dollars">2836.94</cost>
        <when_is_available>1968-02-18</when_is_available>
    </pricing>
</item>
<item id="I886" subject="BUSINESS" verid="1">
    <title>Social,possibleBABABABAALATULchildrenfor</title>
    <authors>
        <author>
            <name>
                <first_name>Madhura</first_name>
                <middle_name>multip</middle_name>
                <last_name>BABABABAOGSEBA</last_name>
```

```
            </name>
            <contact_information>
                <mailing_address verid="210">
                    <street_information name_of_country="Eastern Caribbean">
                        <street_address verid="14">
                        ]o=,4sXBˆIe6x4xKkTRpH6Ht˜Ikw0p*PmH3M7 n-
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="148">+69(444)48546076</phone_number>
                <email_address>BABABABAOGSEBA@inria.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Debbie</first_name>
                <middle_name>permane</middle_name>
                <last_name>BABABABAOGATAL</last_name>
            </name>
            <contact_information>
                <mailing_address verid="211">
                    <street_information name_of_country="Armenia">
                        <street_address verid="15">zc/RyT+zt42_d?ˆˆqX/P]9B
                        </street_address>
                        <street_address verid="16">].b/RrRr(k.S(GsomgMp]L
                        </street_address>
                    </street_information>
                </mailing_address>
```

```
<phone_number verid="149">
    <timestamp vtbegin="2002-01-01" vtend="2002-02-01"/>
    +87(781)19242914</phone_number>
<phone_number verid="149">
    <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
    +56(344)19242914</phone_number>
<email_address>BABABABAOGATAL@ubs.com</email_address>
```

```
            </contact_information>
        </author>
    </authors>
    <related_items>
        <related_item verid="176">
            <item_id>I172</item_id>
        </related_item>
```

```
<related_item verid="232">
    <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
    <item_id verid="233">I1959</item_id>
</related_item>
```

```
    </related_items>
    <pricing quantity_in_stock="16">
        <cost currency="Dollars">4944.09</cost>
        <when_is_available>1982-08-28</when_is_available>
    </pricing>
```

```
</item>
<item id="I172" subject="COOKING" verid="2">
    <title>BA</title>
    <authors>
        <author>
            <name>
                <first_name>Gigliola</first_name>
                <middle_name>furious, </middle_name>
                <last_name>BABABABABAOGIN</last_name>
            </name>
            <contact_information>
                <mailing_address verid="212">
                    <street_information name_of_country="Ireland">
                        <street_address verid="17">3?61)gphl 8H1UEY,t80hNA
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="150">+73(450)67471136</phone_number>
                <email_address>BABABABABAOGIN@ogi.edu</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Akeo</first_name>
                <middle_name>st</middle_name>
                <last_name>BABABABAOGULBAXWSl4</last_name>
            </name>
            <contact_information>
                <mailing_address verid="213">
```


```
<timevarying-attr name="confirmed" value="yes"
        vtbegin="2002-02-01" vtend="9999-12-31"/>
```


```
                    <street_information name_of_country="Poland">
                        <street_address verid="18">
                        Zx8pvbWT5e1dP4^xbv@^6G~0uzl 23}A:7Ih/
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="151">+45(307)102286.5</phone_number>
                <email_address>BABABABAOGULBAXWSl4@imag.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Kyung</first_name>
                <middle_name>rea</middle_name>
                <last_name>BABABABABAOGOG</last_name>
            </name>
            <contact_information>
                <mailing_address verid="214">
                    <street_information name_of_country="Bangla Desh">
                        <street_address verid="19">XecJy!COgBwf4#I7p
                        </street_address>
                        <street_address verid="20">
```

```
                        ]3^DiSjZMQK}hkpxlwB9f|rJj?M+2,N@
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="152">+10(783)41845260</phone_number>
                <email_address>BABABABABAOGOG@monmouth.edu</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items/>
    <pricing quantity_in_stock="11">
        <cost currency="Dollars">2704.13</cost>
        <when_is_available>1963-11-04</when_is_available>
    </pricing>
  </item>
</catalog>
```

B.2.3 Program of Second Modification in Current Mode

Here we test the following modification functions of τDOM in current mode.

1. Delete a child by removing a related item.
2. Change an attribute by changing the quantity in stock of one item and the country name of an author's address.
3. Add a child by adding a related item.
4. Delete an attribute from the an author's address.

```
/*
 * The TDOM API Sample, Version 1.0
 *
 * Copyright (c) 1999-2002
 * The Computer Science Department of University of Arizona.
 * All rights reserved.
 *
 * TDOM APIs extend DOM API to support operations on temporal XML document.
 * For more information about TDOM API and temporal XML document,
 * visit the homepage of TAU Project <http://www.cs.arizona.edu/tau/>.
 */

import org.w3c.dom.*;
import org.apache.xerces.parsers.*;
import javax.xml.parsers.*;
import org.w3c.dom.ls.*;

import java.util.*;
import java.text.*;
import java.io.*;

/**
 *  Evolve2to3 reads the output document of Evolve1to2, set the current instant
 *  to a time later, makes some changes and writes out the document.
 */
```

```
public class Evolve2to3
{
    static TDocumentImpl tdoc;

    public static void main(String[] args) throws Exception{
        /*
         *  parse the temporal xml document generated by Evolve1to2
         */
        DOMParser parser = new DOMParser();
        parser.parse("catalog.xml");

        Document snap_doc = parser.getDocument();

        tdoc = new TDocumentImpl(snap_doc);

        /*
         *  set the current instant
         */
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

        tdoc.setModeCurrent("yyyy-MM-dd", "2002-03-01");

        /**
         *  remove the first related item of the first item
         */
        remove_related_item();

        /**
         *  change the quantity_in_stock attribute of the second item
         */
        change_quantity();

        /**
         *  change the name_of_country attribute
         *  of the first author of the third item
         */
        change_name_of_country();

        /**
         *  add a related item to the third item
         */
        add_related_item();

        /**
         *  remove the confirmed attribute of the second
         *  author of the third item
         */
        delete_confirmed();

        tdoc.writeTree();
    }

    static void remove_related_item(){
        NodeList item_nl = tdoc.getElementsByTagName("item");
```

```
    Element item0 = (Element)item_nl.item(0);

    NodeList children = item0.getChildNodes();

    Element related_items = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("related_items")){
            related_items = (Element)children.item(i);
            break;
        }

    Element rel_item = (Element)related_items.getElementsByTagName
                               ("related_item").item(0);

    related_items.removeChild(rel_item);
}

static void change_quantity(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item1 = (Element)item_nl.item(1);

    NodeList children = item1.getChildNodes();

    Element pricing = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("pricing")){
            pricing = (Element)children.item(i);
            break;
        }

    pricing.setAttribute("quantity_in_stock", "50");
}

static void change_name_of_country(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(2);

    NodeList children = item2.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    Element author0 = (Element)authors.getChildNodes().item(0);

    Element street_info = (Element)author0.getElementsByTagName
                               ("street_information").item(0);
    street_info.setAttribute("name_of_country", "England");
```

```
    }

    static void add_related_item(){
        NodeList item_nl = tdoc.getElementsByTagName("item");

        Element item2 = (Element)item_nl.item(2);

        NodeList children = item2.getChildNodes();

        Element related_items = null;
        for (int i = 0; i < children.getLength(); i++)
            if (children.item(i).getNodeName().equals("related_items")){
                related_items = (Element)children.item(i);
                break;
            }

        Element new_related_item = tdoc.createElement("related_item");
        Element item_id = tdoc.createElement("item_id");

        item_id.appendChild(tdoc.createTextNode("I1959"));
        new_related_item.appendChild(item_id);

        related_items.appendChild(new_related_item);
    }

    static void delete_confirmed(){
        NodeList item_nl = tdoc.getElementsByTagName("item");

        Element item2 = (Element)item_nl.item(2);

        NodeList children = item2.getChildNodes();

        Element authors = null;
        for (int i = 0; i < children.getLength(); i++)
            if (children.item(i).getNodeName().equals("authors")){
                authors = (Element)children.item(i);
                break;
            }

        children = authors.getChildNodes();
        Element author1 = (Element)children.item(1);

        Element mailing_address = (Element)author1.getElementsByTagName
                                        ("mailing_address").item(0);

        mailing_address.removeAttribute("confirmed");
    }
}
```

B.2.4 The Document after Second Modification in Current Mode

The changes made in this step are enclosed in boxes.

```
<?xml version="1.0"?>
```

```
<catalog cur_date="2002-01-01" cur\_verid="236">
    <item id="I1959" subject="LITERATURE" verid="0">
        <title>Unable,newusersBABABABAALNGOGmake.Annual,particular</title>
        <authors>
            <author>
                <name>
                    <first_name>Alla</first_name>
                    <middle_name>fu</middle_name>
                    <last_name>BABABABABAOGINXEbH</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="207">
                        <street_information name_of_country="Chile">
                            <street_address verid="10">
                                <timestamp vtbegin="2002-01-01"
                                        vtend="2002-02-01"/>Cs:Q}:_}Z=zI++@}J/dUJ(].lU[P
                                </street_address>
                            <street_address verid="10">
                                <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
                                1234 1st ave, New York</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="145">+71(81)96767976</phone_number>
                    <email_address>BABABABABAOGINXEbH@uni-sb.de</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Jianan</first_name>
                    <middle_name>sentiments would d</middle_name>
                    <last_name>BABABABABAULRI</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="208">
                        <street_information name_of_country="Venezuela">
                            <street_address verid="11">S5!U+uhLGI;g{y$4
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="146">+41(149)33135070</phone_number>
                    <email_address>BABABABABAULRI@concordia.ca</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Anind</first_name>
                    <middle_name>careful a</middle_name>
                    <last_name>BABABABAALBAIN</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="209">
                        <street_information name_of_country="Croatia">
                            <street_address verid="12">OnFn+M,GjimXh#R?ujJ2|}
                            </street_address>
```

```
                        <street_address verid="13">
                        k7q;4?*2t8;3$d9,#mAfcydE@,|e,^6ER!oU6p
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="147">+6(889)98249888</phone_number>
                <email_address>BABABABAALBAIN@sun.com</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items>
```

```
<related_item verid="170">
    <timestamp vtbegin="2002-01-01" vtend="2002-03-01"/>
    <item_id>I886</item_id>
</related_item>
```

```
        <related_item verid="171">
            <item_id>I172</item_id>
        </related_item>
    </related_items>
    <pricing quantity_in_stock="25">
        <cost currency="Dollars">2836.94</cost>
        <when_is_available>1968-02-18</when_is_available>
    </pricing>
</item>
<item id="I886" subject="BUSINESS" verid="1">
    <title>Social,possibleBABABABAALATULchildrenfor</title>
    <authors>
        <author>
            <name>
                <first_name>Madhura</first_name>
                <middle_name>multip</middle_name>
                <last_name>BABABABAOGSEBA</last_name>
            </name>
            <contact_information>
                <mailing_address verid="210">
                    <street_information name_of_country="Eastern Caribbean">
                        <street_address verid="14">
                        ]o=,4sXBˆIe6x4xKkTRpH6Ht˜Ikw0p*PmH3M7 n-
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="148">+69(444)48546076</phone_number>
                <email_address>BABABABAOGSEBA@inria.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Debbie</first_name>
                <middle_name>permane</middle_name>
                <last_name>BABABABAOGATAL</last_name>
            </name>
            <contact_information>
                <mailing_address verid="211">
```

```
                        <street_information name_of_country="Armenia">
                            <street_address verid="15">zc/RyT+zt42_d?^^qX/P]9B
                            </street_address>
                            <street_address verid="16">].b/RrRr(k.S(GsomgMp]L
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="149">
                        <timestamp vtbegin="2002-01-01"
                        vtend="2002-02-01"/>+87(781)19242914
                        </phone_number>
                    <phone_number verid="149">
                        <timestamp vtbegin="2002-02-01"
                        vtend="9999-12-31"/>+56(344)19242914
                        </phone_number>
                    <email_address>BABABABAOGATAL@ubs.com</email_address>
                </contact_information>
            </author>
        </authors>
        <related_items>
            <related_item verid="176">
                <item_id>I172</item_id>
            </related_item>
```

```
<related_item verid="232">
    <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
    <item_id verid="233">I1959</item_id>
</related_item>
```

```
        </related_items>
        <pricing>
```

```
<timevarying-attr name="quantity_in_stock" value="50"
    vtbegin="2002-03-01" vtend="9999-12-31"/>
<timevarying-attr name="quantity_in_stock" value="16"
    vtbegin="2002-01-01" vtend="2002-03-01"/>
```

```
            <cost currency="Dollars">4944.09</cost>
            <when_is_available>1982-08-28</when_is_available>
        </pricing>
    </item>
    <item id="I172" subject="COOKING" verid="2">
        <title>BA</title>
        <authors>
            <author>
                <name>
                    <first_name>Gigliola</first_name>
                    <middle_name>furious, </middle_name>
                    <last_name>BABABABABAOGIN</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="212">
                        <street_information>
                            <timevarying-attr name="name_of_country" value="England"
                                vtbegin="2002-03-01" vtend="9999-12-31"/>
```

```
                        <timevarying-attr name="name_of_country"
                            value="Ireland" vtbegin="2002-01-01" vtend="2002-03-01"/
                        <street_address verid="17">
                            3?61)gphl 8H1UEY,t80hNA
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="150">+73(450)67471136</phone_number>
                <email_address>BABABABABAOGIN@ogi.edu</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Akeo</first_name>
                <middle_name>st</middle_name>
                <last_name>BABABABAOGULBAXWSl4</last_name>
            </name>
            <contact_information>
                <mailing_address verid="213">

<timevarying-attr name="confirmed" value="yes"
        vtbegin="2002-02-01" vtend="2002-03-01"/>

                    <street_information name_of_country="Poland">
                        <street_address verid="18">
                        Zx8pvbWT5e1dP4^xbv@^6G~0uzl 23}A:7Ih/
                        </street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="151">+45(307)102286.5</phone_number>
                <email_address>BABABABAOGULBAXWSl4@imag.fr</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Kyung</first_name>
                <middle_name>rea</middle_name>
                <last_name>BABABABABAOGOG</last_name>
            </name>
            <contact_information>
                <mailing_address verid="214">
                    <street_information name_of_country="Bangla Desh">
                        <street_address verid="19">
                        XecJy!COgBwf4#I7p</street_address>
                        <street_address verid="20">
                        ]3^DiSjZMQK}hkpxlwB9f|rJj?M+2,N@</street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="152">+10(783)41845260</phone_number>
                <email_address>BABABABABAOGOG@monmouth.edu</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items>
```

```
<related_item verid="234">
    <timestamp vtbegin="2002-03-01" vtend="9999-12-31"/>
    <item_id verid="235">I1959</item_id>
</related_item>
```

```
        </related_items>
        <pricing quantity_in_stock="11">
            <cost currency="Dollars">2704.13</cost>
            <when_is_available>1963-11-04</when_is_available>
        </pricing>
    </item>
</catalog>
```

B.2.5 Program of Modification in Sequenced Mode

Here we test τDOM's functions in sequenced mode, including changing content of element, appending child, removing element, adding attribute, modifying attribute and removing attribute. The sequential applicability is from 2002-04-05 through 2004-05-04. The changes made to the document comprise the following.

1. Change the address of an author.
2. Change the phone number of an author.
3. Add a related item.
4. Remove a related item.
5. Add an attribute.
6. Change the value of an attribute.
7. Remove an attribute.

```
/*
 * The TDOM API Sample, Version 1.0
 *
 * Copyright (c) 1999-2002
 * The Computer Science Department of University of Arizona.
 * All rights reserved.
 *
 * TDOM APIs extend DOM API to support operations on temporal XML document.
 * For more information about TDOM API and temporal XML document,
 * visit the homepage of TAU Project <http://www.cs.arizona.edu/tau/>.
 */


import org.w3c.dom.*;
import org.apache.xerces.parsers.*;
import javax.xml.parsers.*;
import org.w3c.dom.ls.*;
import java.util.*;
import java.text.*;
import java.io.*;

/**
 *  SequencedModeTest reads the output document of Evolve2to3,
```

```
 *  set the period of applicability
 *  makes some changes and writes out the document.
 */
public class SequencedModeTest
{
    static TDocumentImpl tdoc;

    public static void main(String[] args) throws Exception{
        /**
         *  parse the temporal document generated by Evolve2to3
         */
        DOMParser parser = new DOMParser();
        parser.parse("catalog.xml");

        Document snap_doc = parser.getDocument();

        tdoc = new TDocumentImpl(snap_doc);

        /*
         *  set period of applicability
         */
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

        tdoc.setModeSequenced("yyyy-MM-dd", "2002-04-05", "2004-05-04");

        /**
         *  change the address of the second author of the second item
         */
        change_address();

        /**
         *  change the phone number of the first author of the third item
         */
        change_phone_number();

        /**
         *  add a related item to the third item
         */
        add_related_item();

        /**
         *  remove the first related item of the first item
         */
        remove_related_item();

        /**
         *  add confirmed address attribute to the second author of the second item
         */
        add_confirmed_address();

        /**
         *  change the name_of_attribute of the first author of the second item
         */
        change_name_of_country();
```

```
    /**
     *  remove the name_of_country attribute of the second author of the third item
     */
    remove_name_of_country();

    tdoc.writeTree();
}

static void change_address(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(1);

    NodeList children = item2.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    children = authors.getChildNodes();
    Element author2 = (Element)children.item(1);

    Element street = (Element)author2.getElementsByTagName("street_address").item(0)

    NodeList cl = street.getChildNodes();

    Text t = (Text)street.getFirstChild();

    t.setData("1234 1st ave, New York");
}

static void change_phone_number(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(2);

    NodeList children = item2.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    children = authors.getChildNodes();
    Element author1 = (Element)children.item(0);

    Element phone = (Element)author1.getElementsByTagName("phone_number").item(0);
    Text t = (Text)phone.getFirstChild();
```

```
    t.setData("+56(344)19242914");
}

static void add_related_item(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(2);

    NodeList children = item2.getChildNodes();

    Element related_items = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("related_items")){
            related_items = (Element)children.item(i);
            break;
        }

    Element new_related_item = tdoc.createElement("related_item");
    Element item_id = tdoc.createElement("item_id");

    item_id.appendChild(tdoc.createTextNode("I886"));
    new_related_item.appendChild(item_id);

    related_items.appendChild(new_related_item);
}

static void remove_related_item(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item0 = (Element)item_nl.item(0);

    NodeList children = item0.getChildNodes();

    Element related_items = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("related_items")){
            related_items = (Element)children.item(i);
            break;
        }

    NodeList rel_item_nl = related_items.getElementsByTagName("related_item");

    related_items.removeChild(rel_item_nl.item(0));
}

static void add_confirmed_address(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item1 = (Element)item_nl.item(1);

    NodeList children = item1.getChildNodes();

    Element authors = null;
```

```
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    children = authors.getChildNodes();
    Element author1 = (Element)children.item(1);

    Element mailing_address = (Element)author1.getElementsByTagName
                              ("mailing_address").item(0);

    mailing_address.setAttribute("confirmed", "yes");
}

static void change_name_of_country(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item1 = (Element)item_nl.item(1);

    NodeList children = item1.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    Element author0 = (Element)authors.getChildNodes().item(0);

    Element street_info = (Element)author0.getElementsByTagName
                  ("street_information").item(0);
    street_info.setAttribute("name_of_country", "England");
}

static void remove_name_of_country(){
    NodeList item_nl = tdoc.getElementsByTagName("item");

    Element item2 = (Element)item_nl.item(2);

    NodeList children = item2.getChildNodes();

    Element authors = null;
    for (int i = 0; i < children.getLength(); i++)
        if (children.item(i).getNodeName().equals("authors")){
            authors = (Element)children.item(i);
            break;
        }

    Element author1 = (Element)authors.getChildNodes().item(1);

    Element street_info = (Element)author1.getElementsByTagName
                              ("street_information").item(0);
```

```
        street_info.removeAttribute("name_of_country");
    }
}
```

B.2.6 The Document after Modification in Sequenced Mode

The changes made in this step are enclosed in boxes.

```
<?xml version="1.0"?>
<catalog cur_date="2002-01-01" cur_verid="236">
    <item id="I1959" subject="LITERATURE" verid="0">
        <title>Unable,newusersBABABABAALNGOGmake.Annual,particular</title>
        <authors>
            <author>
                <name>
                    <first_name>Alla</first_name>
                    <middle_name>fu</middle_name>
                    <last_name>BABABABABAOGINXEbH</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="207">
                        <street_information name_of_country="Chile">
                            <street_address verid="10">
                                <timestamp vtbegin="2002-01-01" vtend="2002-02-01"/>
                                Cs:Q}:_}Z=zI++@}J/dUJ(].lU[P</street_address>
                            <street_address verid="10">
                                <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
                                1234 1st ave, New York</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="145">+71(81)96767976</phone_number>
                    <email_address>BABABABABAOGINXEbH@uni-sb.de</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Jianan</first_name>
                    <middle_name>sentiments would d</middle_name>
                    <last_name>BABABABABAULRI</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="208">
                        <street_information name_of_country="Venezuela">
                            <street_address verid="11">S5!U+uhLGI;g{y$4
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="146">+41(149)33135070</phone_number>
                    <email_address>BABABABABAULRI@concordia.ca</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Anind</first_name>
                    <middle_name>careful a</middle_name>
```

```
                <last_name>BABABABAALBAIN</last_name>
            </name>
            <contact_information>
                <mailing_address verid="209">
                    <street_information name_of_country="Croatia">
                        <street_address verid="12">OnFn+M,GjimXh#R?ujJ2|}
                        </street_address>
                        <street_address verid="13">
                        k7q;4?*2t8;3$d9,#mAfcydE@,|e,^6ER!oU6p</street_address>
                    </street_information>
                </mailing_address>
                <phone_number verid="147">+6(889)98249888</phone_number>
                <email_address>BABABABAALBAIN@sun.com</email_address>
            </contact_information>
        </author>
    </authors>
    <related_items>
        <related_item verid="170">
            <timestamp vtbegin="2002-01-01" vtend="2002-03-01"/>
            <item_id>I886</item_id>
        </related_item>
```

```
<related_item verid="171">
    <timestamp vtbegin="2002-01-01" vtend="2002-04-05"/>
    <item_id>I172</item_id>
</related_item>
<related_item verid="171">
    <timestamp vtbegin="2004-05-04" vtend="9999-12-31"/>
    <item_id>I172</item_id>
</related_item>
```

```
    </related_items>
    <pricing quantity_in_stock="25">
        <cost currency="Dollars">2836.94</cost>
        <when_is_available>1968-02-18</when_is_available>
    </pricing>
</item>
<item id="I886" subject="BUSINESS" verid="1">
    <title>Social,possibleBABABABAALATULchildrenfor</title>
    <authors>
        <author>
            <name>
                <first_name>Madhura</first_name>
                <middle_name>multip</middle_name>
                <last_name>BABABABAOGSEBA</last_name>
            </name>
            <contact_information>
                <mailing_address verid="210">
                    <street_information>
```

```
<timevarying-attr name="name_of_country"
        value="England" vtbegin="2002-04-05" vtend="2004-05-04"/>
<timevarying-attr name="name_of_country"
        value="Eastern Caribbean" vtbegin="2002-01-01" vtend="2002-04-05"/>
<timevarying-attr name="name_of_country"
        value="Eastern Caribbean" vtbegin="2004-05-04" vtend="9999-12-31"/>
```

```
                        <street_address verid="14">
                            ]o=,4sXBˆIe6x4xKkTRpH6Ht˜Ikw0p*PmH3M7 n-</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="148">+69(444)48546076</phone_number>
                    <email_address>BABABABAOGSEBA@inria.fr</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Debbie</first_name>
                    <middle_name>permane</middle_name>
                    <last_name>BABABABAOGATAL</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="211">
```

```
<timevarying-attr name="confirmed" value="yes"
        vtbegin="2002-04-05" vtend="2004-05-04"/>
```

```
                        <street_information name_of_country="Armenia">
```

```
<street_address verid="15">
    <timestamp vtbegin="2002-01-01" vtend="2002-04-05"/>
    zc/RyT+zt42_d?ˆˆqX/P]9B
    </street_address>
<street_address verid="15">
    <timestamp vtbegin="2002-04-05" vtend="2004-05-04"/>
    1234 1st ave, New York
    </street_address>
<street_address verid="15">
    <timestamp vtbegin="2004-05-04" vtend="9999-12-31"/>
    zc/RyT+zt42_d?ˆˆqX/P]9B
</street_address>
```

```
                            <street_address verid="16">].b/RrRr(k.S(GsomgMp]L
                            </street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="149">
                        <timestamp vtbegin="2002-01-01"
                        vtend="2002-02-01"/>+87(781)19242914</phone_number>
                    <phone_number verid="149">
                        <timestamp vtbegin="2002-02-01"
                        vtend="9999-12-31"/>+56(344)19242914</phone_number>
                    <email_address>BABABABAOGATAL@ubs.com</email_address>
                </contact_information>
            </author>
        </authors>
        <related_items>
            <related_item verid="176">
                <item_id>I172</item_id>
            </related_item>
            <related_item verid="232">
                <timestamp vtbegin="2002-02-01" vtend="9999-12-31"/>
```

```
            <item_id verid="233">I1959</item_id>
        </related_item>
    </related_items>
    <pricing>
        <timevarying-attr name="quantity_in_stock"
                value="50" vtbegin="2002-03-01" vtend="9999-12-31"/>
        <timevarying-attr name="quantity_in_stock"
                value="16" vtbegin="2002-01-01" vtend="2002-03-01"/>
        <cost currency="Dollars">4944.09</cost>
        <when_is_available>1982-08-28</when_is_available>
    </pricing>
</item>
<item id="I172" subject="COOKING" verid="2">
    <title>BA</title>
    <authors>
        <author>
            <name>
                <first_name>Gigliola</first_name>
                <middle_name>furious, </middle_name>
                <last_name>BABABABABAOGIN</last_name>
            </name>
            <contact_information>
                <mailing_address verid="212">
                    <street_information>
                        <timevarying-attr name="name_of_country"
                                value="England" vtbegin="2002-03-01" vtend="9999-12-
                        <timevarying-attr name="name_of_country"
                                value="Ireland" vtbegin="2002-01-01" vtend="2002-03-
                        <street_address verid="17">
                        3?61)gphl 8H1UEY,t80hNA</street_address>
                    </street_information>
                </mailing_address>
                    <phone_number verid="150">
                        <timestamp vtbegin="2002-01-01"
                        vtend="2002-04-05"/>+73(450)67471136</phone_number>
                    <phone_number verid="150">
                        <timestamp vtbegin="2002-04-05"
                        vtend="2004-05-04"/>+56(344)19242914</phone_number>
                    <phone_number verid="150">
                        <timestamp vtbegin="2004-05-04"
                        vtend="9999-12-31"/>+73(450)67471136</phone_number>
                <email_address>BABABABABAOGIN@ogi.edu</email_address>
            </contact_information>
        </author>
        <author>
            <name>
                <first_name>Akeo</first_name>
                <middle_name>st</middle_name>
                <last_name>BABABABAOGULBAXWSl4</last_name>
            </name>
            <contact_information>
                <mailing_address verid="213">
                    <timevarying-attr name="confirmed"
```

```
                        value="yes" vtbegin="2002-02-01" vtend="2002-03-01"/>
                        <street_information>
```

```
<timevarying-attr name="name_of_country"
            value="Poland" vtbegin="2002-01-01" vtend="2002-04-05"/>
<timevarying-attr name="name_of_country"
            value="Poland" vtbegin="2004-05-04" vtend="9999-12-31"/>
```

```
                            <street_address verid="18">Z
                            x8pvbWT5e1dP4^xbv@^6G~0uzl 23}A:7Ih/</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="151">+45(307)102286.5</phone_number>
                    <email_address>BABABABAOGULBAXWSl4@imag.fr</email_address>
                </contact_information>
            </author>
            <author>
                <name>
                    <first_name>Kyung</first_name>
                    <middle_name>rea</middle_name>
                    <last_name>BABABABABAOGOG</last_name>
                </name>
                <contact_information>
                    <mailing_address verid="214">
                        <street_information name_of_country="Bangla Desh">
                            <street_address verid="19">
                            XecJy!COgBwf4#I7p</street_address>
                            <street_address verid="20">
                            ]3^DiSjZMQK}hkpxlwB9f|rJj?M+2,N@</street_address>
                        </street_information>
                    </mailing_address>
                    <phone_number verid="152">+10(783)41845260</phone_number>
                    <email_address>BABABABABAOGOG@monmouth.edu</email_address>
                </contact_information>
            </author>
        </authors>
        <related_items>
            <related_item verid="234">
                <timestamp vtbegin="2002-03-01" vtend="9999-12-31"/>
                <item_id verid="235">I1959</item_id>
            </related_item>
```

```
<related_item>
    <timestamp vtbegin="2002-04-05" vtend="2004-05-04"/>
    <item_id>I886</item_id>
</related_item>
```

```
        </related_items>
        <pricing quantity_in_stock="11">
            <cost currency="Dollars">2704.13</cost>
            <when_is_available>1963-11-04</when_is_available>
        </pricing>
    </item>
</catalog>
```