

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11

MPEG2001/M7048

March 2001, Singapore

Source: Electrotechnical Laboratory and University of Brescia

Title: Abstraction and Instantiation

Author: Kôiti Hasida and Riccardo Leonardi

Status: for discussion

1 Introduction

This document proposes tools for describing abstractions and their instantiations. The merit of abstraction is the economy of description. Abstractions allow you to avoid repeating similar descriptions over and over again. You write one abstraction to address the shared structure of descriptions and reuse it when you want similar descriptions. Suppose you want to describe many occurrences of a common pattern of events and states of affairs, such as a type of configuration of soccer players in the field. You will describe the positions of eleven people to address this configuration. Once you set up an abstraction to describe this common pattern, you can address each occurrence of the pattern by just binding the eleven non-specific people to particular players, without repeating the descriptions of their positions.

2 The Abstraction DS and the Instantiation DataType

The Abstraction DS addresses how to describe abstractions, and the Instantiation DataType addresses how to instantiate those abstractions. That is, an Abstraction element defines a type of descriptions, and an Instantiation element describes an instance of that type. The Abstraction DS and the Instantiation DataType are meta-level tools. That is, the content of an Abstraction element makes no sense as it is, but is interpreted only after instantiation. In particular, AbstractionLevel is not interpreted inside an Abstraction element.

2.1 Syntax

```
<!-- ##### -->
<!-- Definition of the Abstraction DS -->
<!-- ##### -->

<complexType name="AbstractionType">
  <complexContent>
    <extension base="mpeg7:DSType">
      <choice minOccurs="1" maxOccurs="unbounded">
        <element name="Segment" type="mpeg7:SegmentType"
          minOccurs="0" maxOccurs="1"/>
        <element name="SemanticBase" type="mpeg7:SemanticBaseType"
          minOccurs="0" maxOccurs="1"/>
        <element name="LinguisticEntity"
          type="mpeg7:LinguisticEntityType"
          minOccurs="0" maxOccurs="1"/>
        <element name="Relation" type="mpeg7:RelationType"
          minOccurs="0" maxOccurs="1"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

```

        <element name="Abstraction" type="mpeg7:AbstractionType"
            minOccurs="0" maxOccurs="1"/>
        <element name="Instantiation" type="mpeg7:InstantiationType"
            minOccurs="0" maxOccurs="1"/>
    </choice>
</extension>
</complexContent>
</complexType>

<!-- ##### -->
<!-- Definition of the Instantiation DataType -->
<!-- ##### -->

<complexType name="InstantiationType">
    <complexContent>
        <element name="Bind" minOccurs="1" maxOccurs="unbounded">
            <complexType>
                <complexContent>
                    <attribute name="variable" type="IDREFS" use="required"/>
                    <attribute name="value" type="IDREF" use="required"/>
                </complexContent>
            </complexType>
        </element>
        <attribute name="in" type="IDREF" use="required"/>
    </complexContent>
</complexType>

```

2.2 Semantics

Semantics for AbstractionType:

<i>Name</i>	<i>Definition</i>
AbstractionType	Type of abstraction. The elements in it are variables. All the descendant elements are variables, representing nonspecific entities. So they are implicitly regarded as having non-zero dimension values in AbstractionLevel.
SemanticBase	Semantic entity, which actually appears as Event, Object, SemanticPlace, etc.
LinguisticEntity	Linguistic entity, which actually appears as LinguisticDocument, Paragraph, Sentence, etc.
Relation	Relation among Semantic entities, Linguistic entities, Relation entities, and abstractions.
Abstraction	Abstraction.
Instantiation	Instantiation.

Semantics of InstantiationType:

<i>Name</i>	<i>Definition</i>
InstantiationType	Type of instantiation of abstraction. An instantiation is equivalent to the content of the abstraction duplicated while substituting its parts as specified by the bindings. The information in a bound variable is conveyed to the binding value as much as consistent with the information already in the value. See the example given later.
in	Points to the abstraction to instantiate.
Bind	Binding of a variable by a value.
variable	Points to the variable to be bound.
value	Points to the value to bind the variable.

2.3 Example

The following is an abstraction which means that a woman kisses a man, where the woman, the man, and the kissing event are all non-specific.

```
<Abstraction id="WomanKissMan">
  <Object id="woman" .../>
  <Object id="man" .../>
  <Event id="kiss" .../>
    <Relation name="agent" target="woman"/>
    <Relation name="patient" target="man"/>
    ...
  </Event>
</Abstraction>
```

The following example shows how to instantiate this abstraction.

```
<Instantiation in="WomanKissMan">
  <Bind variable="woman" value="mary"/>
  <Bind variable="man" value="tom"/>
</Instantiation>
```

This description means that Mary kisses Tom. That is, it is equivalent to the following.

```
<Event ...>
  <Relation name="agent" target="mary"/>
  <Relation name="patient" target="tom"/>
  ...
</Event>
```

So the search engine should find the above `Instantiation` element in response to a query for "Mary kisses Tom".

3 Conclusion

The CE on the proposed tools will consist of a closer study of their syntax and semantics, some revisions of them, and empirical estimation of their usefulness perhaps in reference to the Monster Description data. The last item will evaluate how the tools contribute to the reduction of the complexity of the description. Our recommendation is to approve this CE.