Semi-structured Data

6 - XPath

Mantas Šimkus

Outline

- XPath Terminology
- XPath at First Glance
- Location Paths (Axis, Node Test, Predicate)
- Abbreviated Syntax

What is XPath?

- A language for extracting parts of an XML document

- A basic query language for XML - plays the same role as the SQL SELECT statement plays for relational databases

- An important component of other XML-related technologies (such as XSD, XQuery and XSLT)

- As expected, XPath is a W3C standard

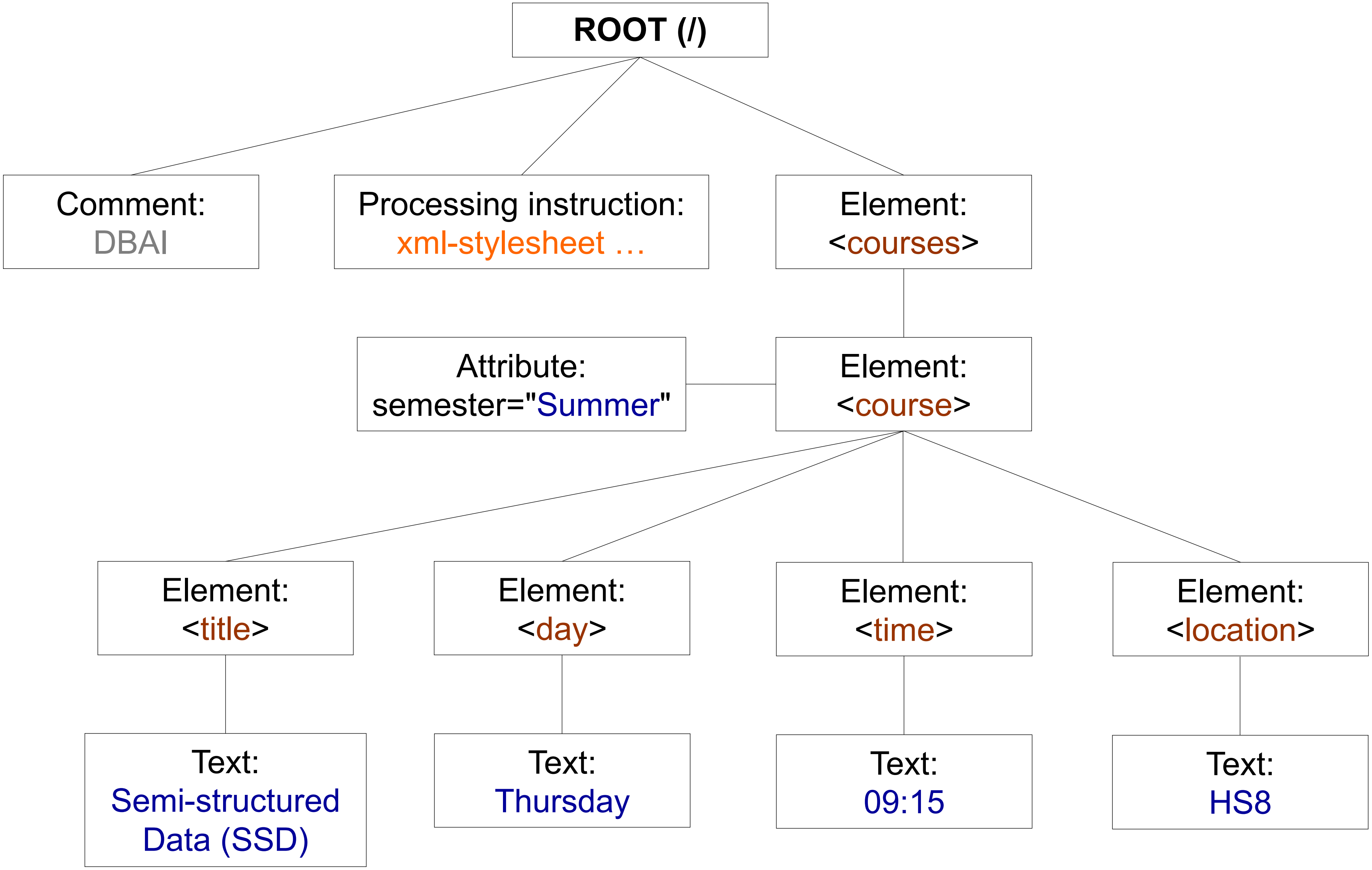
XPath Terminology

- XML documents are treated as trees of nodes
- There are seven kinds of nodes:
  - Document nodes
  - Element nodes
  - Attribute nodes
  - Text nodes
  - Namespace nodes
  - Processing-instruction nodes
  - Comment nodes

XPath Terminology - Nodes

```xml
<?xml version="1.0"?>
<!-- DBAI -->
<?xml-stylesheet href="course_style.css" type="text/css"?>
<courses>
    <course semester="Summer">
        <title> Semi-structured Data (SSD) </title>
        <day> Thursday </day>
        <time> 09:15 </time>
        <location> HS8 </location>
    </course>
</courses>
```

XPath Terminology - Nodes



ROOT (/)

Comment:
DBAI

Processing instruction:
xml-stylesheet …

Element:
<courses>

Attribute:
semester="Summer"

Element:
<course>

Element:
<title>

Element:
<day>

Element:
<time>

Element:
<location>

Text:
Semi-structured Data (SSD)

Text:
Thursday

Text:
09:15

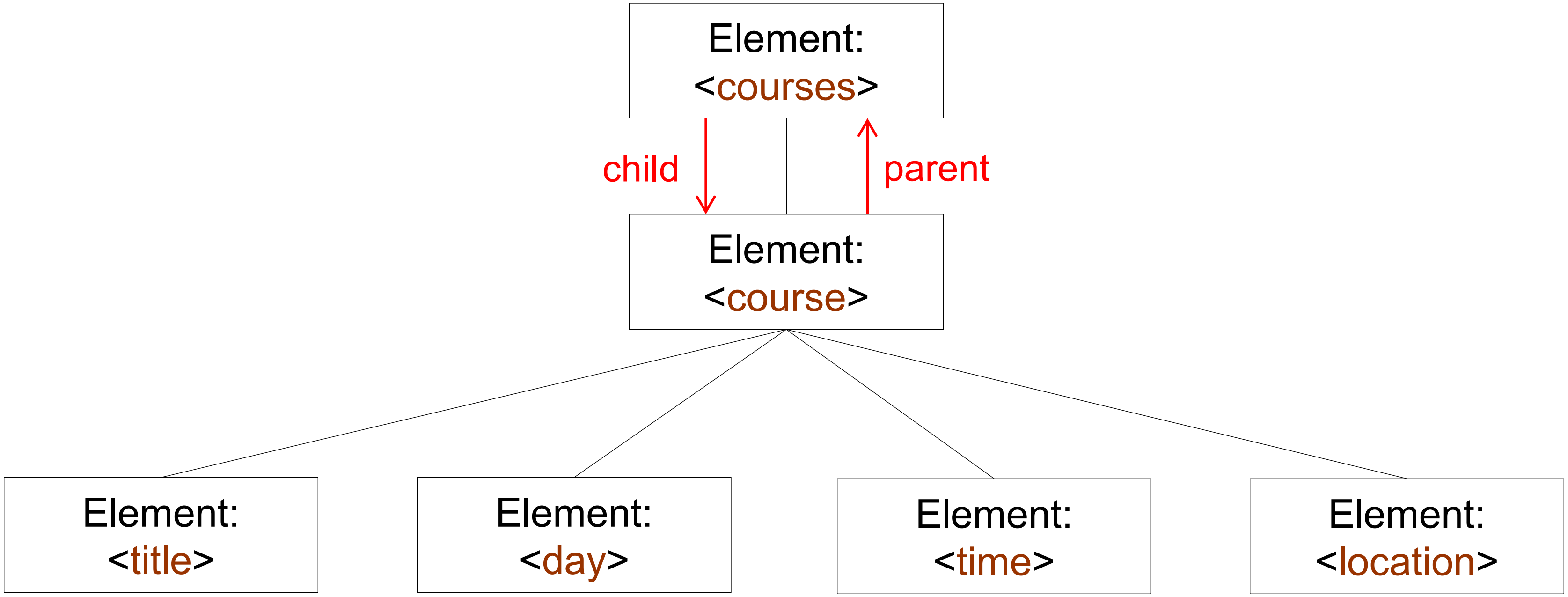
Text:
HS8

Relationships Among Nodes

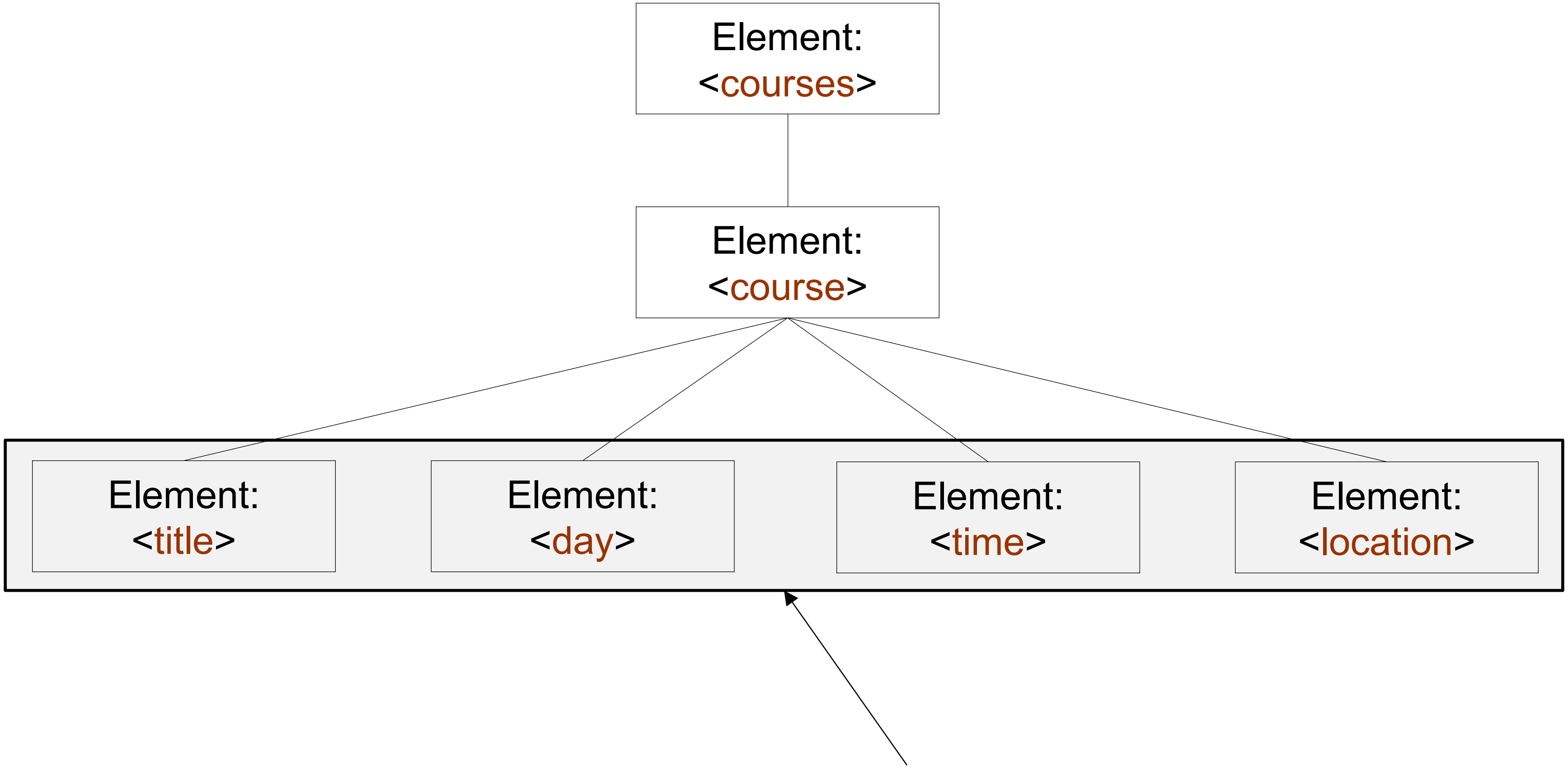
- The terms parent, child, sibling, ancestor and descendant are describing the relationships among nodes

- In an XML tree:
  - Every node has exactly one parent (except the root)
  - A node can have an unbounded number of children
  - A leaf node has no children
  - Siblings have the same parent

Relationships Among Nodes



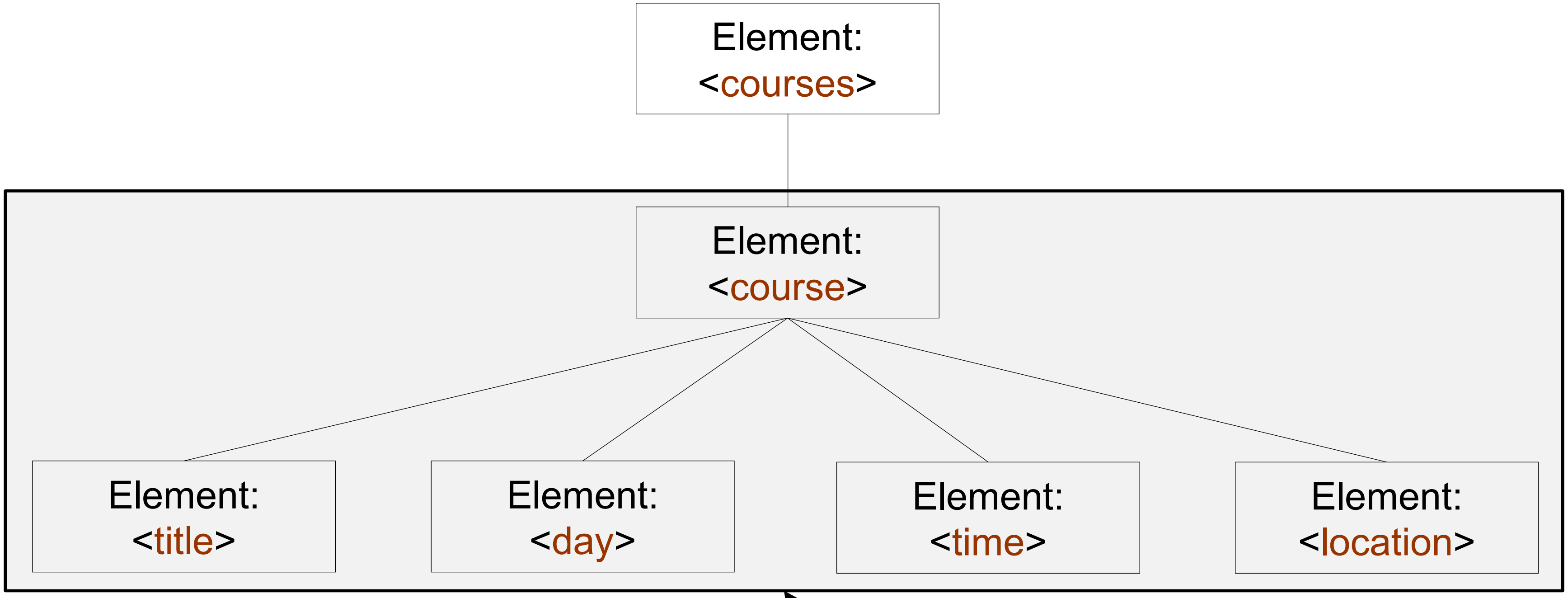
Relationships Among Nodes



child nodes to <course>

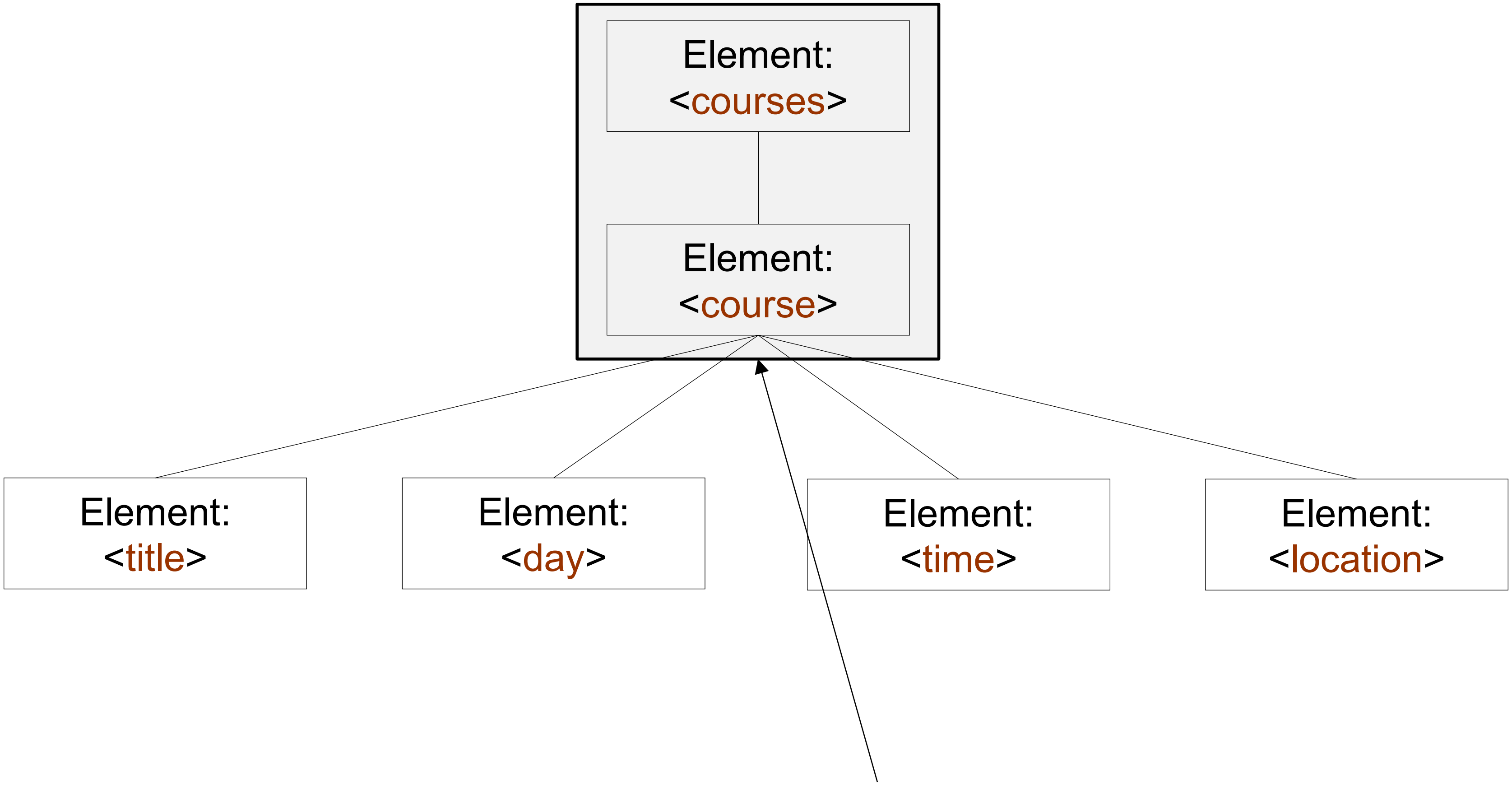
sibling nodes to each other

Relationships Among Nodes



descendant nodes to <courses>

Relationships Among Nodes



Element:
<courses>

Element:
<course>

Element:
<title>

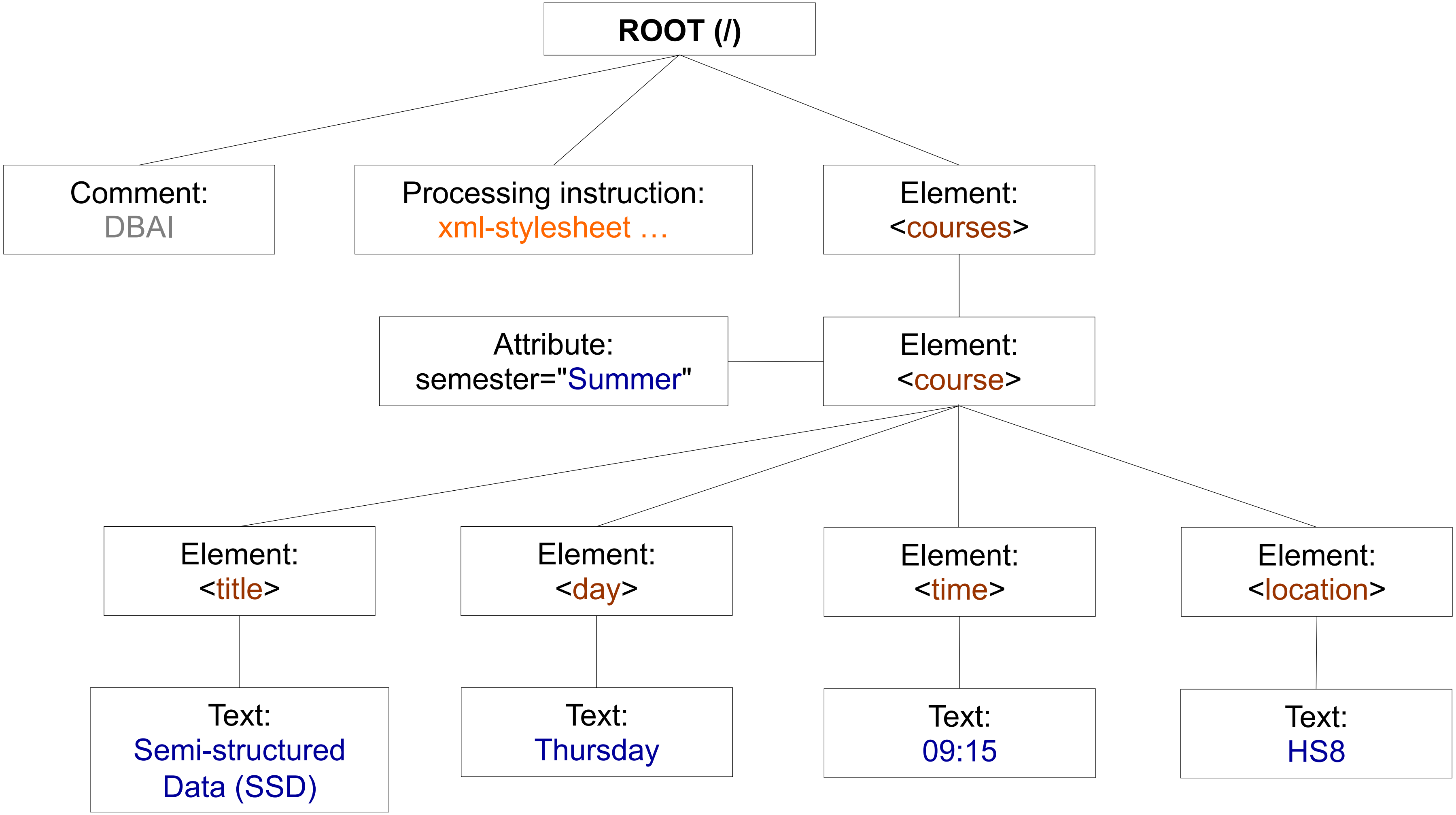
Element:
<day>

Element:
<time>

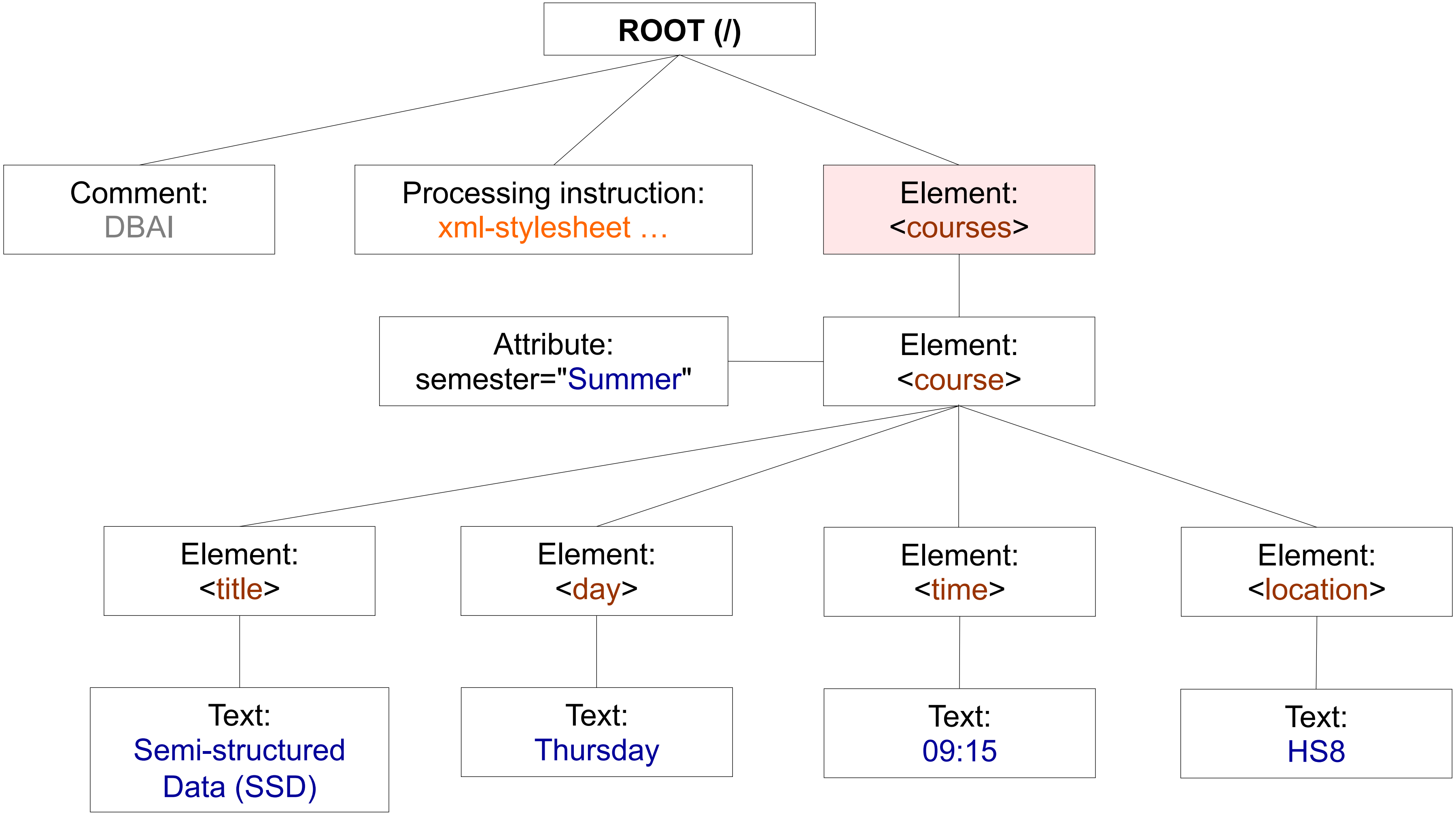
Element:
<location>

ancestor nodes to <title>, <day>, <time> and <location>

XPath at First Glance

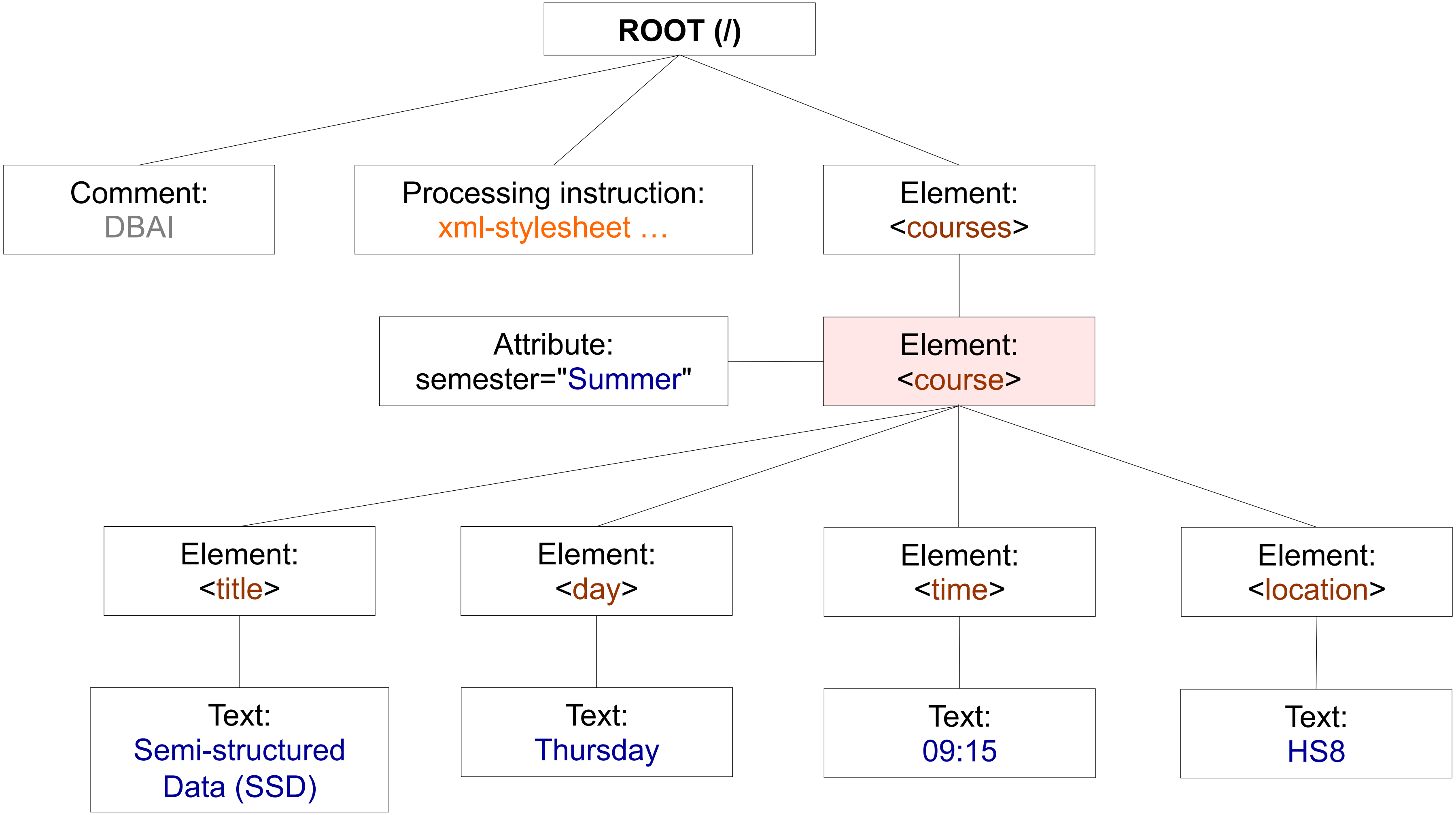


XPath at First Glance



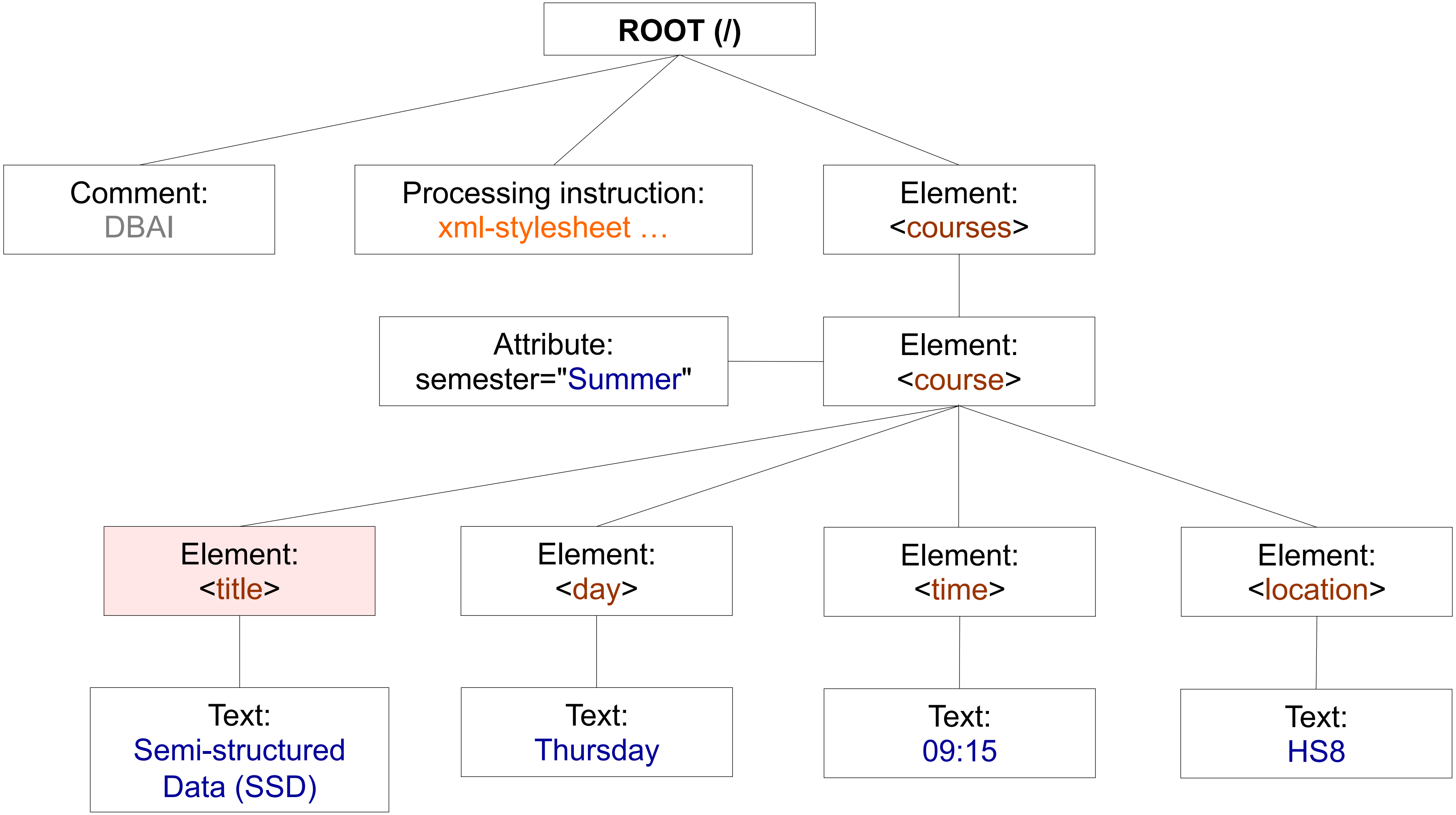
/child::courses

XPath at First Glance



/child::courses/child::course

XPath at First Glance



ROOT (/)

Comment:
DBAI

Processing instruction:
xml-stylesheet …

Element:
<courses>

Attribute:
semester="Summer"

Element:
<course>

Element:
<title>

Element:
<day>

Element:
<time>

Element:
<location>

Text:
Semi-structured Data (SSD)

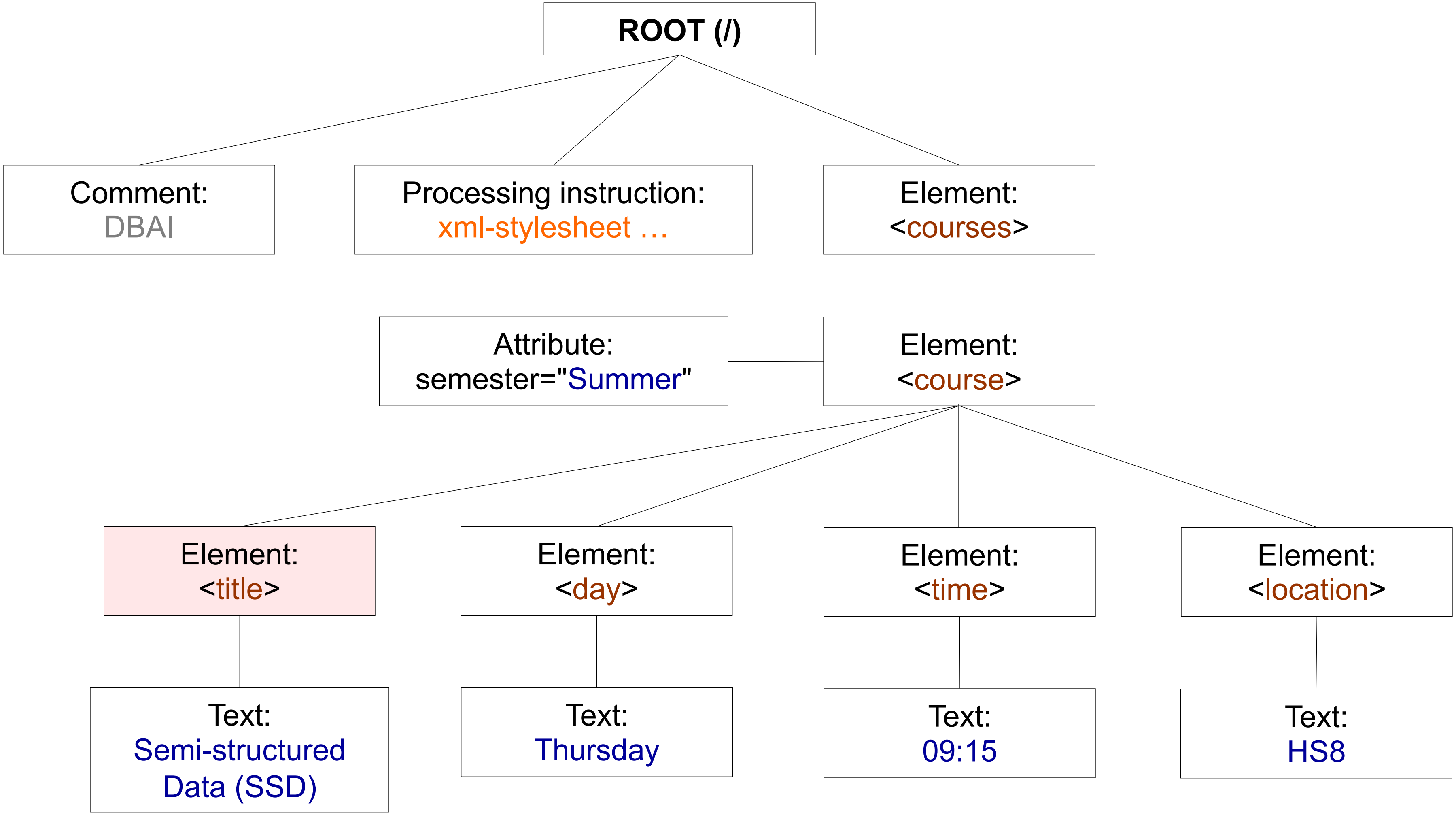
Text:
Thursday

Text:
09:15

Text:
HS8

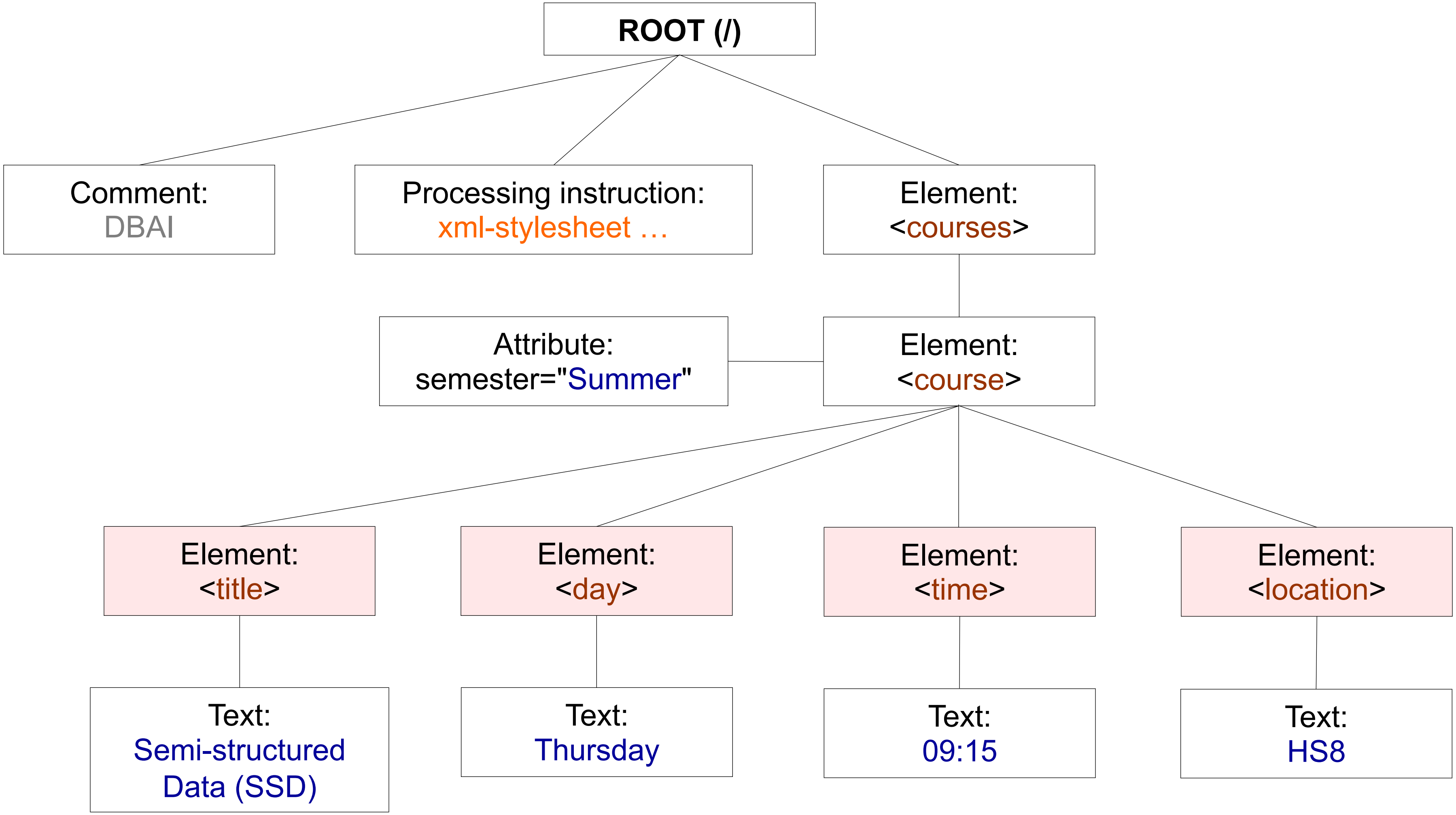
/child::courses/child::course/child::title

XPath at First Glance



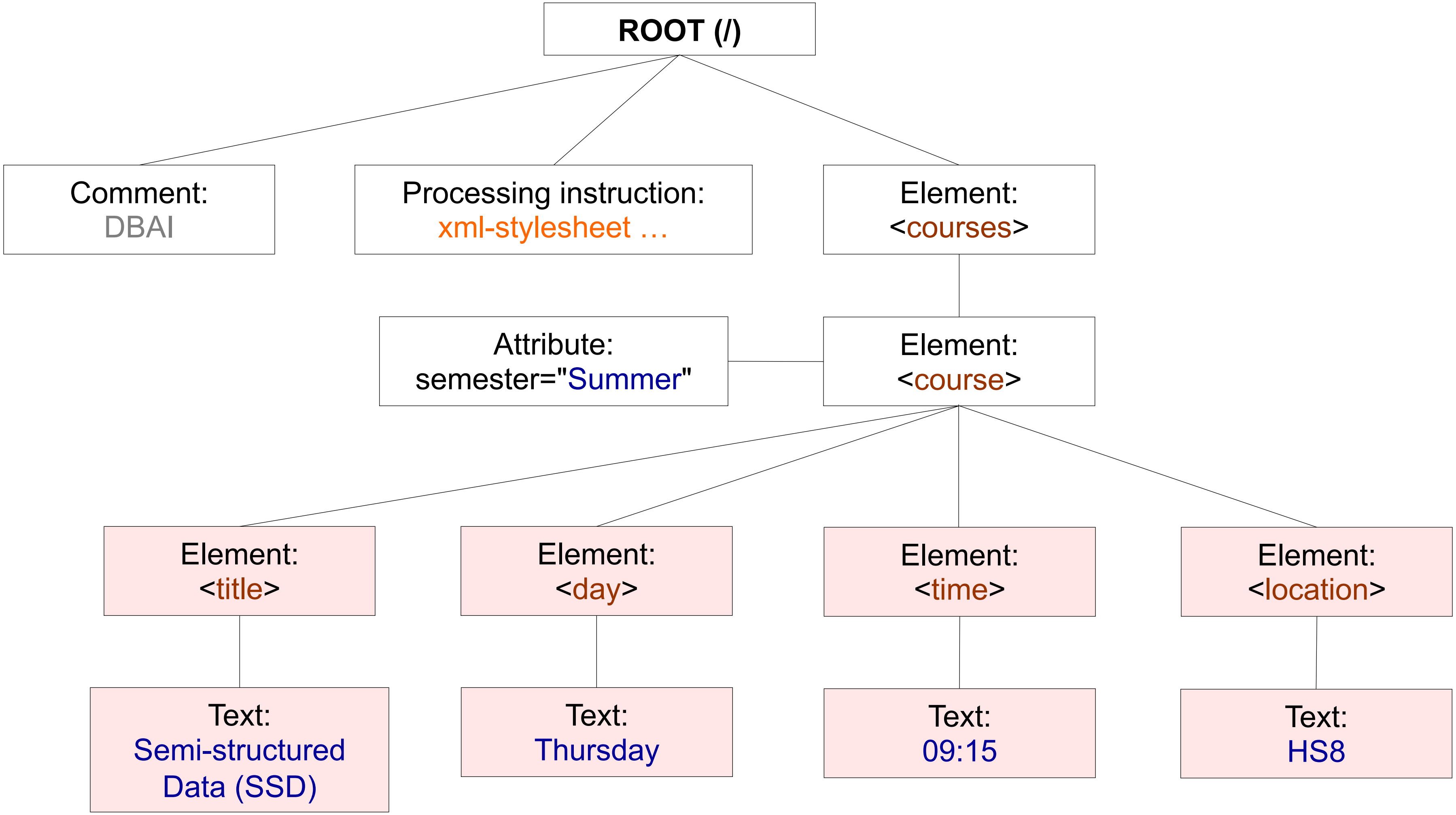
/descendant::course/child::title

XPath at First Glance



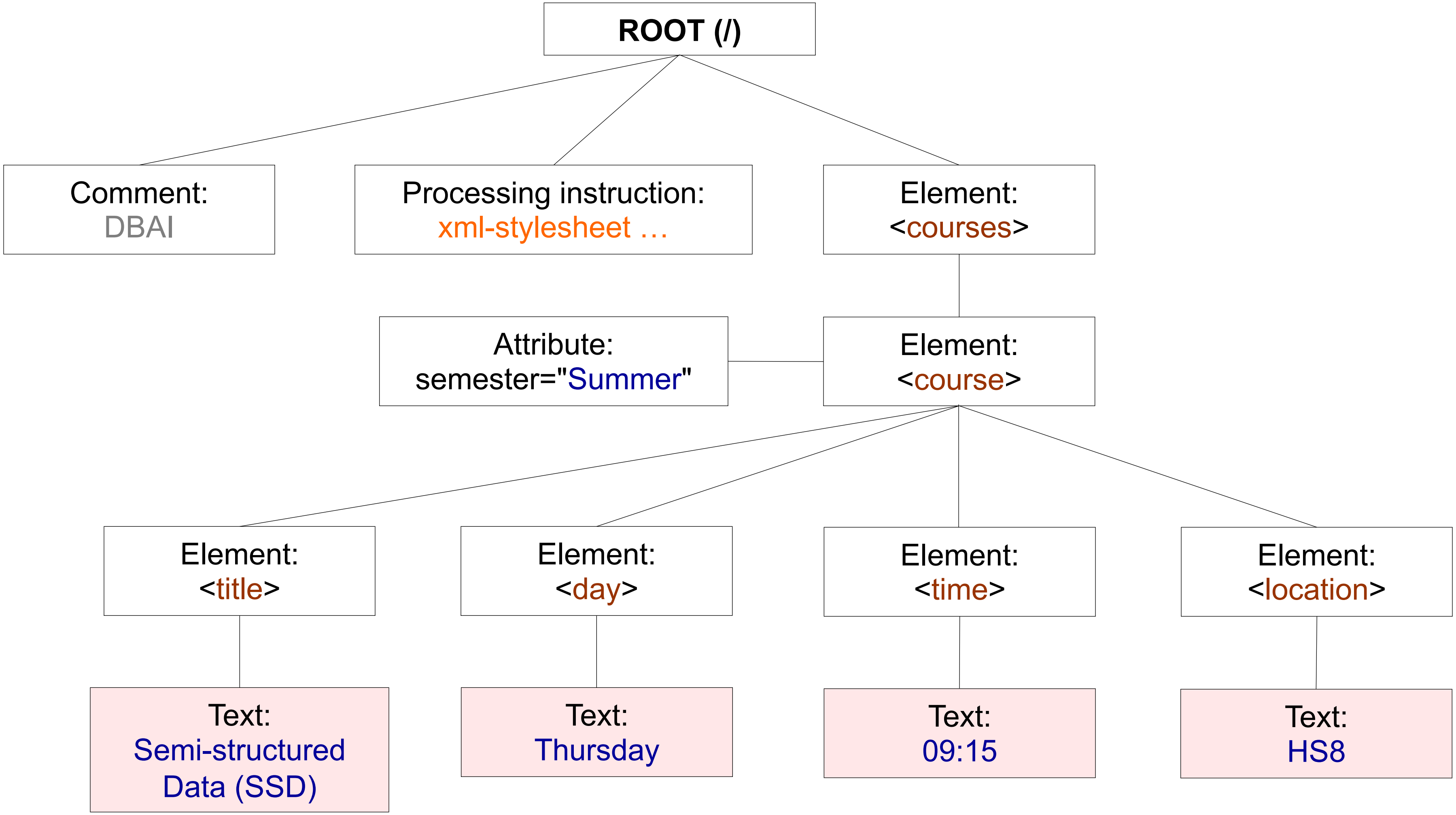
/descendant::course/child::*

XPath at First Glance



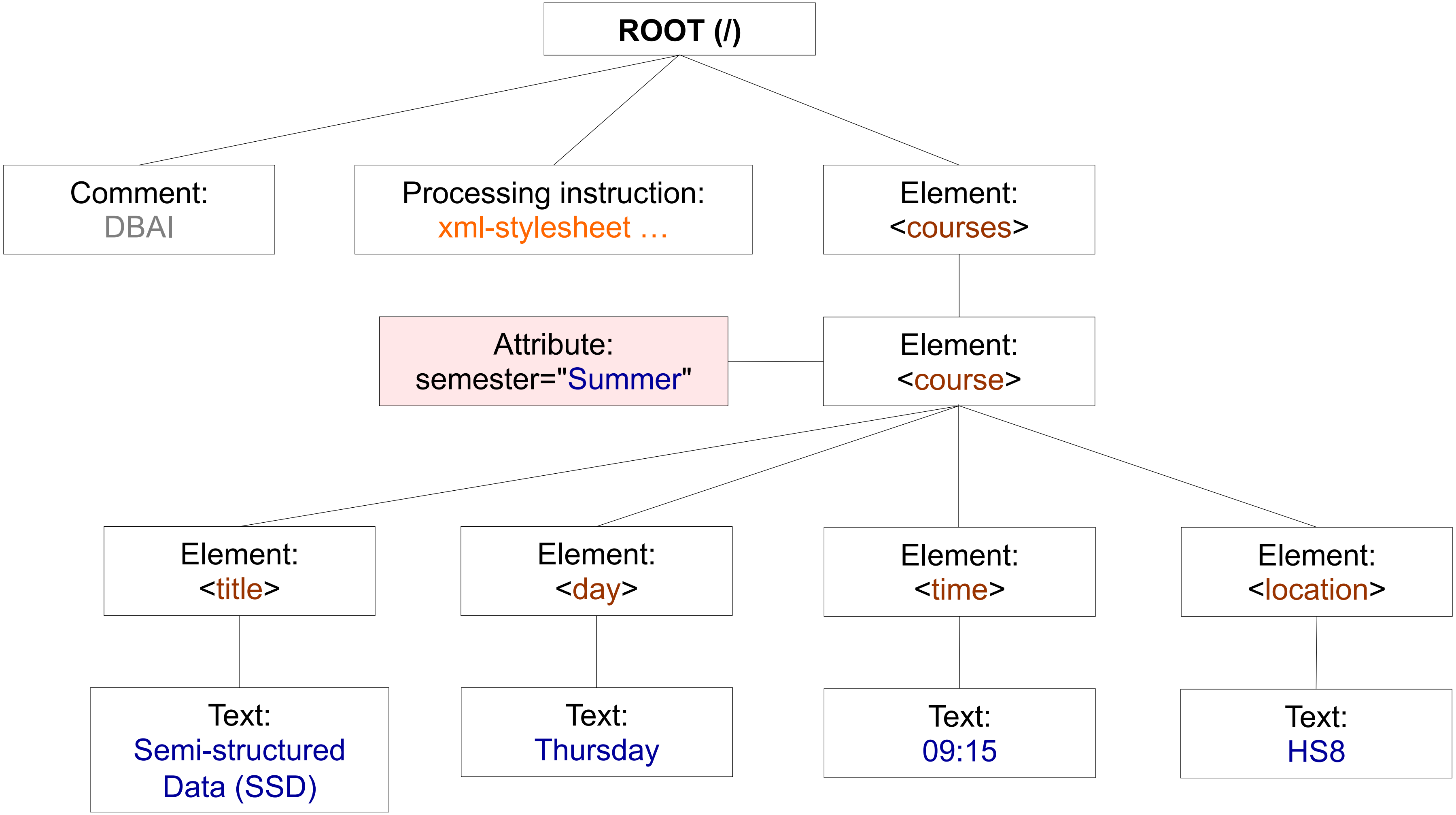
/descendant::course/descendant::node()

XPath at First Glance



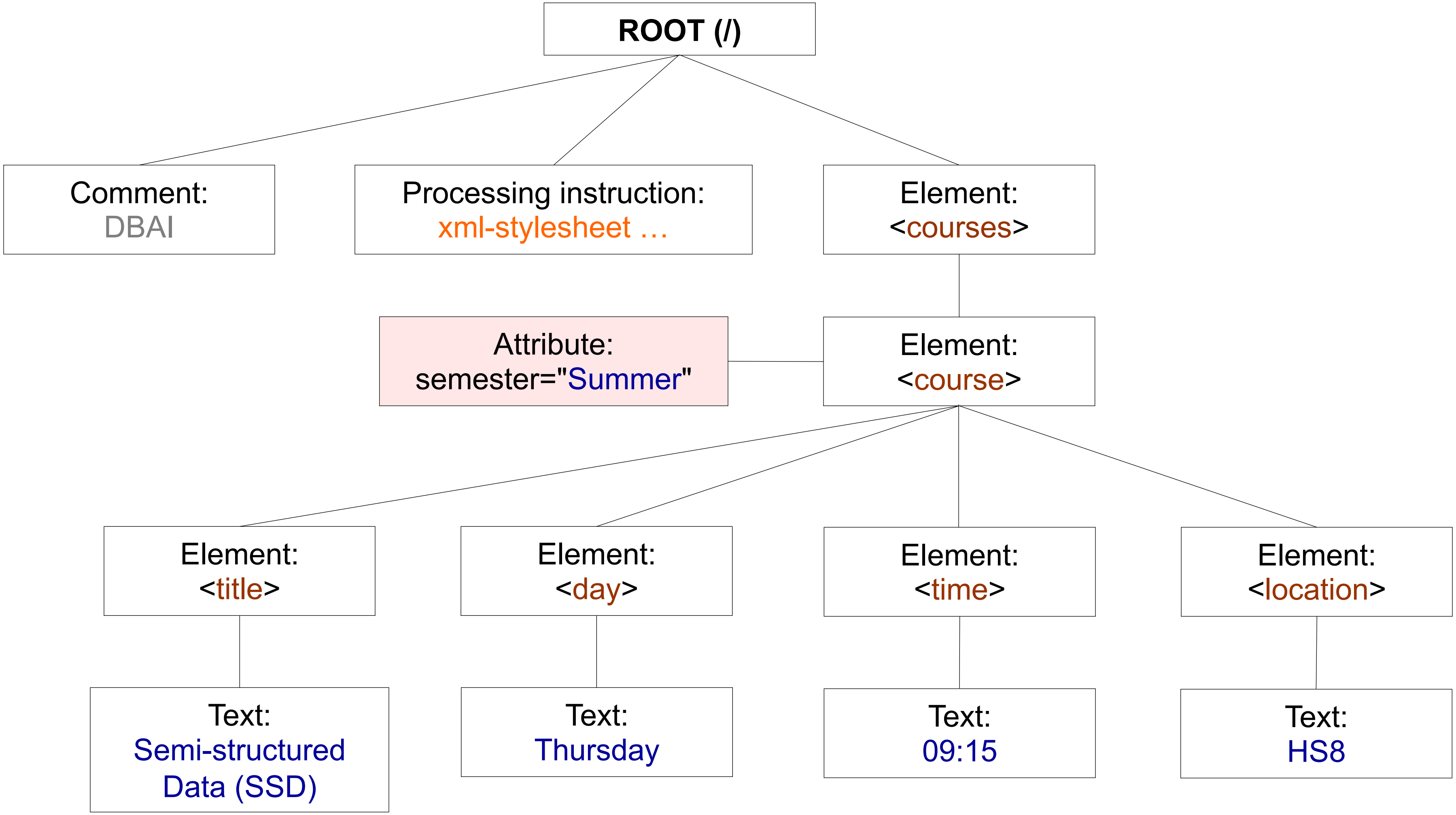
/descendant::course/descendant::text()

XPath at First Glance



/child::courses/child::course/attribute::semester

XPath at First Glance



ROOT (/)

Comment:
DBAI

Processing instruction:
xml-stylesheet …

Element:
<courses>

Attribute:
semester="Summer"

Element:
<course>

Element:
<title>

Element:
<day>

Element:
<time>

Element:
<location>

Text:
Semi-structured Data (SSD)

Text:
Thursday

Text:
09:15

Text:
HS8

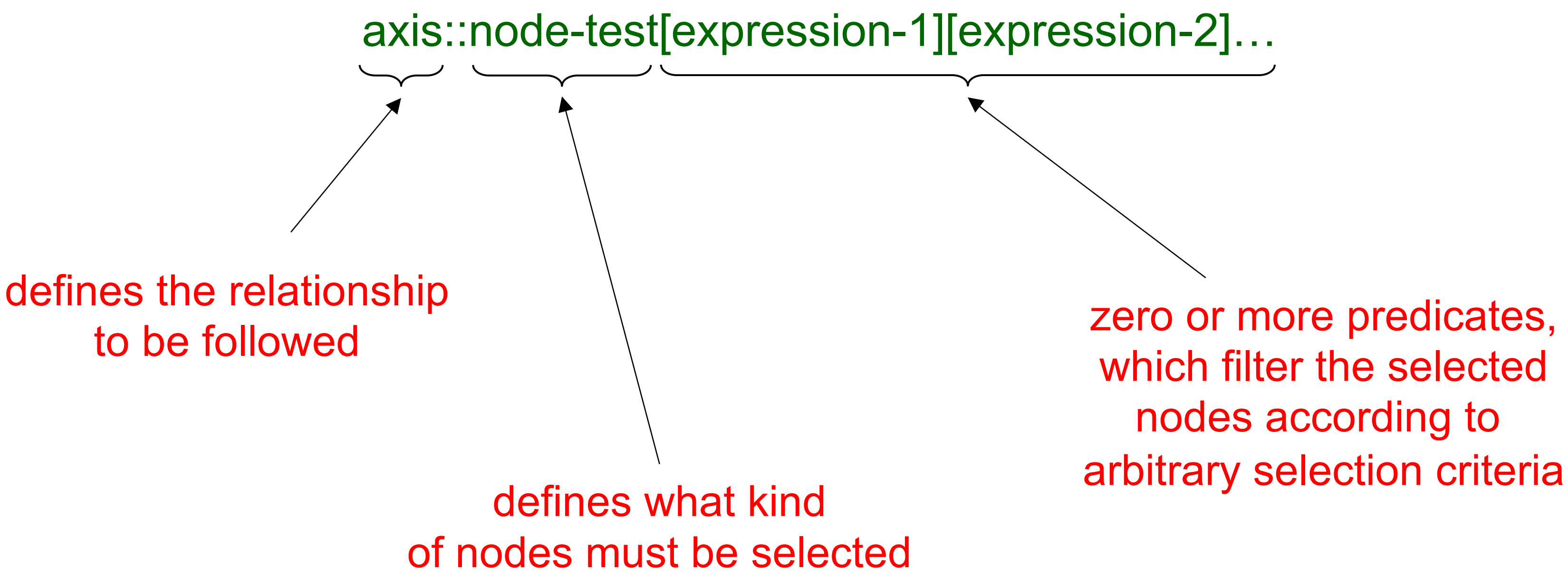
/descendant::course/attribute::semester

Up to Now

- **XPath Terminology**
- **XPath at First Glance**
- Location Paths (Axis, Node Test, Predicate)
- Abbreviated Syntax
- Further Examples

Location Paths

- XPath uses location paths to select nodes in a tree
- A location path is a series of location steps separated by the symbol /
- Each location step has the form



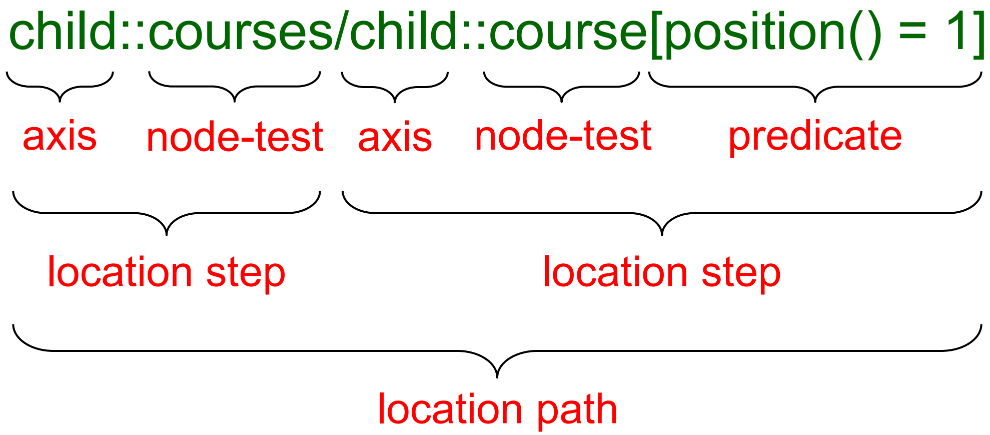
axis::node-test[expression-1][expression-2]…

defines the relationship to be followed

defines what kind of nodes must be selected

zero or more predicates, which filter the selected nodes according to arbitrary selection criteria

The Anatomy of a Location Path



child::courses/child::course[position() = 1]

axis

node-test

axis

node-test

predicate

location step

location step

location path

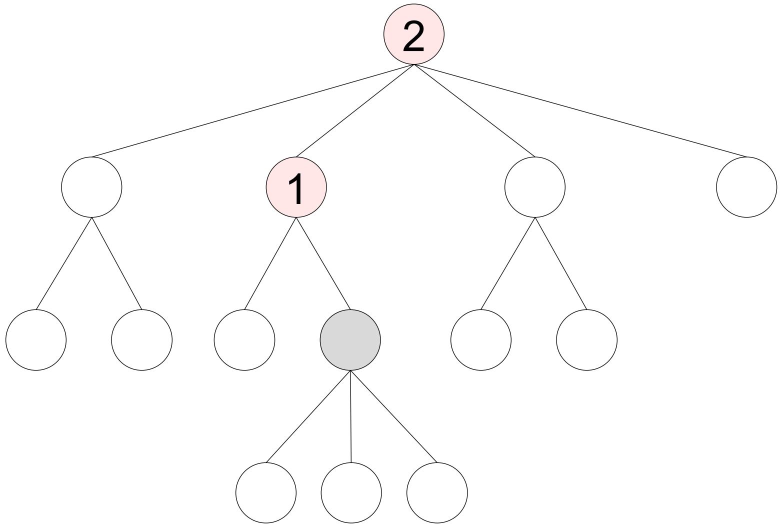
ATTENTION: The first location step does not have a predicate

Axes

- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self

Axes

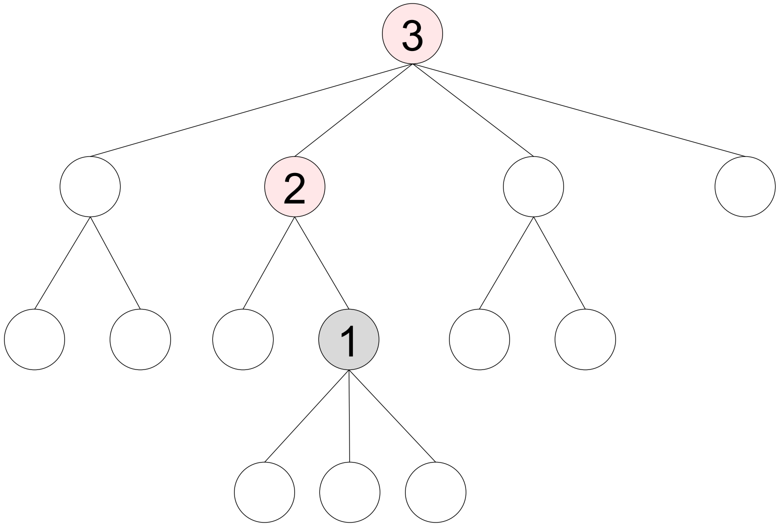
- XPath defines 13 axes:
  - **ancestor**
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects all the nodes that are ancestors of the origin node
- The first node on the axis is the parent of the origin, the second is its grandparent, and so on
- The last node on the axis is the root of the tree

Axes

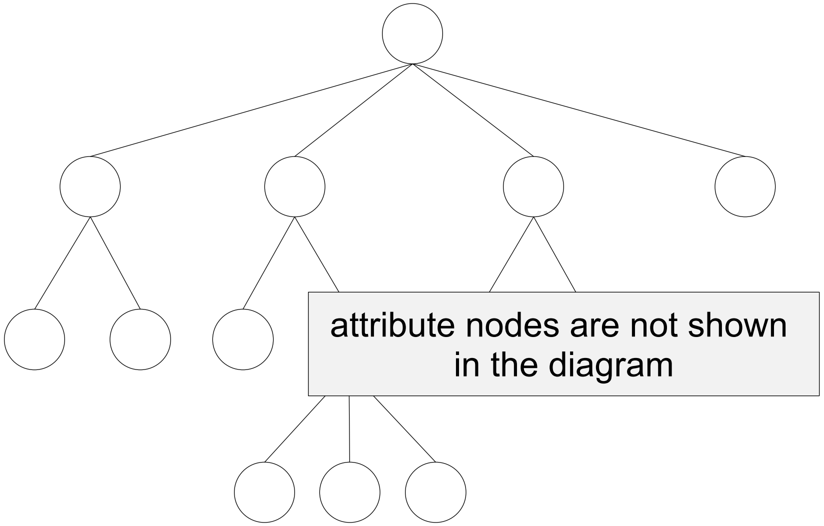
- XPath defines 13 axes:
  - ancestor
  - **ancestor-or-self**
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects the same nodes as the ancestor axis
- … but starting with the origin node (instead of the parent of the origin node)

Axes

- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - **attribute**
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self


attribute nodes are not shown in the diagram

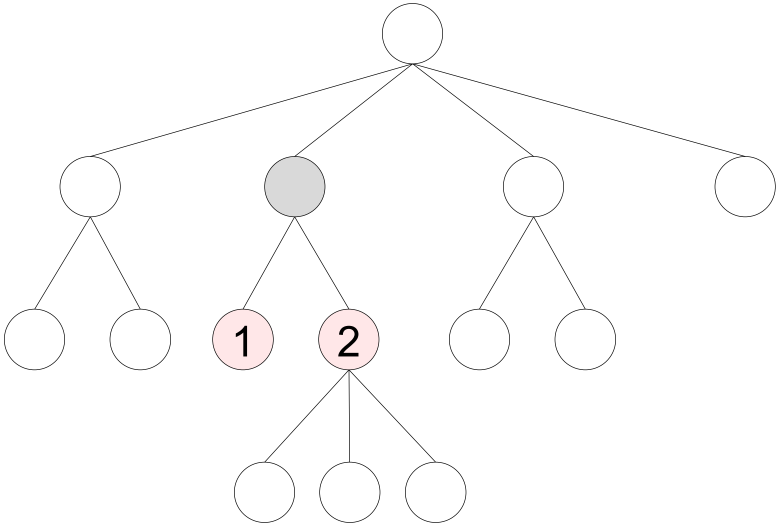
- If the origin is an element node, then this axis selects all its attribute nodes; otherwise, it selects nothing (empty sequence)

- The attributes will not necessarily be in the order in which they appear in the document

- Namespace nodes are not selected

Axes

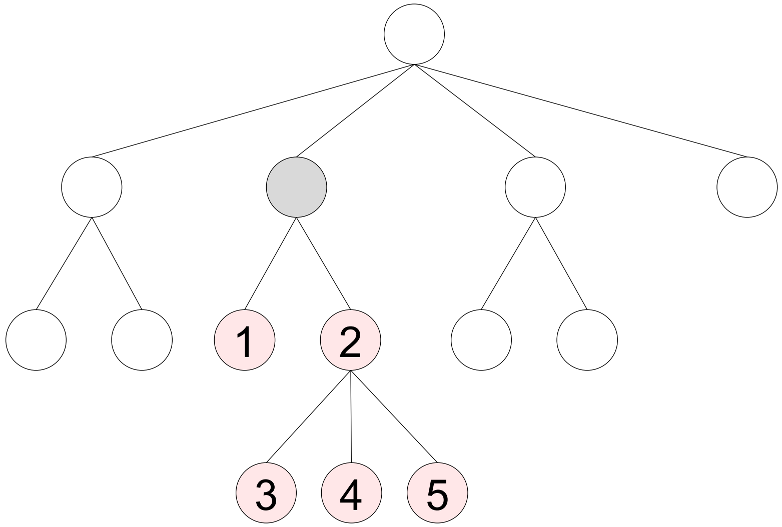
- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - **child**
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects all the children of the origin in document order
- If the origin is other than a document or element node, then this axis selects nothing
- The children of an element node do not include attribute or namespaces

Axes

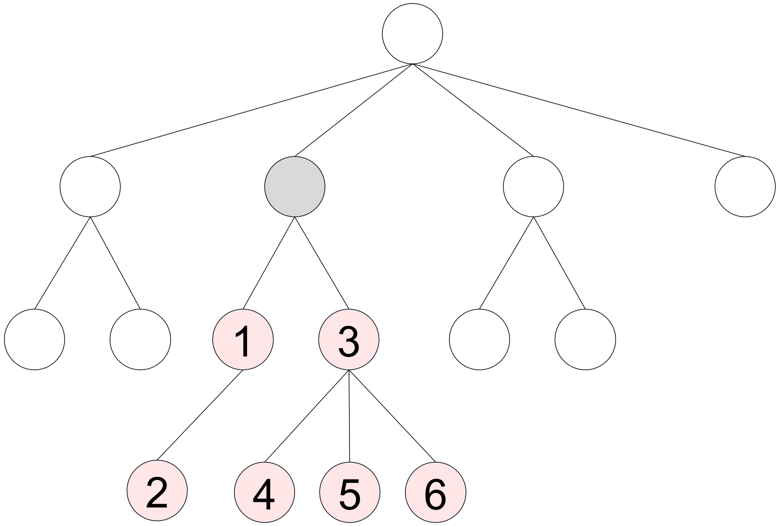
- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - **descendant**
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects all the children of the origin, and their children, and so on recursively in document order

Axes

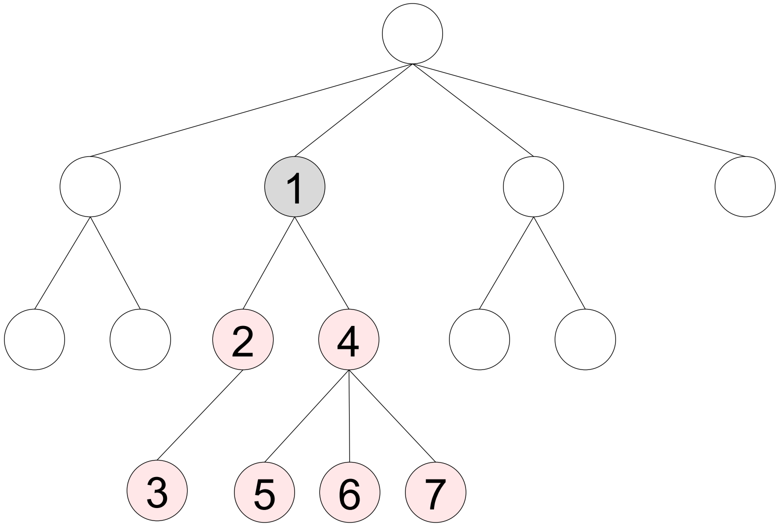
- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - **descendant**
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects all the children of the origin, and their children, and so on recursively in document order

Axes

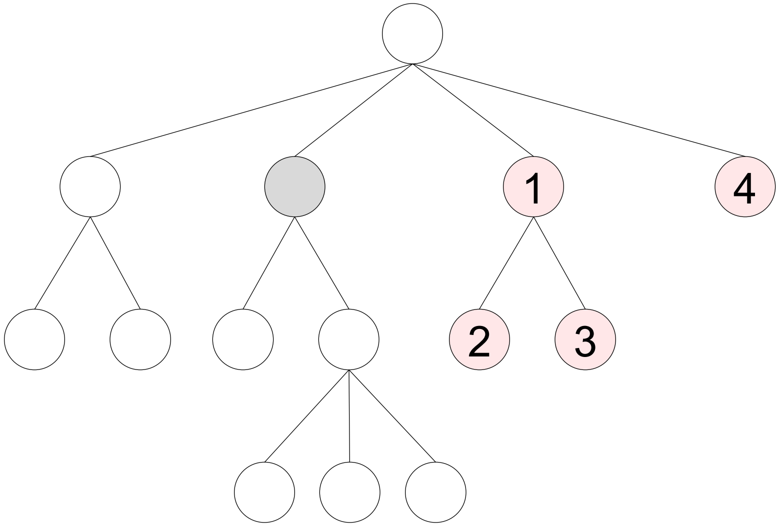
- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - **descendant-or-self**
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self



- Selects the same nodes as the descendant axis, except that the first node selected is the origin

Axes

- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - **following**
  - following-sibling
  - namespace
  - parent
  - preceding
  - preceding-sibling
  - self

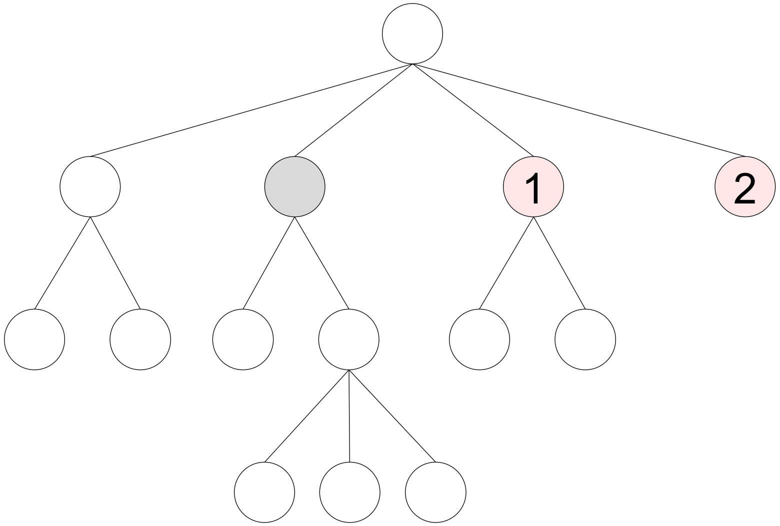


- Selects all the nodes that appear after the origin in document order, excluding the descendants of the origin

- The following axis will never contain attributes or namespaces

Axes

- XPath defines 13 axes:
    - ancestor
    - ancestor-or-self
    - attribute
    - child
    - descendant
    - descendant-or-self
    - following
    - **following-sibling**
    - namespace
    - parent
    - preceding
    - preceding-sibling
    - self

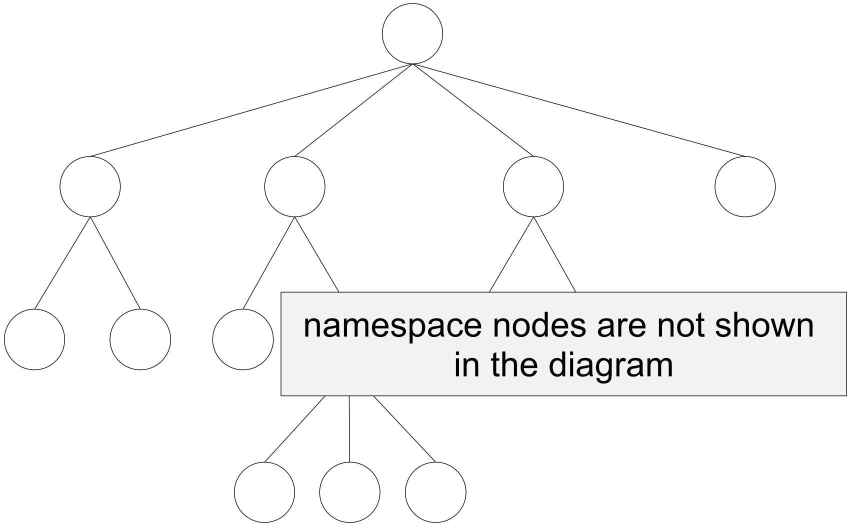


- Selects all the nodes that follow the origin in document order, and that are children of the same parent

- For document, attribute and namespace nodes, this axis is empty

Axes

- XPath defines 13 axes:
    - ancestor
    - ancestor-or-self
    - attribute
    - child
    - descendant
    - descendant-or-self
    - following
    - following-sibling
    - **namespace**
    - parent
    - preceding
    - preceding-sibling
    - self



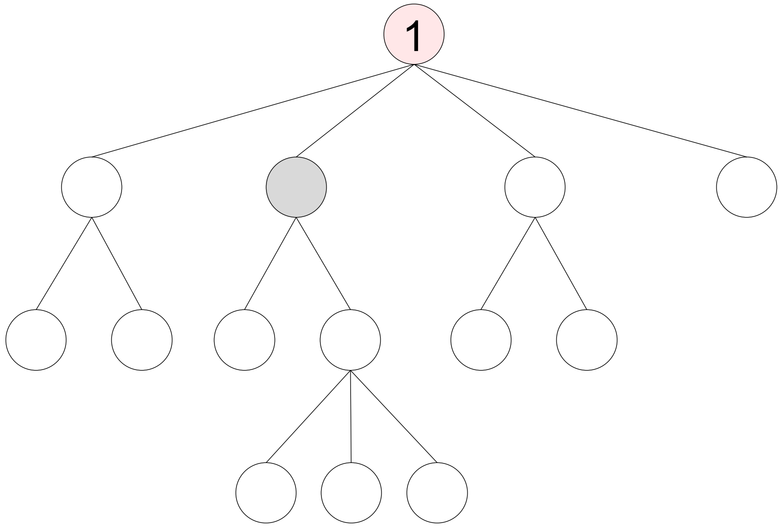
namespace nodes are not shown in the diagram

- If the origin is an element node, then this axis selects all the namespace nodes (or simply, namespaces) that are defined for that element; otherwise, it is empty

- The namespaces will not necessarily be in the order in which they appear in the document

Axes

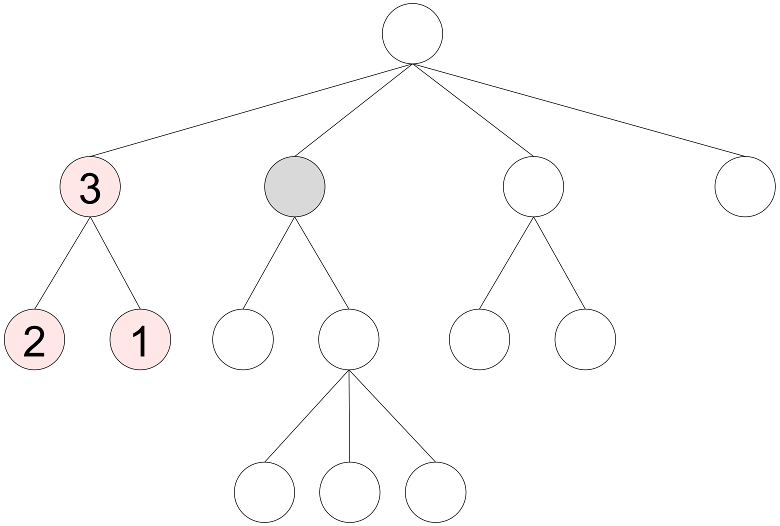
- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - **parent**
  - preceding
  - preceding-sibling
  - self



- Selects the parent of the origin node (i.e., a single node)
- If the origin node does not have a parent, then the parent axis is empty

Axes

- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - **preceding**
  - preceding-sibling
  - self

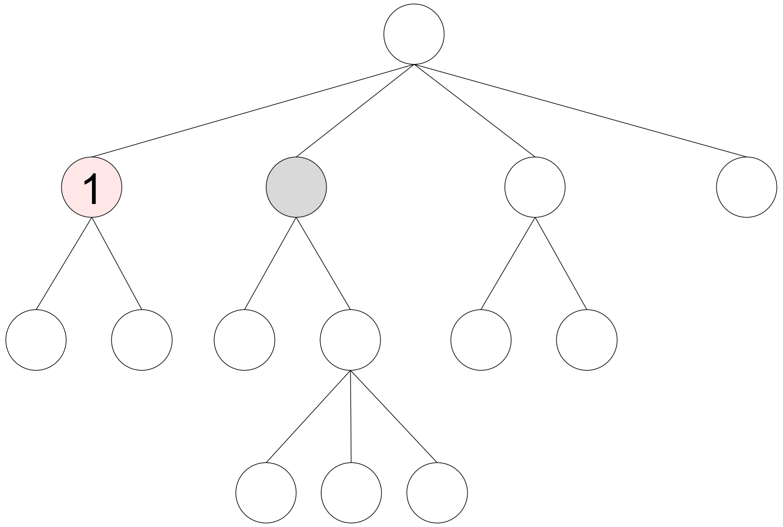


- Selects all the nodes that appear before the origin, excluding the ancestors of the origin node

- The preceding axis will never contain attributes or namespaces

Axes

- XPath defines 13 axes:
  - ancestor
  - ancestor-or-self
  - attribute
  - child
  - descendant
  - descendant-or-self
  - following
  - following-sibling
  - namespace
  - parent
  - preceding
  - **preceding-sibling**
  - self

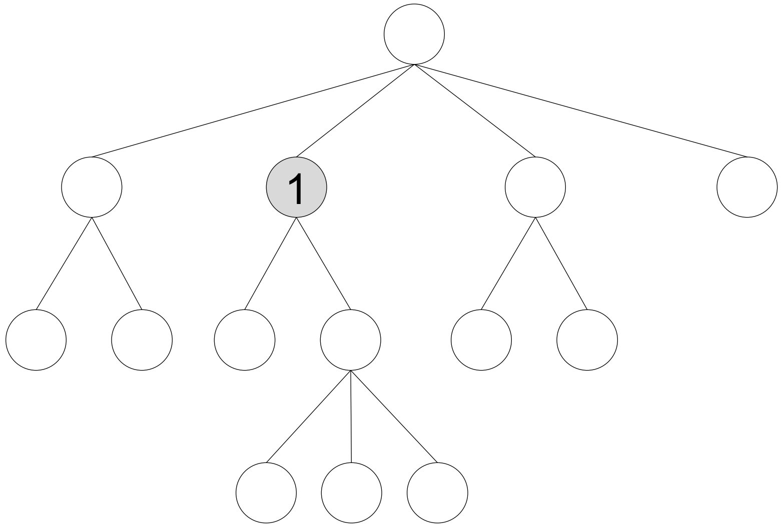


- Selects all the nodes that precede the origin, and that are children of the same parent

- For document, attribute and namespace nodes, this axis is empty

Axes

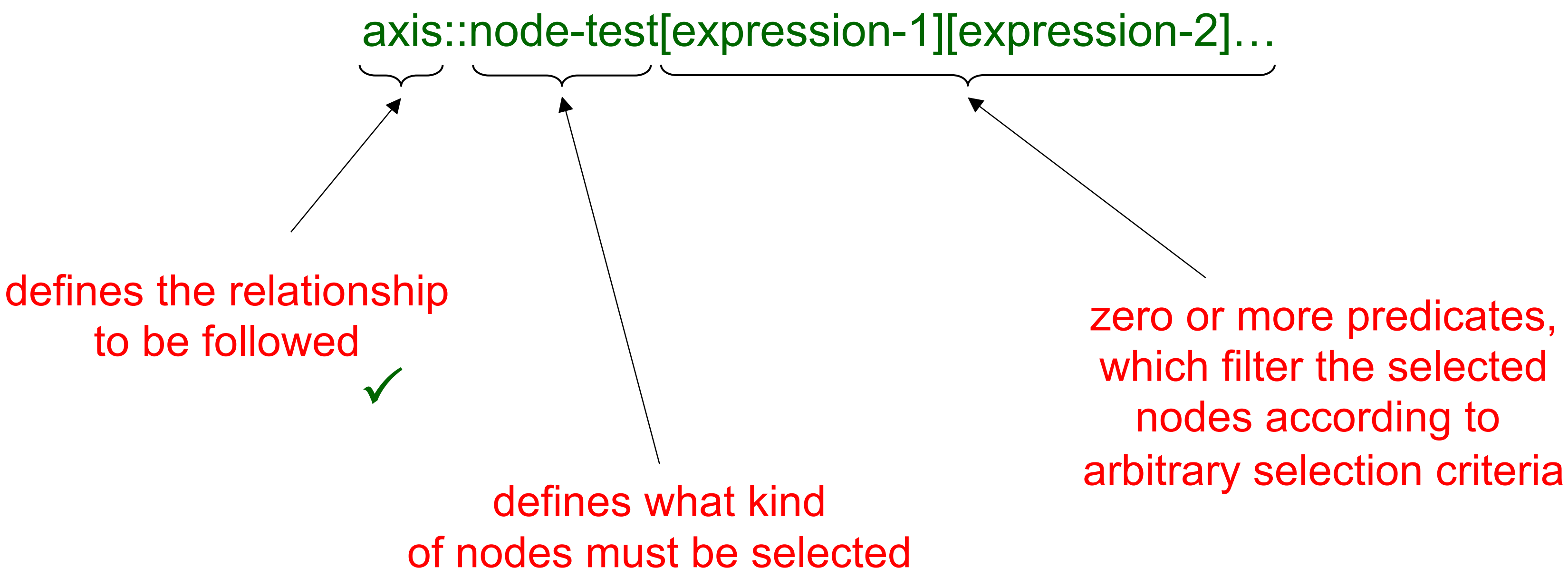
- XPath defines 13 axes:
    - ancestor
    - ancestor-or-self
    - attribute
    - child
    - descendant
    - descendant-or-self
    - following
    - following-sibling
    - namespace
    - parent
    - preceding
    - preceding-sibling
    - <span style="color:red">**self**</span>



- Selects the origin node
- This axis is always non-empty
- Usually, this axis is used in a node-test in order to test whether the current node pass that node-test

Location Paths

- XPath uses location paths to select nodes in a tree
- A location path is a series of location steps separated by the symbol /
- Each location step has the form



axis::node-test[expression-1][expression-2]…

defines the relationship to be followed ✓

defines what kind of nodes must be selected

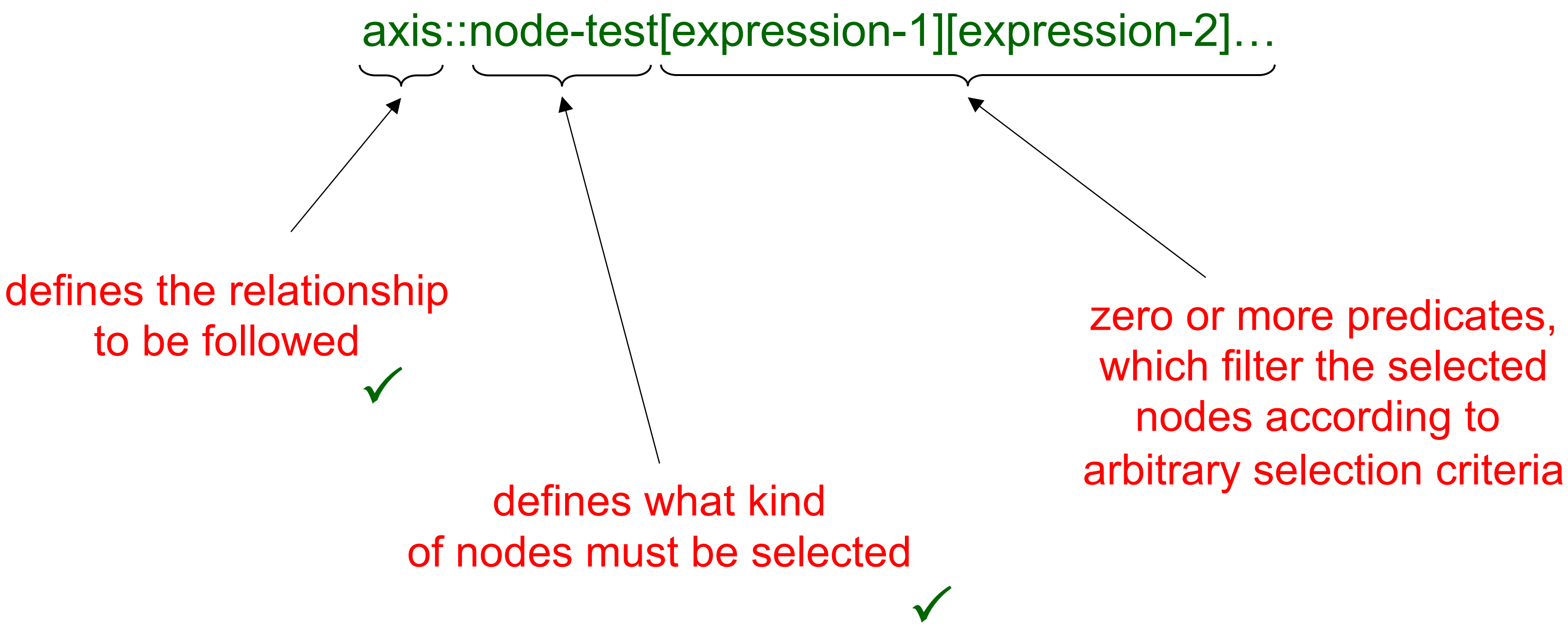
zero or more predicates, which filter the selected nodes according to arbitrary selection criteria

Node Test

node()	selects all nodes
text()	selects only text nodes
name	selects only elements nodes with tag “name” (child::*name*) …but, if it is used with the attribute axis (attribute::*name*), then it selects the “*name*” attribute nodes …and if it is used with the namespace axis (namespace::*name*), then is selects the namespace nodes with prefix “*name*”
*	selects all element nodes (child::*) …but, if it is used with the attribute axis (attribute::*), then it selects all the attribute nodes …and if it is used with the namespace axis (namespace::*), then it selects all the namespace nodes

Location Paths

- XPath uses location paths to select nodes in a tree
- A location path is a series of location steps separated by the symbol /
- Each location step has the form



axis::node-test[expression-1][expression-2]…

defines the relationship to be followed ✓

defines what kind of nodes must be selected ✓

zero or more predicates, which filter the selected nodes according to arbitrary selection criteria

Predicates

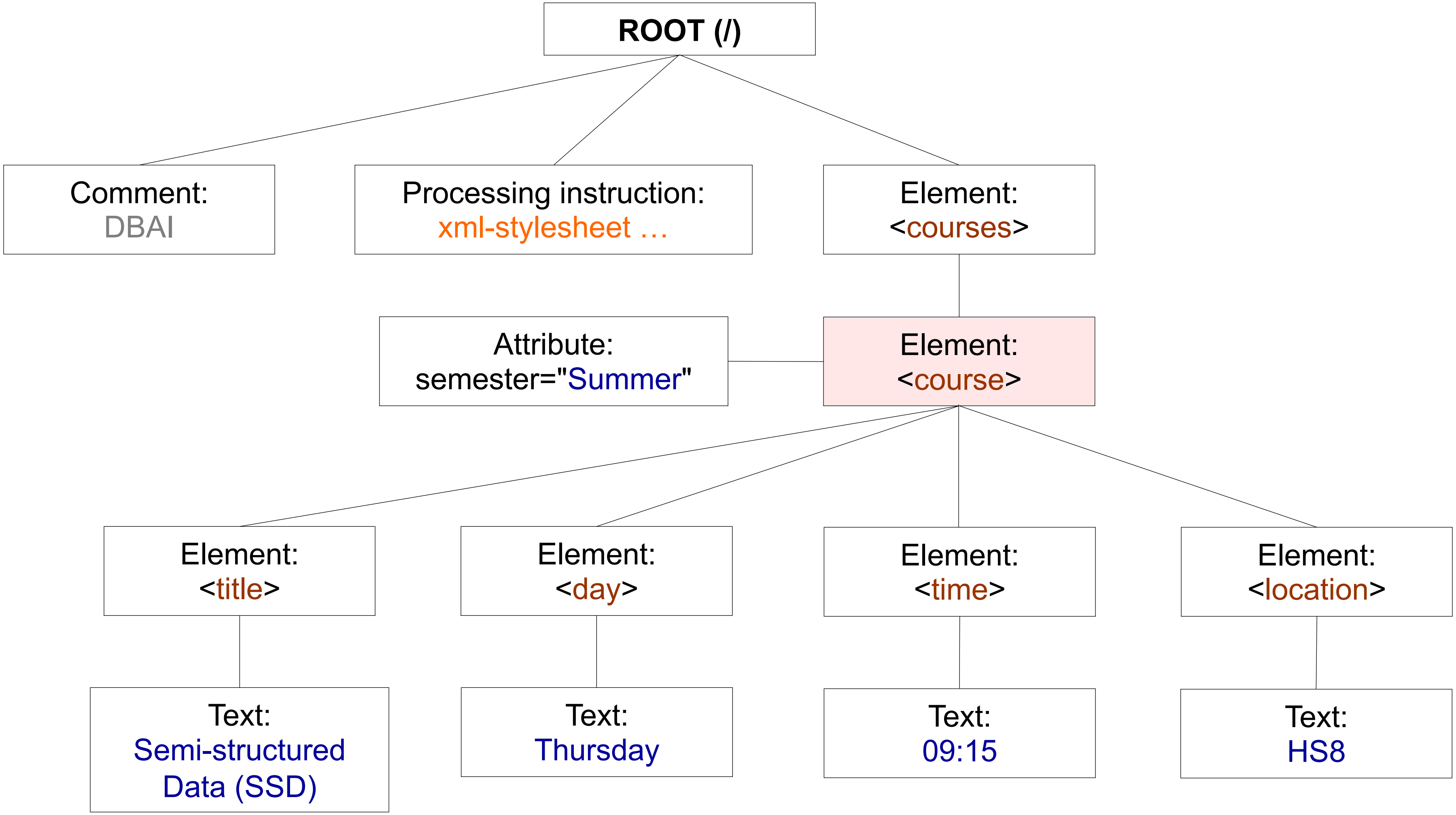
- Evaluating *axis*::*node-test* alone results in an initial list of nodes

- The predicates *[expression-1], [expression-2]*, …are then applied as "filters":
  - first, kick out all nodes that do not satisfy *expression-1*
  - second, kick out all nodes that do not satisfy *expression-2*
  - ….

- Each "expression-*n*" is qualifying expression: a node needs to satisfy the expression in order to be kept for further consideration

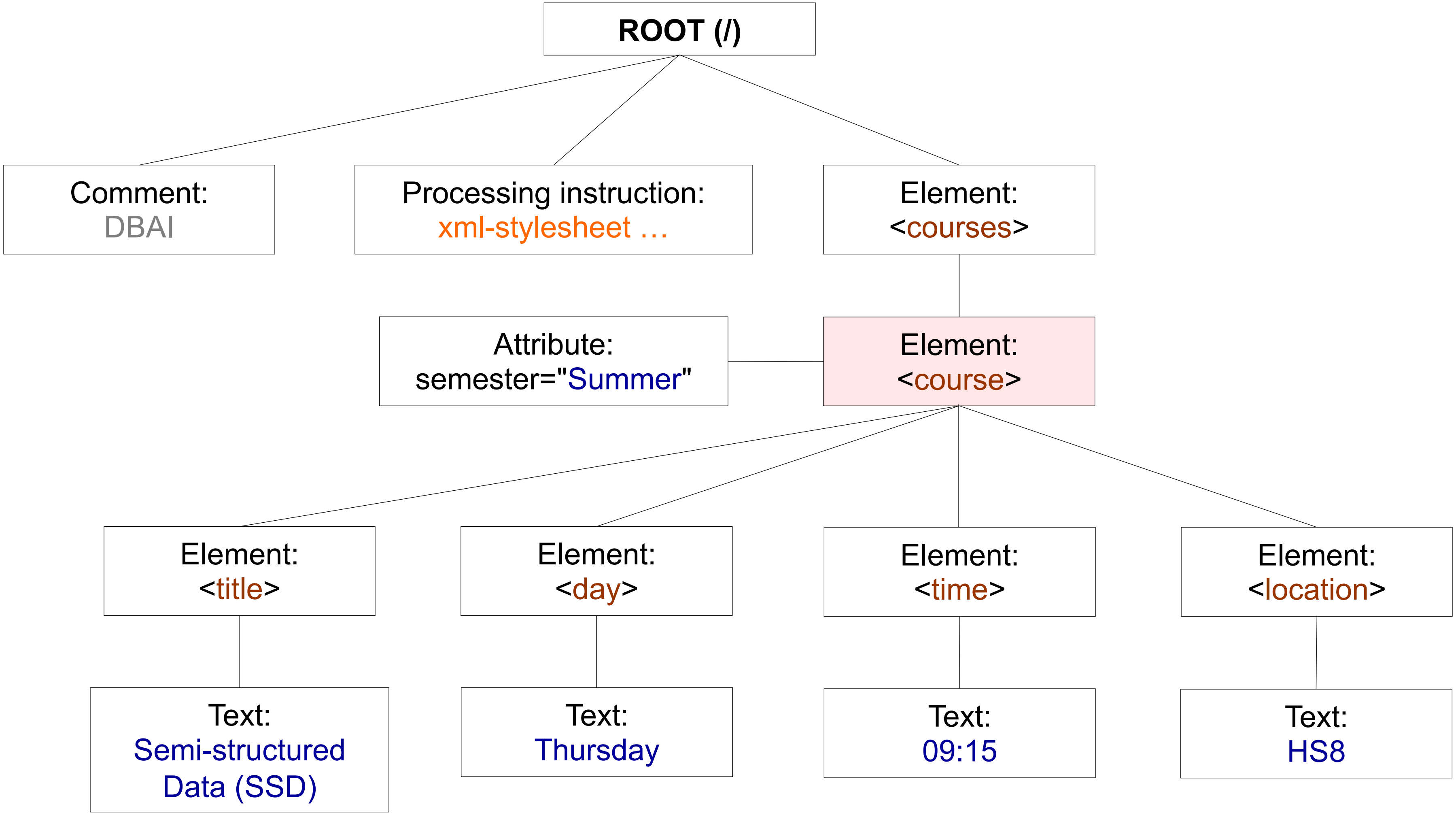
- Each "expression-*n*" may be any XPath expression (not limited to location paths)

Predicates: Examples



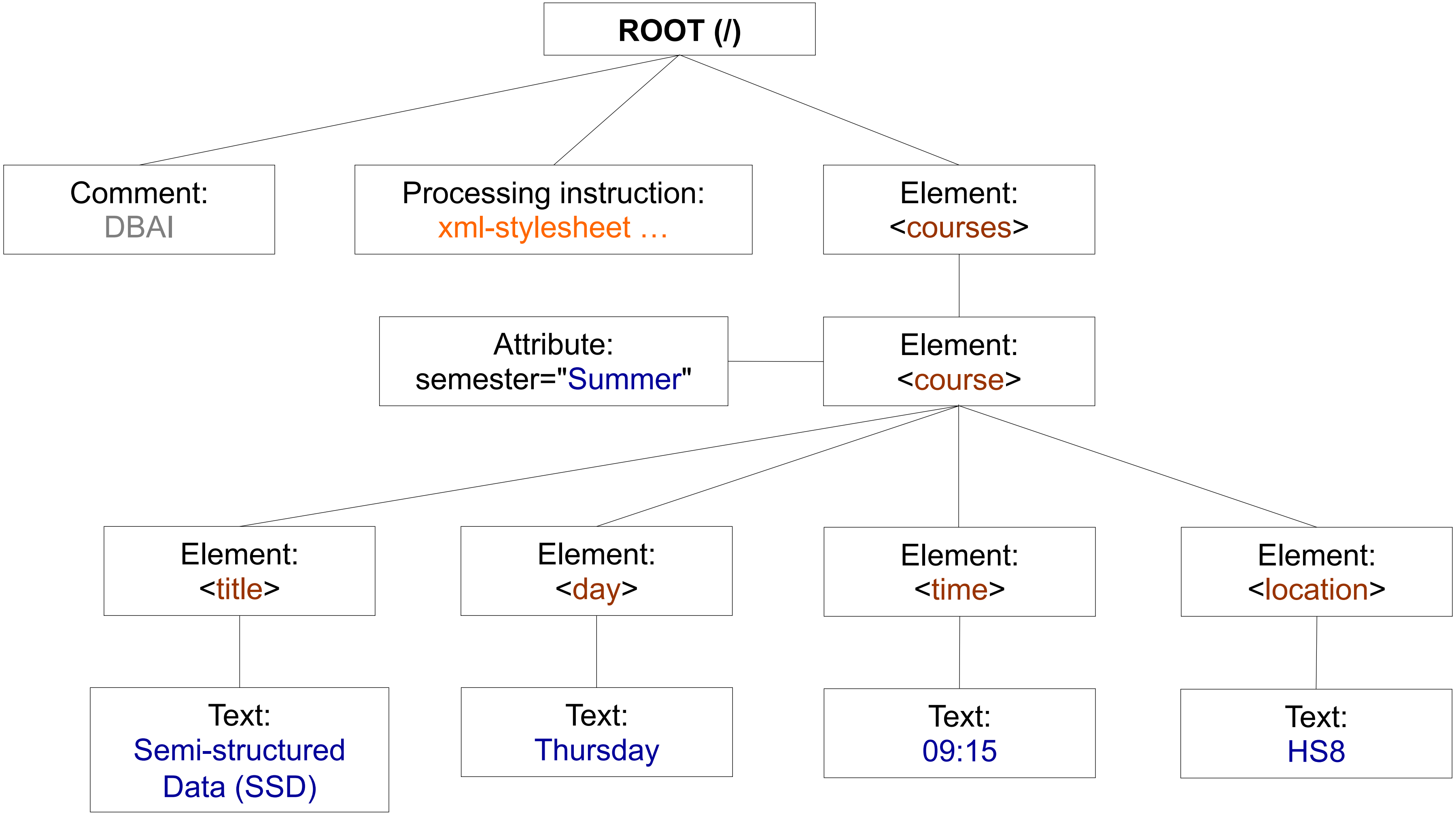
/child::courses/child::course[position() = 1]

Predicates: Examples



/child::courses/child::course[position() = last()]

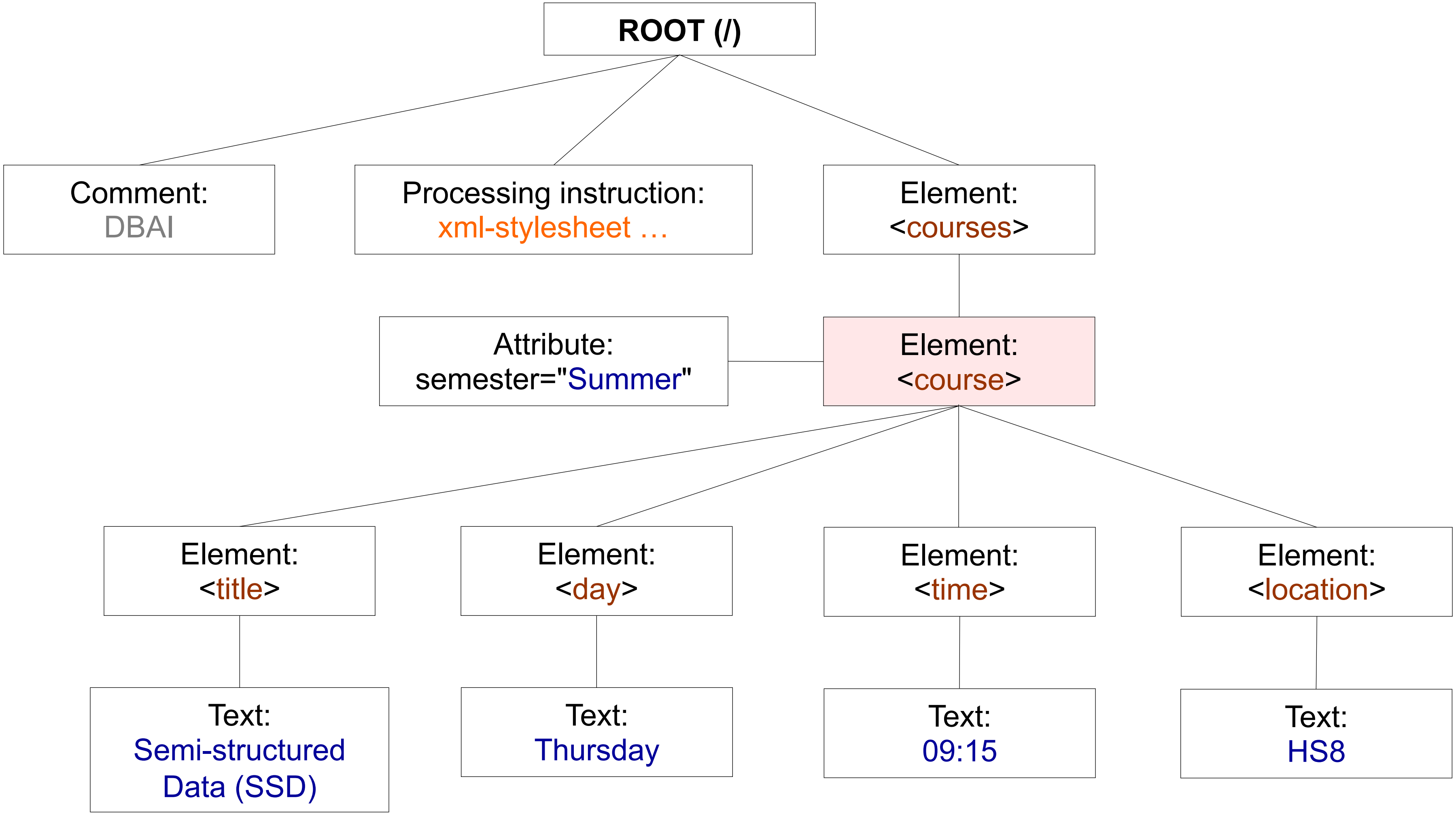
Predicates: Examples



empty!!!

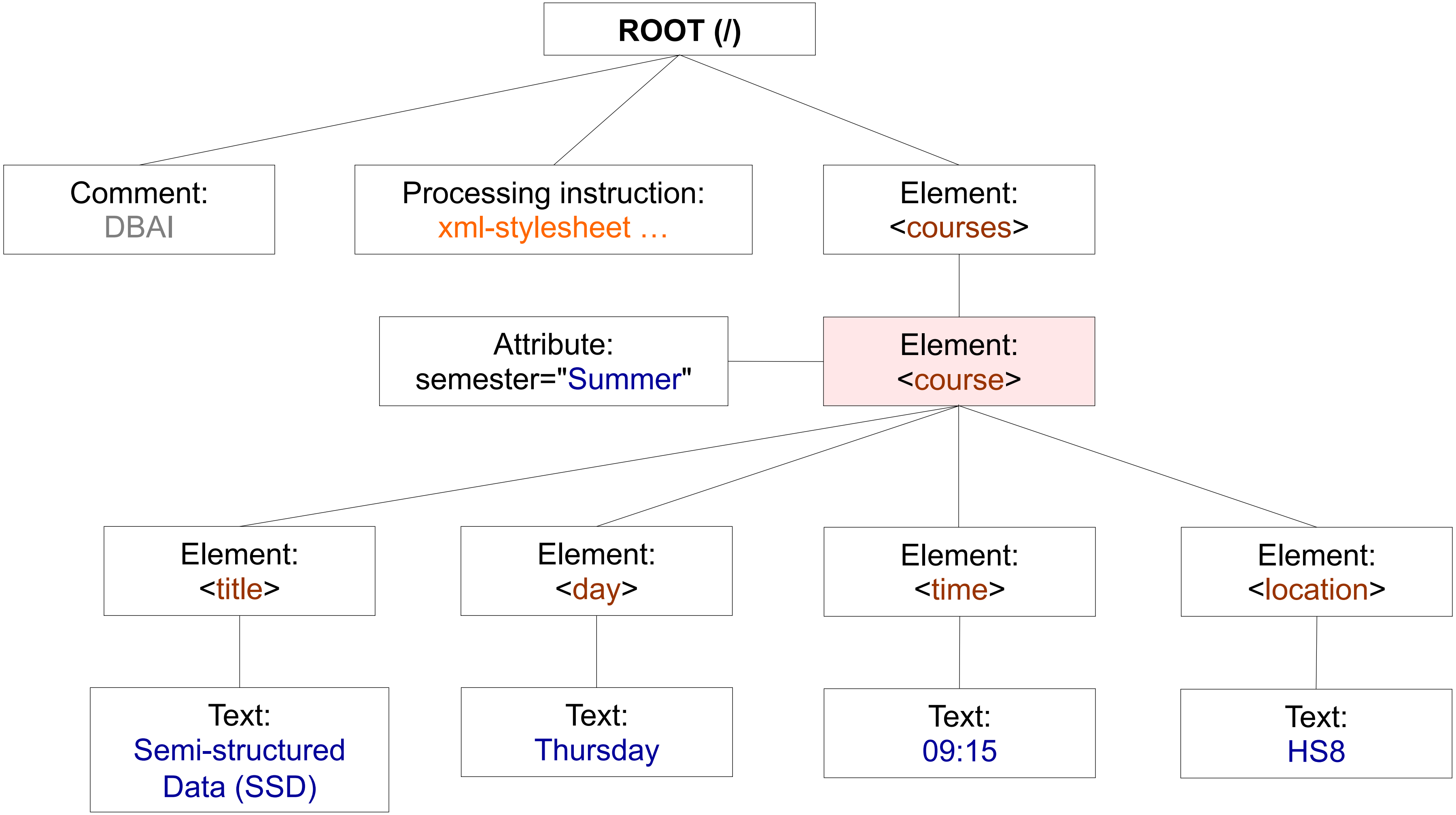
/child::courses/child::course[position() = last()-1]

Predicates: Examples



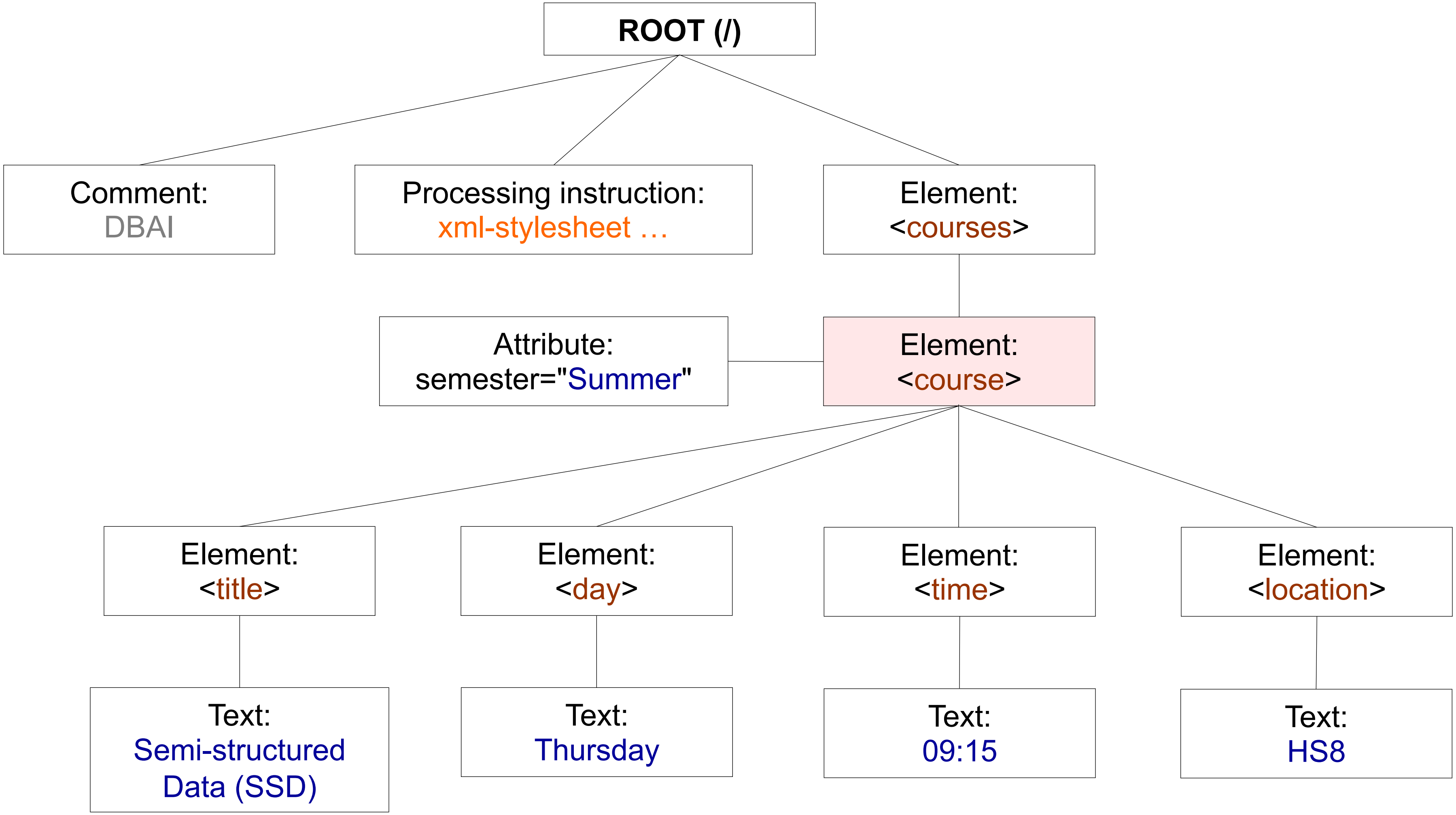
/child::courses/child::course[position() < 3]

Predicates: Examples



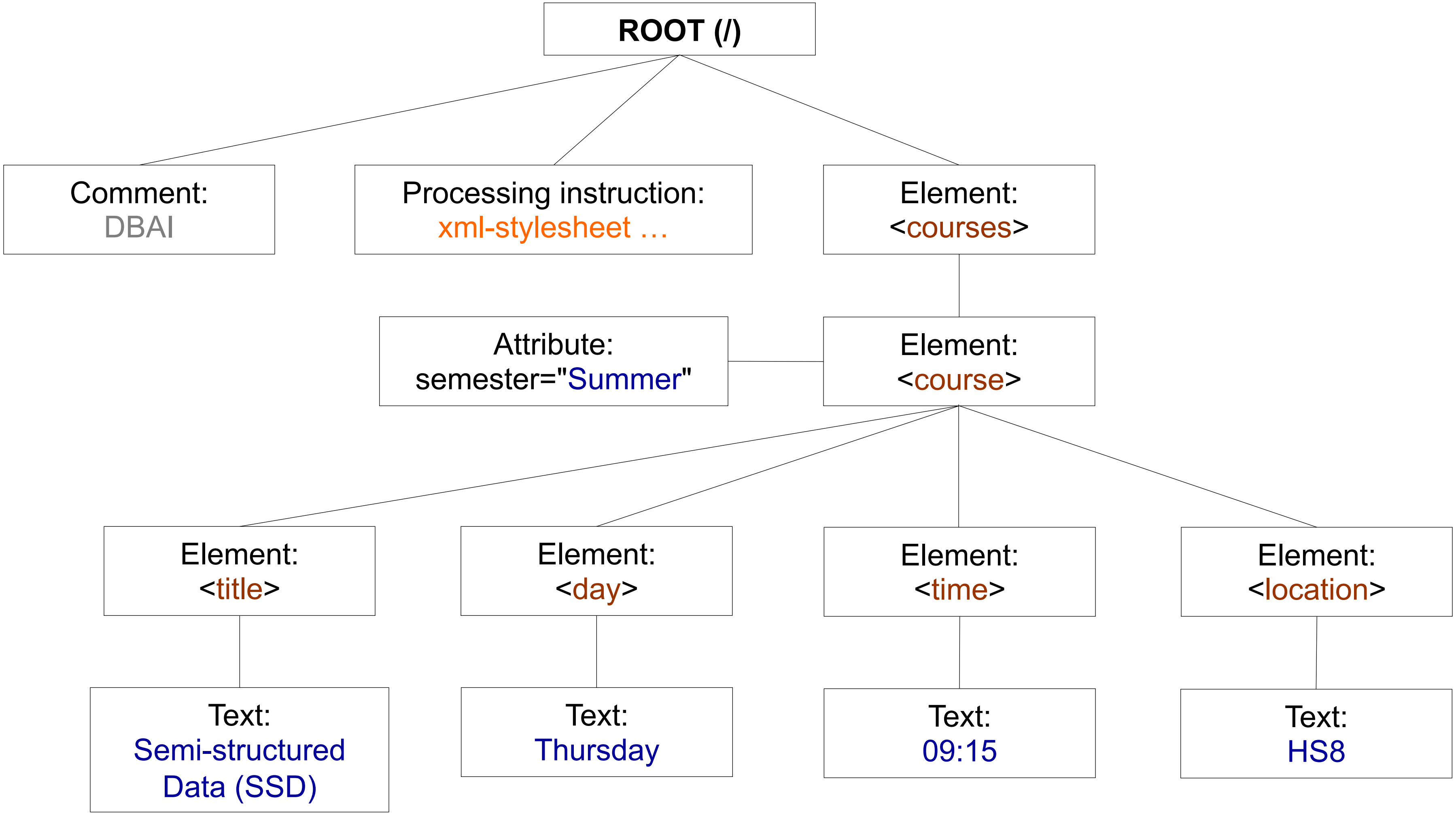
/child::courses/child::course[attribute::semester]

Predicates: Examples



/child::courses/child::course[attribute::semester = "Summer"]

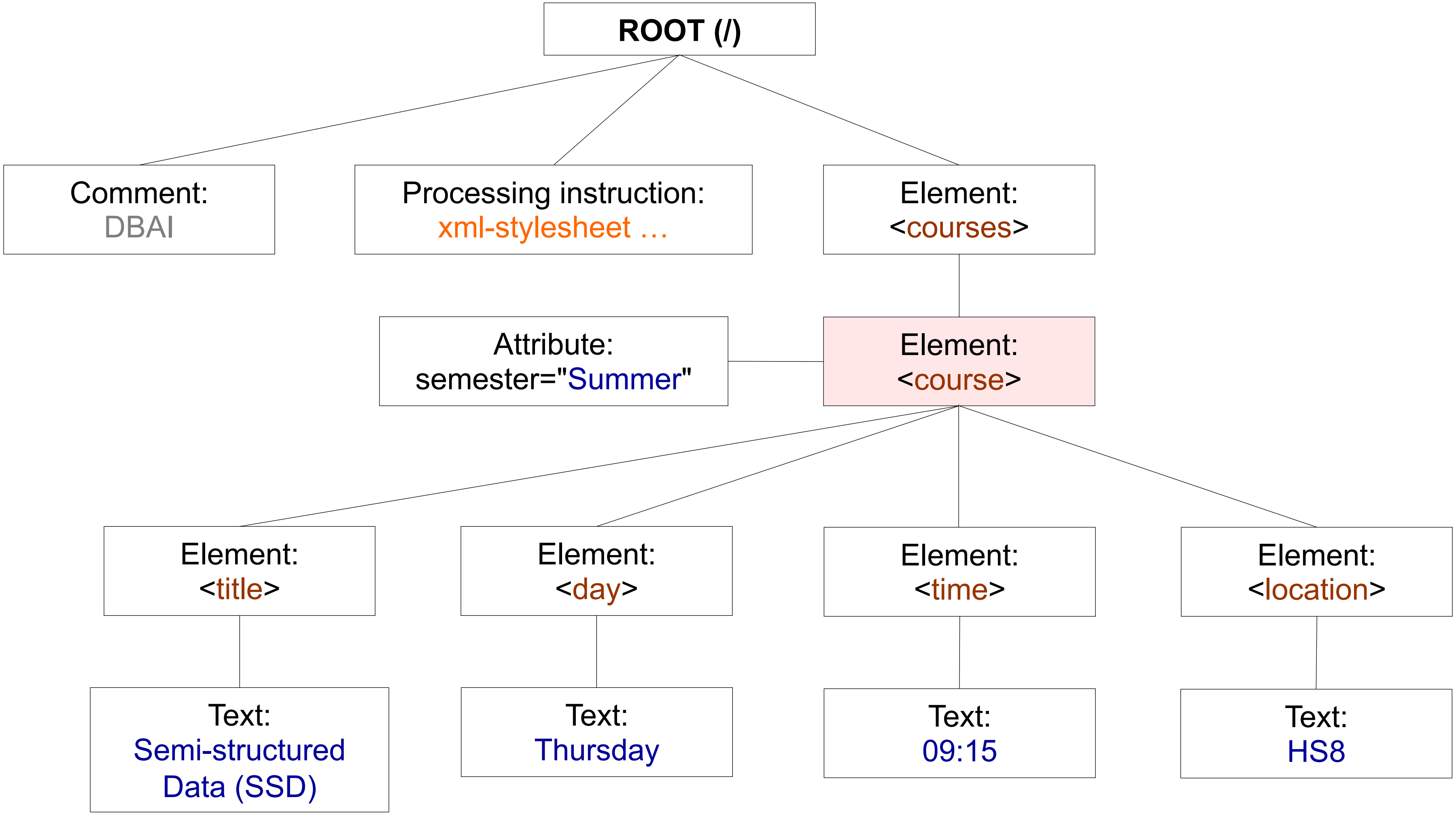
Predicates: Examples



empty!!!

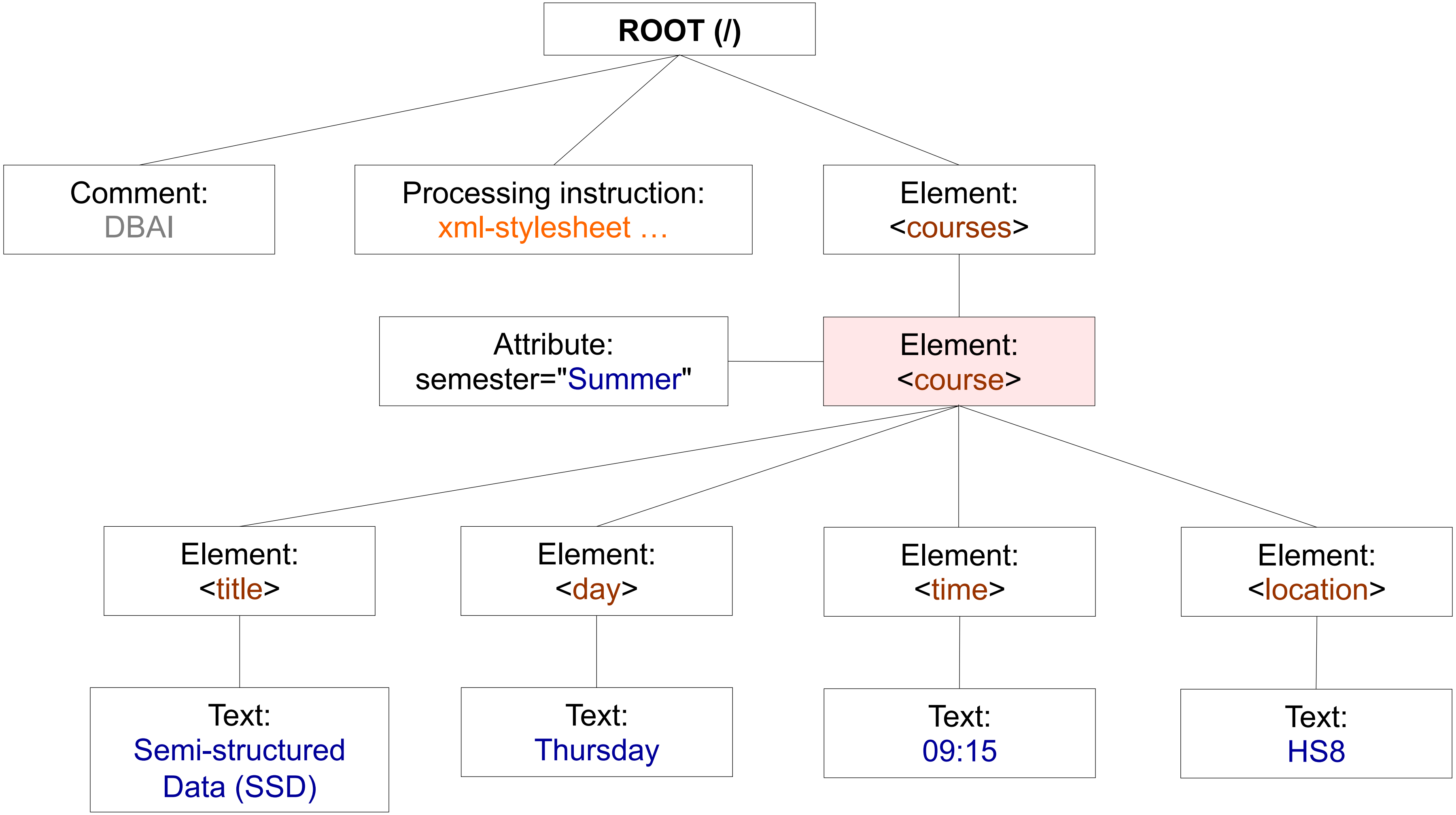
/child::courses/child::course[attribute::semester = “Winter"]

Predicates: Examples



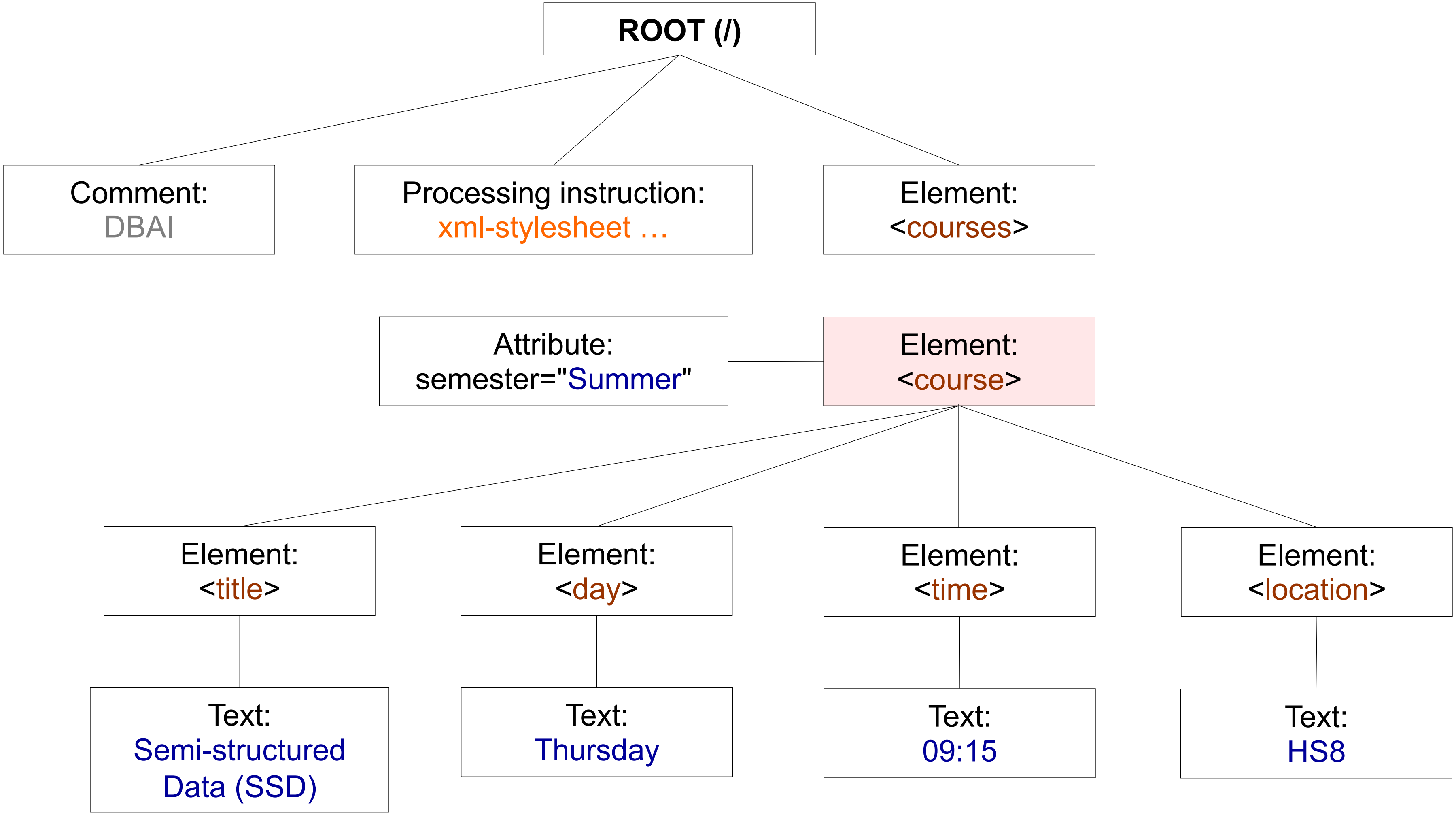
/child::courses/child::course[position() = 1][attribute::semester = "Summer"]

Predicates: Examples



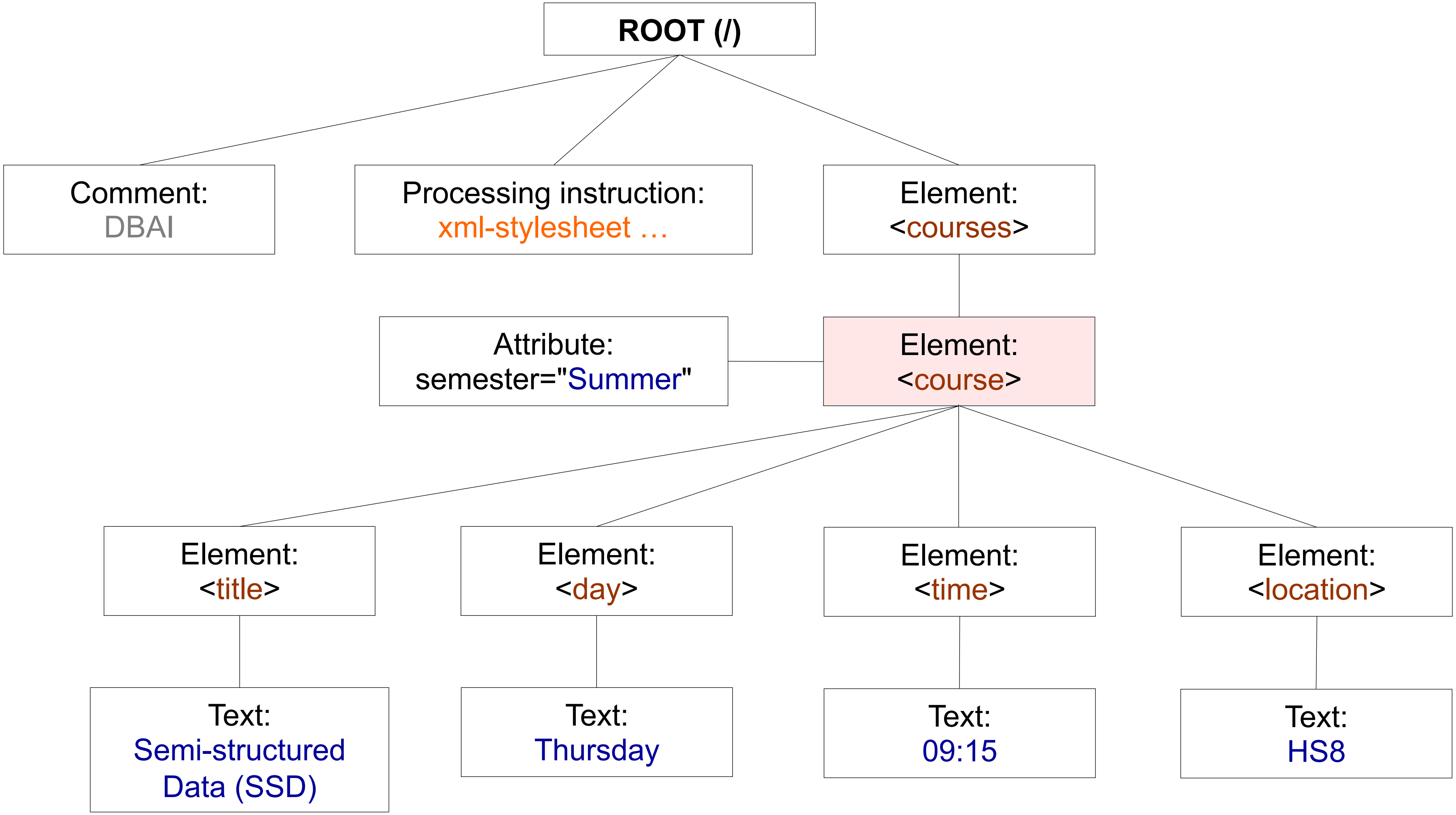
/child::courses/child::course[attribute::*]

Predicates: Examples



/child::courses/child::course[child::day = "Thursday"]

Predicates: Examples



/child::courses/child::course[child::day = “Monday“ or child::day = "Thursday"]

General XPath Expressions

- Location Paths are central subset of XPath and return node-sets
- General Xpath expressions can also return numbers, Booleans and strings

- Data-Types:
  - Numbers
  - Strings
  - Booleans
  - Node-Sets

XPath Operators

Operator	Description	Example
|	Union of two node-sets	/child::A | /child::B
+	Addition	6 + 4
-	Subtraction	6 - 4
*	Multiplication	6 * 4
div	Division	8 div 4
mod	Modulus (division remainder)	5 mod 2
=	Equal	A = 9.80
!=	Not equal	A != 9.80
<	Less than	A < 9.80
<=	Less than or equal to	A <= 9.80
>	Greater than	A > 9.80
>=	Greater than or equal to	A >= 9.80
or	Logical OR	A = 9.80 or A = 9.70
and	Logical AND	A > 9.00 and A < 9.90

XPath Functions

- Node-Set Functions

  count(/descendant-or-self::node()/course)

- String Functions

  starts-with(“Richard”,”Ric”)

- Boolean Functions

  not(attribute::age!=42)

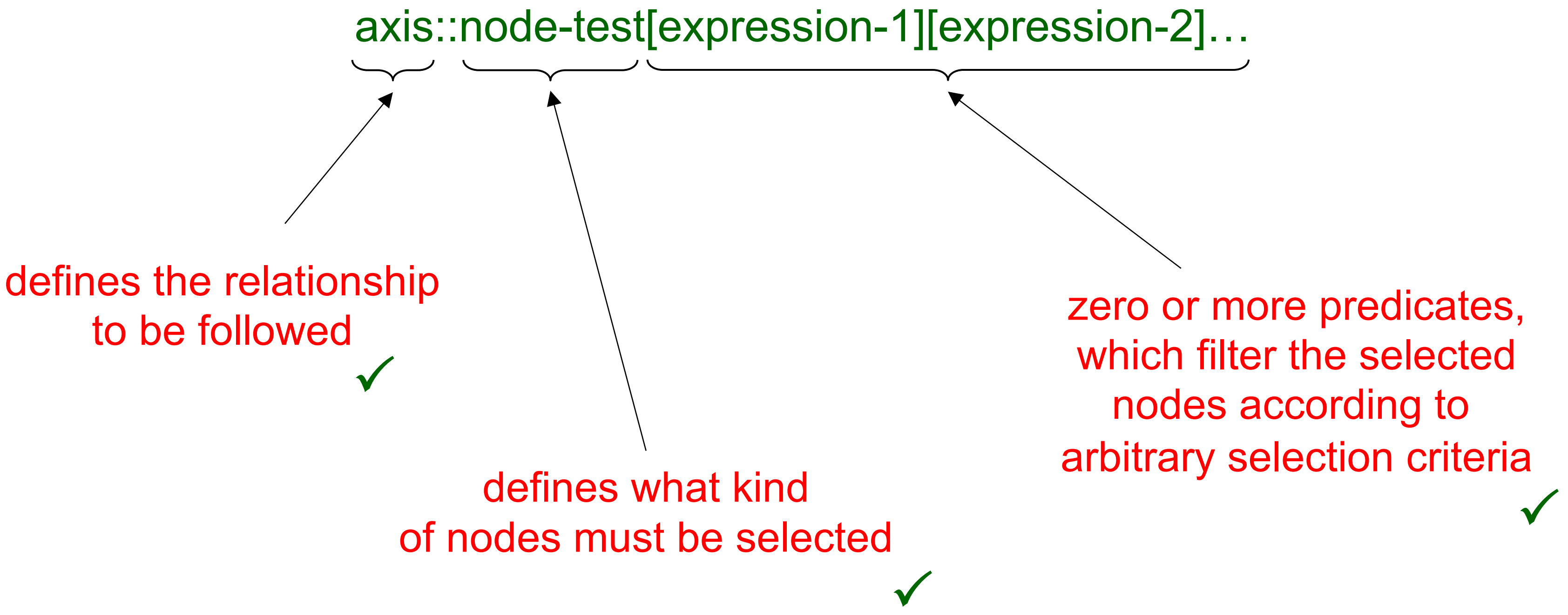
- Number Functions

  floor(attribute::temperature)

We will see them in action later on (and more of them)

Location Paths

- XPath uses location paths to select nodes in a tree
- A location path is a series of location steps separated by the symbol /
- Each location step has the form



axis::node-test[expression-1][expression-2]…

defines the relationship
to be followed
✓

defines what kind
of nodes must be selected
✓

zero or more predicates,
which filter the selected
nodes according to
arbitrary selection criteria
✓

Up to Now

- **XPath Terminology**
- **XPath at First Glance**
- **Location Paths (Axis, Node Test, Predicate)**
- Abbreviated Syntax

Abbreviated Syntax

- The most commonly used location steps can be in an abbreviated syntax
- Simplify XPath expressions

/descendant-or-self::node()/	//
self::node()	.
parent::node()	..
child::	
attribute::	@
position() = n	n

Abbreviated Syntax: Examples

/child::courses/child::course[position() = 1]

/courses/child::course[position() = 1]

/courses/course[position() = 1]

/courses/course[1]

Abbreviated Syntax: Examples

/child::courses/child::course[attribute::semester]

/courses/child::course[attribute::semester]

/courses/course[attribute::semester]

/courses/course[@semester]

Abbreviated Syntax: Examples

/child::courses/child::course[position() = 1][attribute::semester = "Summer"]

/courses/child::course[position() = 1][attribute::semester = "Summer"]

/courses/course[position() = 1][attribute::semester = "Summer"]

/courses/course[1][attribute::semester = "Summer"]

/courses/course[1][@semester = "Summer"]

Abbreviated Syntax: Examples

/descendant-or-self::node()/child::course[position() = 1][attribute::semester = "Summer"]

//child::course[position() = 1][attribute::semester = "Summer"]

//course[position() = 1][attribute::semester = "Summer"]

//course[1][attribute::semester = "Summer"]

//course[1][@semester = "Summer"]

Sum Up

- XPath Terminology
- XPath at First Glance
- Location Paths (Axis, Node Test, Predicate)
- Abbreviated Syntax

Tools

- Web-based Tools:
  - PathEnq: http://www.qutoric.com/xslt/analyser/xpathtool.html
  - xPath tester: http://www.xpathtester.com/xpath
- Example document: in the TUWEL course