A Unified Format for Language Documents

Vadim Zaytsev and Ralf Lämmel
Software Languages Team
Universität Koblenz-Landau

Motivation

★ Formal languages are defined by formal grammars

★ Real languages are defined by their specifications

★ A standard is a complex artifact

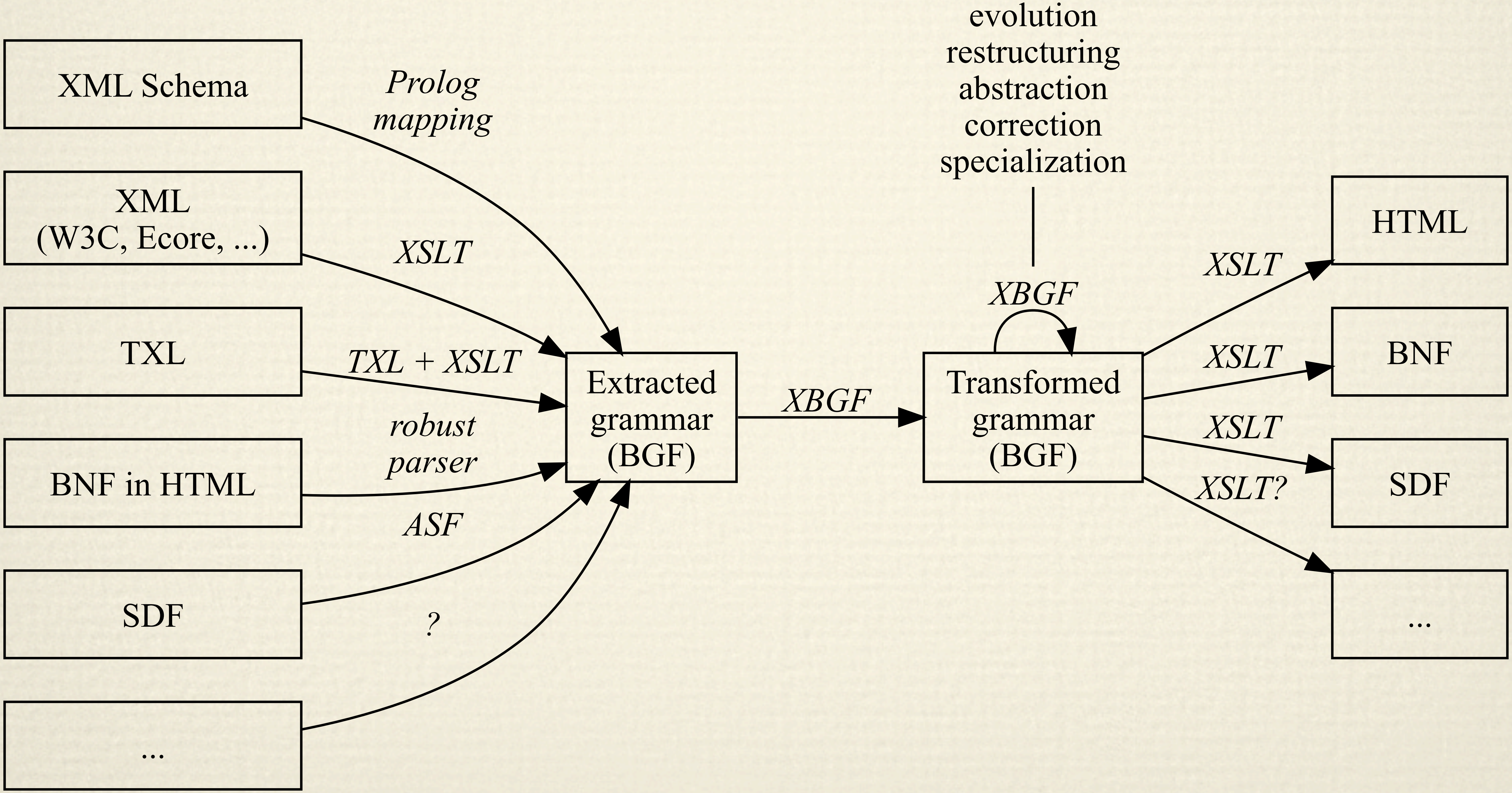
★ Adequate supporting technology is needed

★ Co-evolution of documents and grammarware tools

★ “Literate programming”

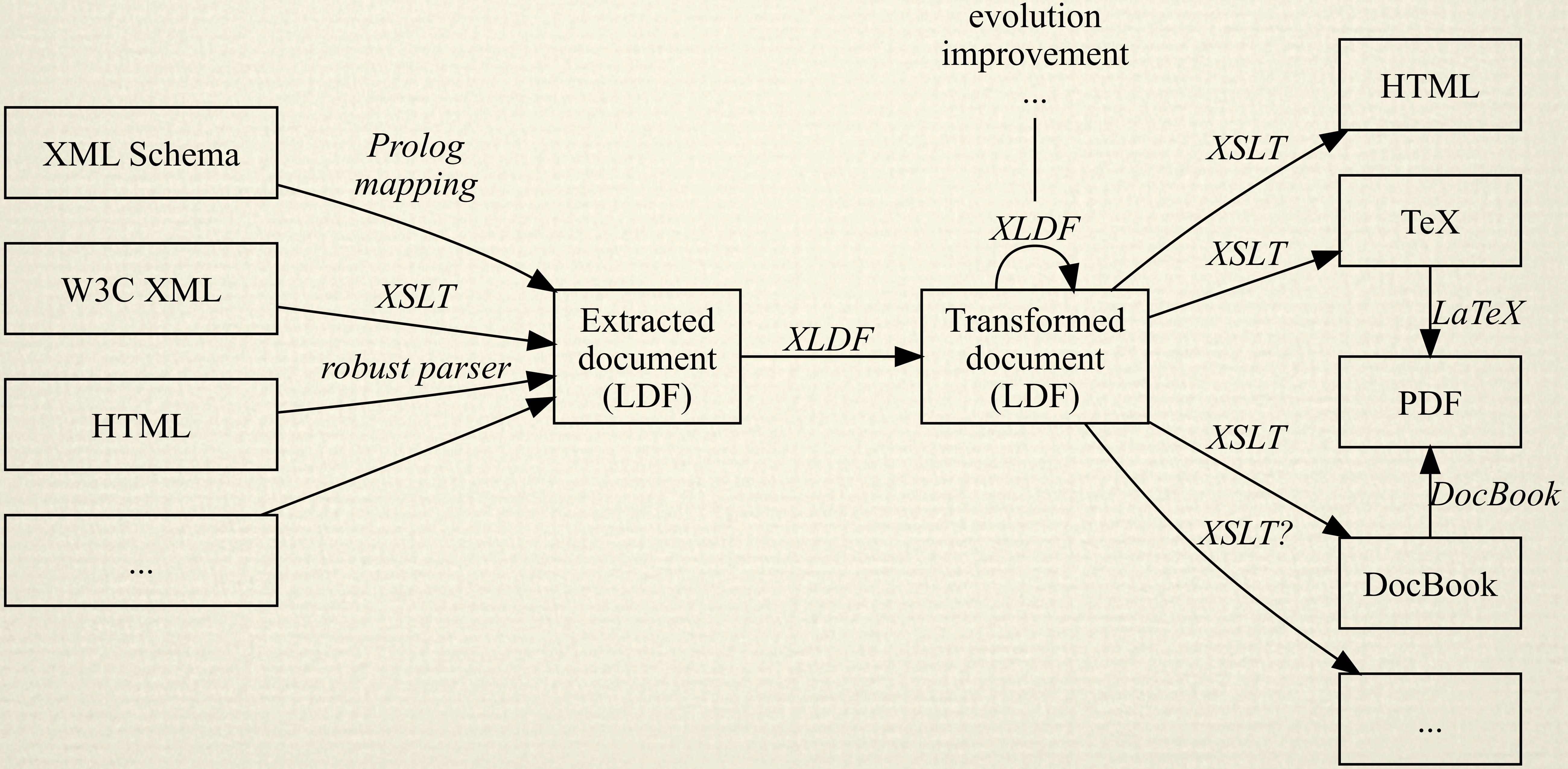
Language document engineering

Focus on grammars



Language document engineering

Focus on documents

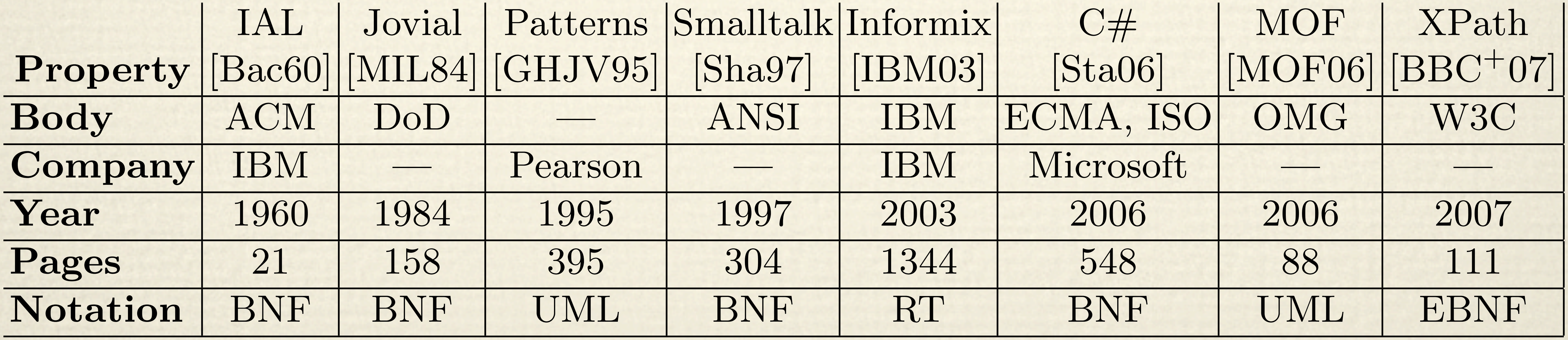


Language standardization

- American National Standards Institute (ANSI, 1918)
- European Computer Manufacturers Association (ECMA, 1961)
- Institute of Electrical and Electronics Engineers Standards Association (IEEE, 1884)
- International Electrotechnical Commission (IEC, 1906)
- International Organization for Standardization (ISO, 1947)
- International Telecommunication Union (ITU, 1865)
- Internet Engineering Task Force (IETF, 1986)
- Object Management Group (OMG, 1989)
- Organization for Advancement of Structured Information Standards (OASIS, 1993)
- Website Standards Association (WSA, 2006)
- World Wide Web Consortium (W3C, 1994)

Control group (for the paper)

- ★ IAL (Algol 58)
- ★ JOVIAL
- ★ Design Patterns
- ★ ANSI Smalltalk
- ★ IBM Informix
- ★ ISO C$^{\sharp}$
- ★ OMG MOF
- ★ W3C XPath



Property	IAL [Bac60]	Jovial [MIL84]	Patterns [GHJV95]	Smalltalk [Sha97]	Informix [IBM03]	C# [Sta06]	MOF [MOF06]	XPath [BBC$^+$07]
Body	ACM	DoD	—	ANSI	IBM	ECMA, ISO	OMG	W3C
Company	IBM	—	Pearson	—	IBM	Microsoft	—	—
Year	1960	1984	1995	1997	2003	2006	2006	2007
Pages	21	158	395	304	1344	548	88	111
Notation	BNF	BNF	UML	BNF	RT	BNF	UML	EBNF

Language documentation

- Presentation
  - Adobe Framemaker, Microsoft Word, …
- Structure
  - DocBook, in-house XML schema
- Topic
  - DITA, Wikis

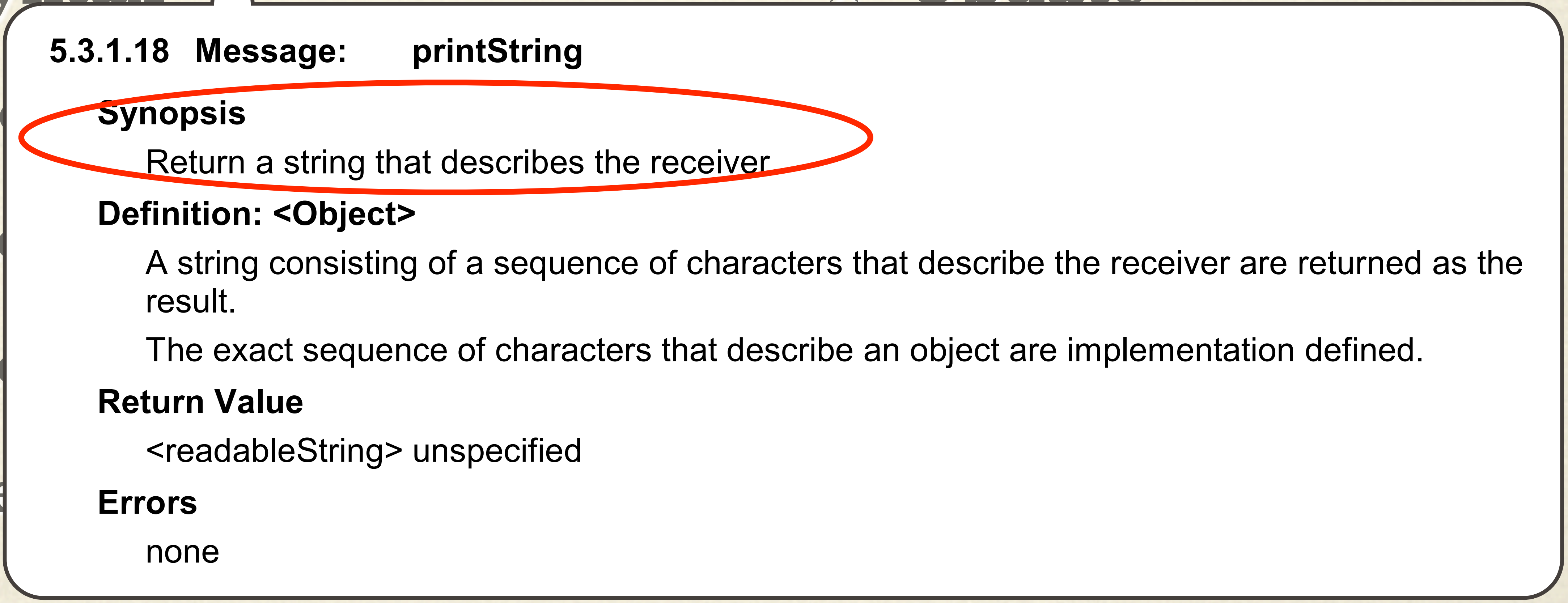
Domain concepts

- ★ Synopsis
- ★ Description
- ★ Syntax
- ★ Constraints
- ★ References
- ★ Relationship
- ★ Semantics
- ★ Rationale
- ★ Example
- ★ Update
- ★ Default
- ★ Value
- ★ List
- ★ Section & Subtopic

Domain concepts

- ★ Synopsis
- ★ Description
- ★ Syntax
- ★ C
- ★ R
- ★ R
- ★ Se

- ★ Rationale
- ★ Example
- ★ Update



5.3.1.18 Message: printString

Synopsis

Return a string that describes the receiver

Definition: <Object>

A string consisting of a sequence of characters that describe the receiver are returned as the result.

The exact sequence of characters that describe an object are implementation defined.

Return Value

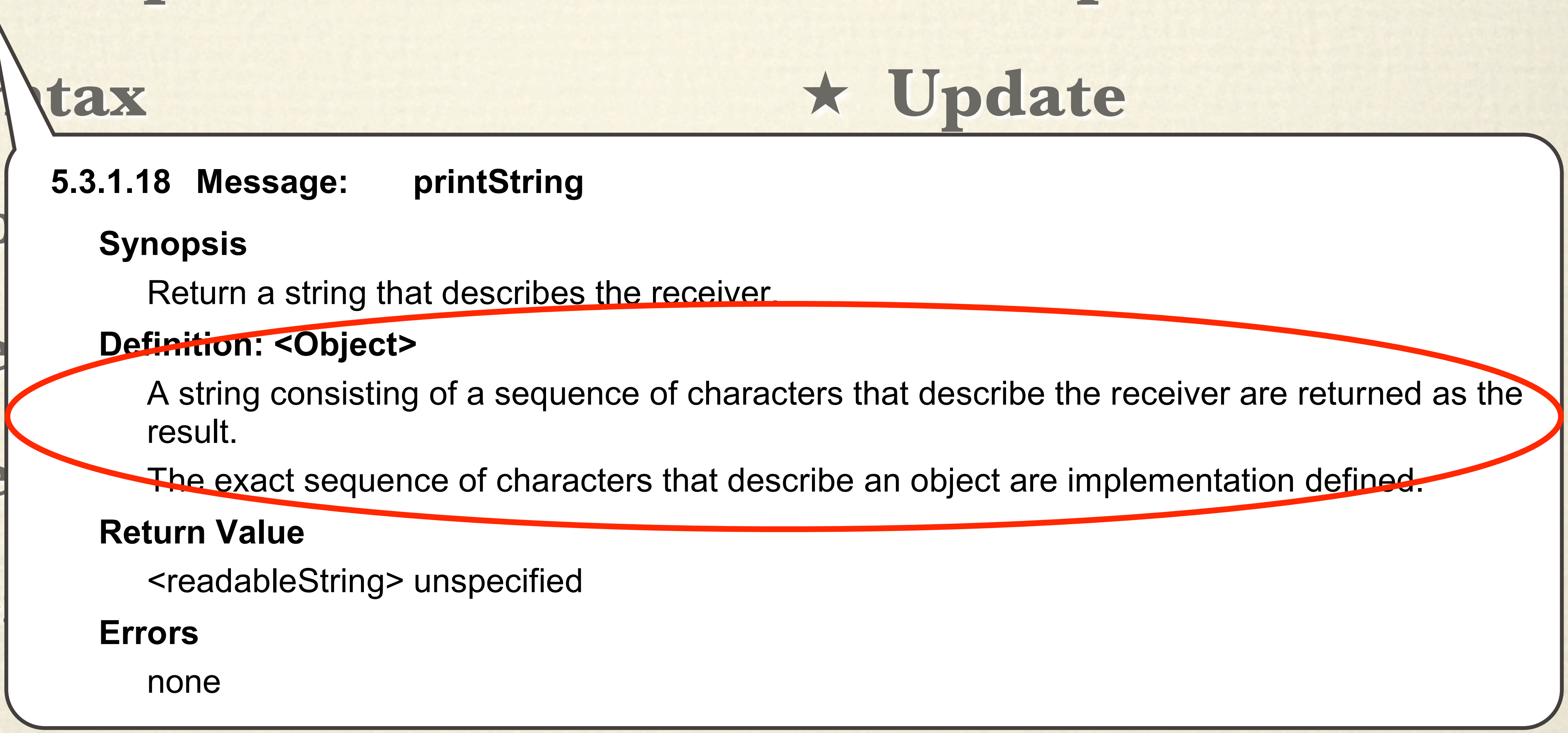
<readableString> unspecified

Errors

none

Domain concepts

- ★ Synopsis
- ★ Description
- ★ Syntax
- ★ Co
- ★ Re
- ★ Re
- ★ Se
- ★ Rationale
- ★ Example
- ★ Update



5.3.1.18 Message: printString

Synopsis

Return a string that describes the receiver.

Definition: <Object>

A string consisting of a sequence of characters that describe the receiver are returned as the result.

The exact sequence of characters that describe an object are implementation defined.

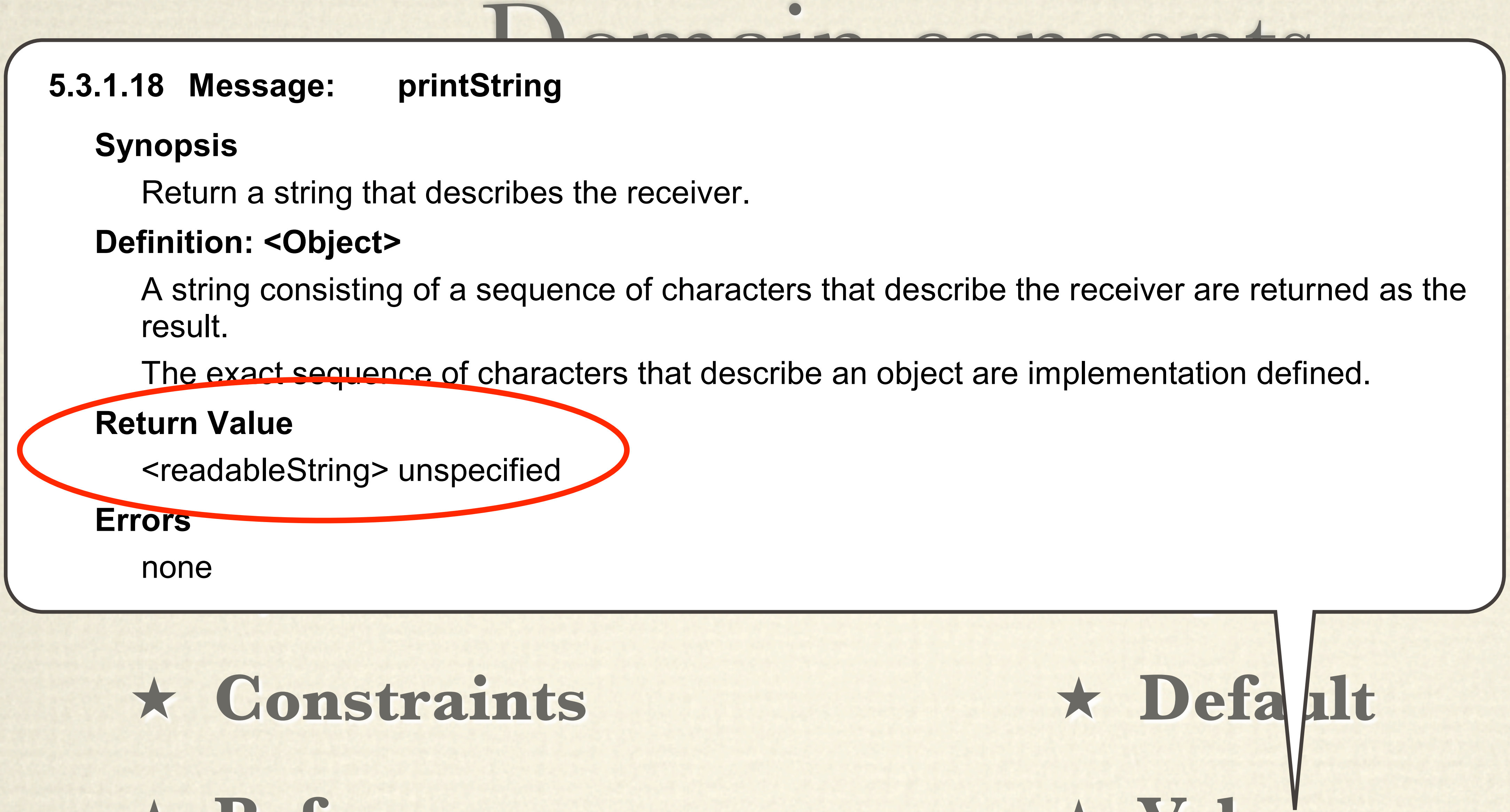
Return Value

<readableString> unspecified

Errors

none

Domain concepts



5.3.1.18 Message: printString

Synopsis

Return a string that describes the receiver.

Definition: <Object>

A string consisting of a sequence of characters that describe the receiver are returned as the result.

The exact sequence of characters that describe an object are implementation defined.

Return Value

<readableString> unspecified

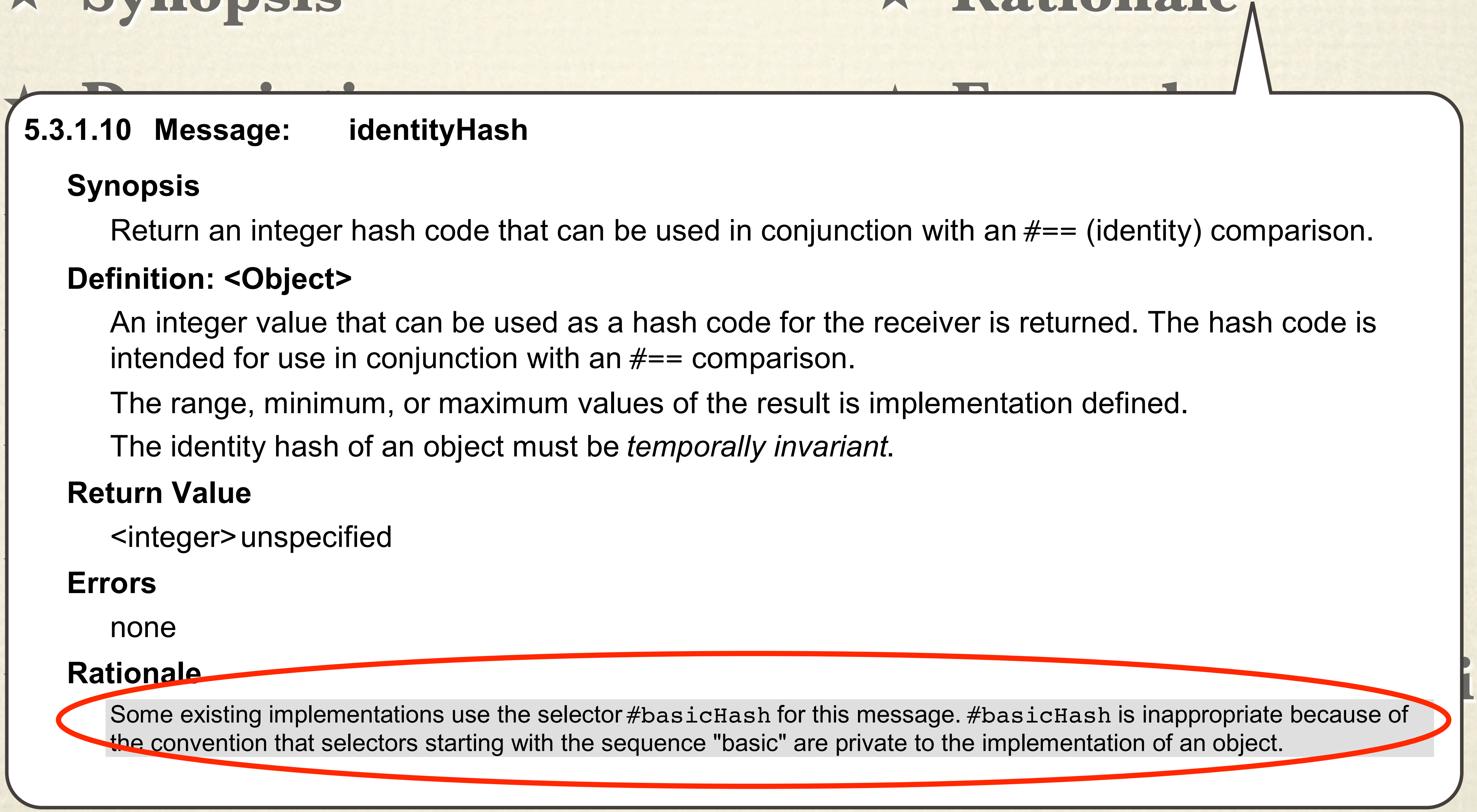
Errors

none

- ★ Constraints
- ★ References
- ★ Relationship
- ★ Semantics
- ★ Default
- ★ Value
- ★ List
- ★ Section & Subtopic

Domain concepts

- ★ Synopsis
- ★ Rationale



5.3.1.10 Message: identityHash

Synopsis

Return an integer hash code that can be used in conjunction with an #== (identity) comparison.

Definition: <Object>

An integer value that can be used as a hash code for the receiver is returned. The hash code is intended for use in conjunction with an #== comparison.

The range, minimum, or maximum values of the result is implementation defined.

The identity hash of an object must be *temporally invariant*.

Return Value

<integer> unspecified

Errors

none

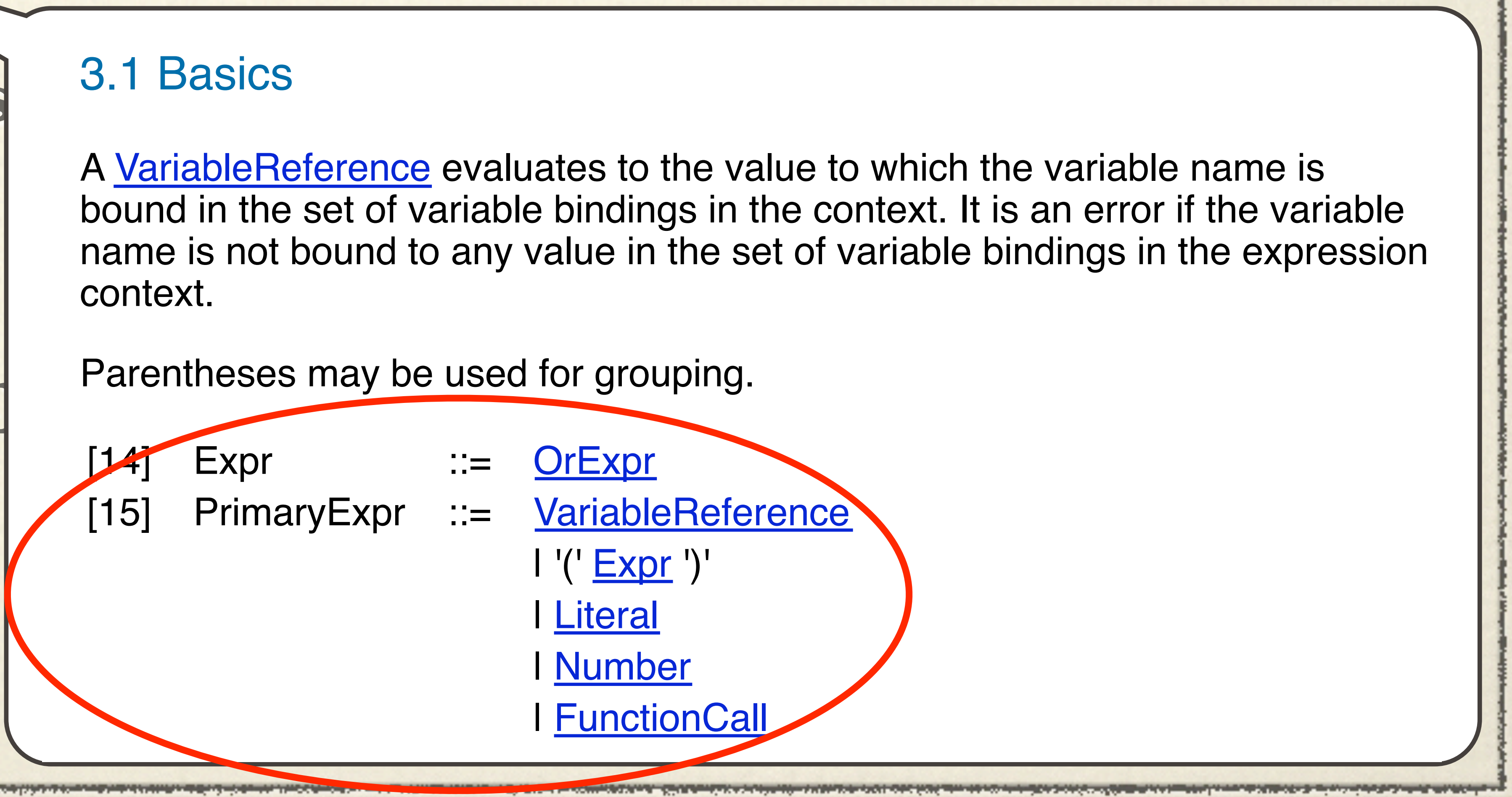
Rationale

Some existing implementations use the selector `#basicHash` for this message. `#basicHash` is inappropriate because of the convention that selectors starting with the sequence "basic" are private to the implementation of an object.

Domain concepts

- ★ Synopsis
- ★ Description
- ★ Syntax
- ★ Constraints
- ★ References
- ★ Relationshi
- ★ Semantics

- ★ Rationale
- ★ Example
- ★ Update

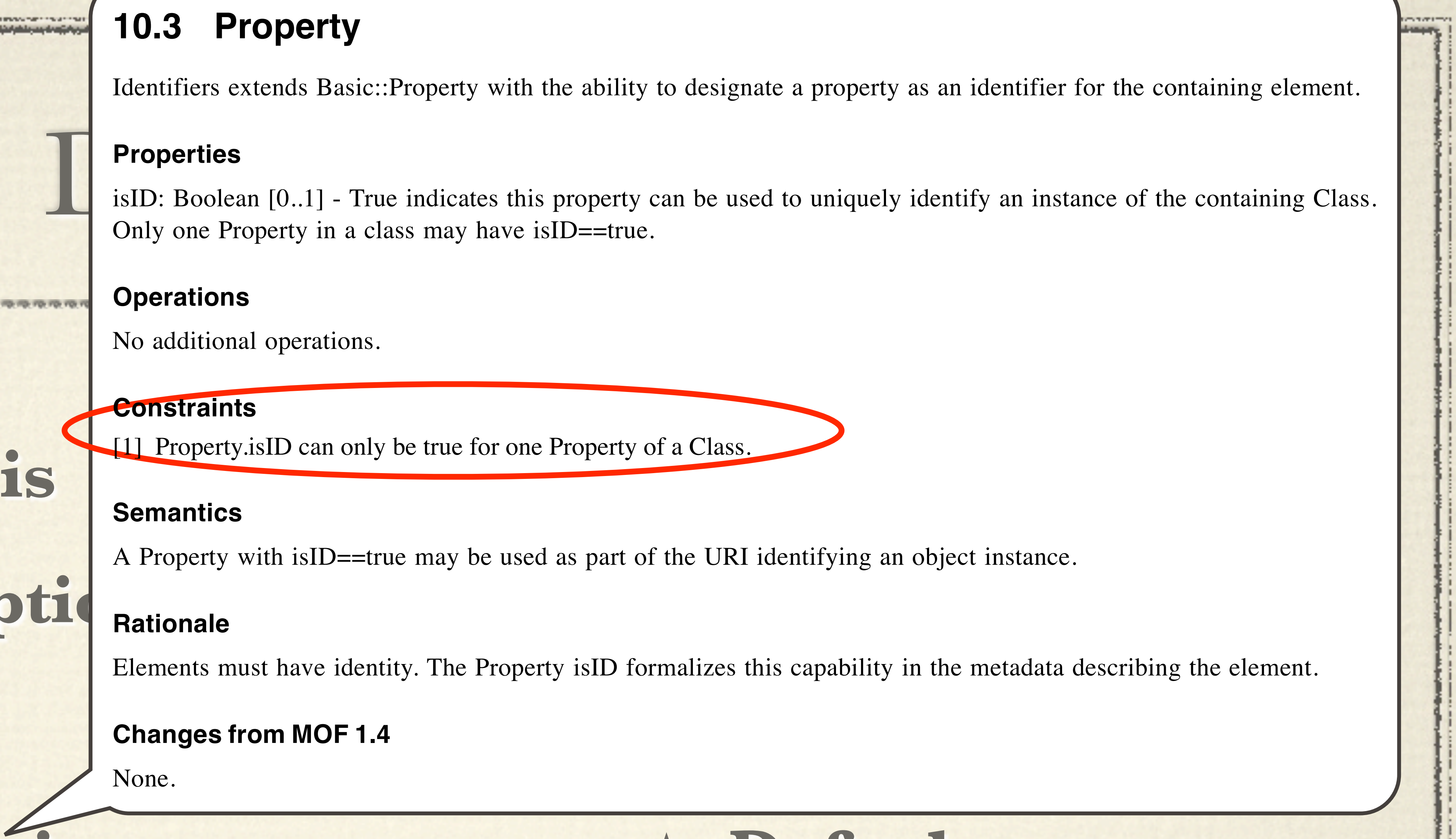


3.1 Basics

A VariableReference evaluates to the value to which the variable name is bound in the set of variable bindings in the context. It is an error if the variable name is not bound to any value in the set of variable bindings in the expression context.

Parentheses may be used for grouping.

```
[14]   Expr          ::=   OrExpr
[15]   PrimaryExpr   ::=   VariableReference
                           | '(' Expr ')'
                           | Literal
                           | Number
                           | FunctionCall
```



10.3 Property

Identifiers extends Basic::Property with the ability to designate a property as an identifier for the containing element.

Properties

isID: Boolean [0..1] - True indicates this property can be used to uniquely identify an instance of the containing Class. Only one Property in a class may have isID==true.

Operations

No additional operations.

Constraints

[1] Property.isID can only be true for one Property of a Class.

Semantics

A Property with isID==true may be used as part of the URI identifying an object instance.

Rationale

Elements must have identity. The Property isID formalizes this capability in the metadata describing the element.

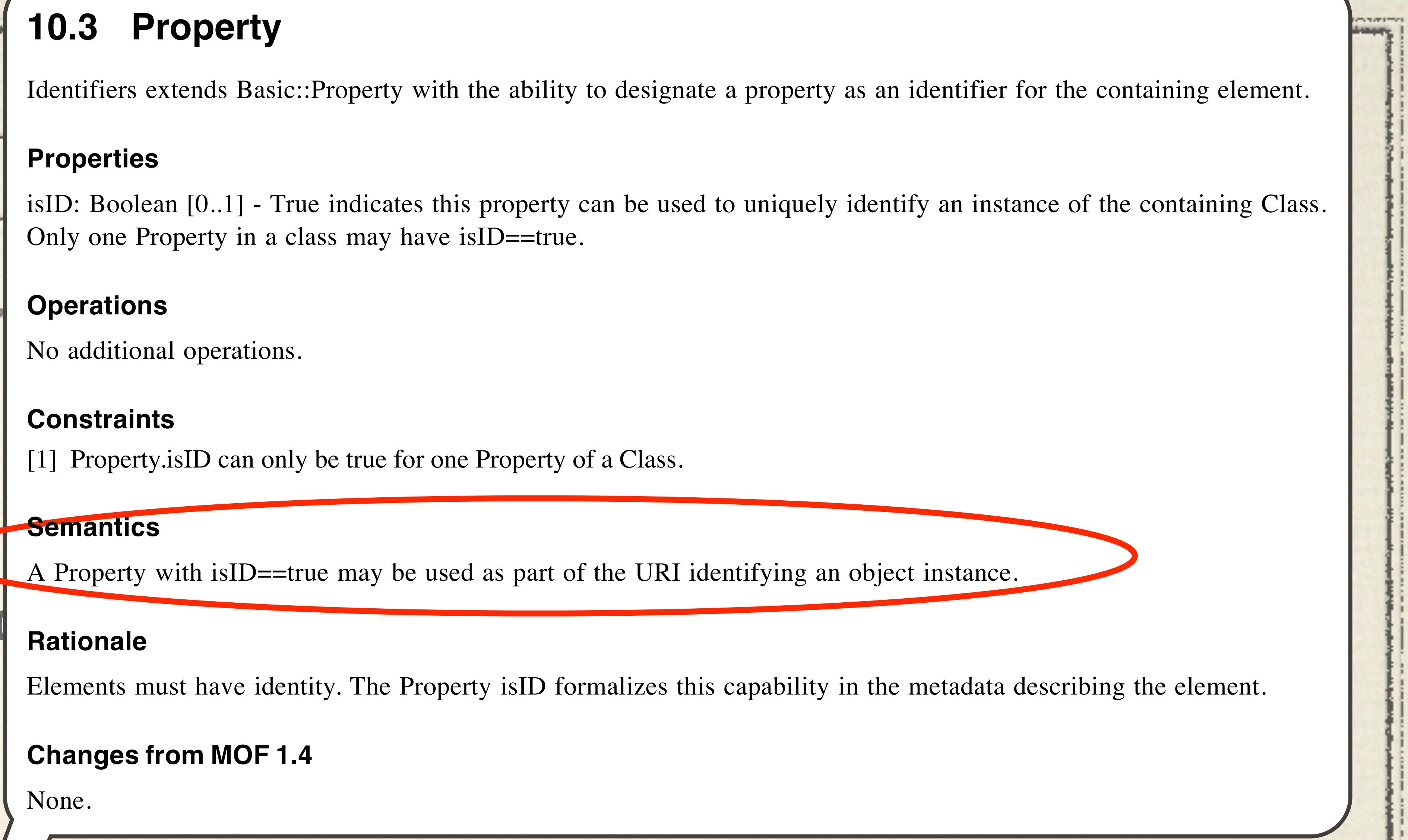
Changes from MOF 1.4

None.

- ★ Synopsis
- ★ Descripti
- ★ Syntax
- ★ Constraints
- ★ References
- ★ Relationship
- ★ Semantics

- ★ Default
- ★ Value
- ★ List
- ★ Section & Subtopic

D



10.3 Property

Identifiers extends Basic::Property with the ability to designate a property as an identifier for the containing element.

Properties

isID: Boolean [0..1] - True indicates this property can be used to uniquely identify an instance of the containing Class. Only one Property in a class may have isID==true.

Operations

No additional operations.

Constraints

[1] Property.isID can only be true for one Property of a Class.

Semantics

A Property with isID==true may be used as part of the URI identifying an object instance.

Rationale

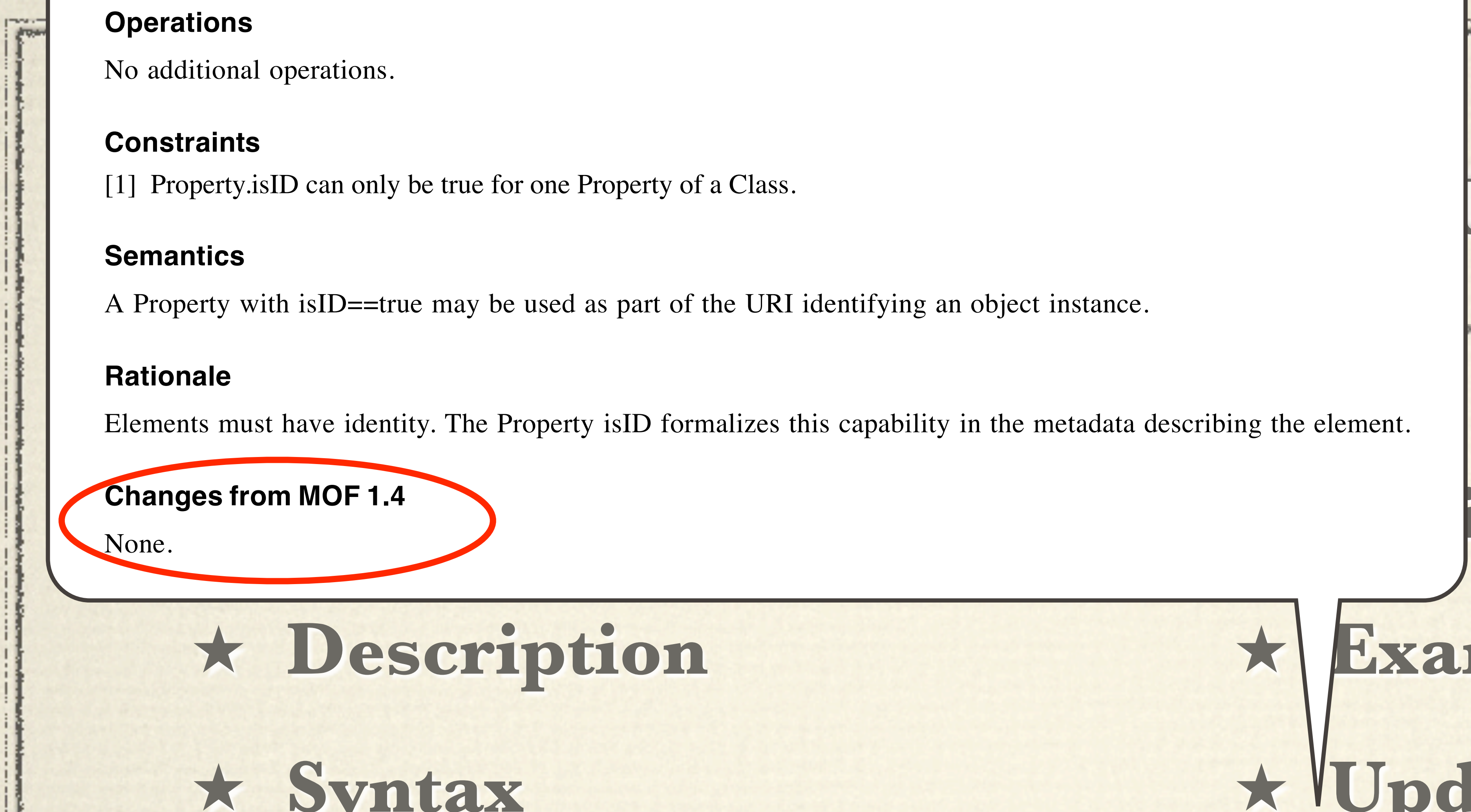
Elements must have identity. The Property isID formalizes this capability in the metadata describing the element.

Changes from MOF 1.4

None.

- ★ Synopsis
- ★ Descripti
- ★ Syntax
- ★ Constraints
- ★ References
- ★ Relationship
- ★ Semantics

- ★ Default
- ★ Value
- ★ List
- ★ Section & Subtopic



Operations

No additional operations.

Constraints

[1] Property.isID can only be true for one Property of a Class.

Semantics

A Property with isID==true may be used as part of the URI identifying an object instance.

Rationale

Elements must have identity. The Property isID formalizes this capability in the metadata describing the element.

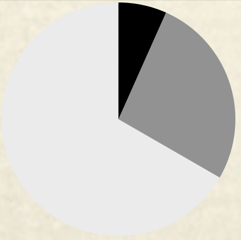
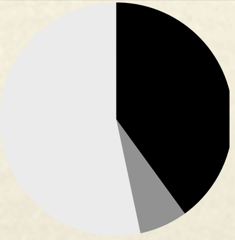
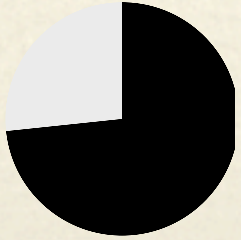
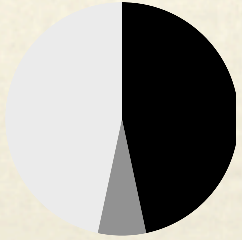
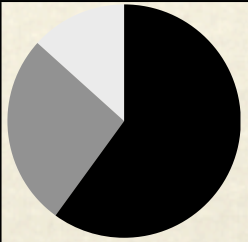
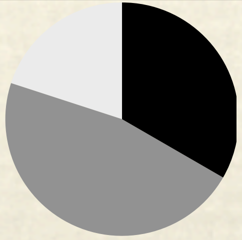
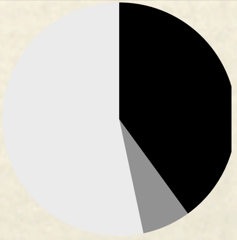
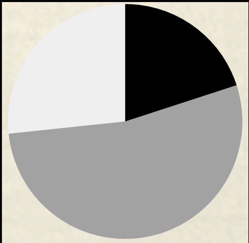
Changes from MOF 1.4

None.

ts

- ★ ionale
- ★ Description
- ★ Syntax
- ★ Constraints
- ★ References
- ★ Relationship
- ★ Semantics
- ★ Example
- ★ Update
- ★ Default
- ★ Value
- ★ List
- ★ Section & Subtopic

Mapping to LDF

Domain concept	IAL [Bac60]	Jovial [MIL84]	Design Patterns [GHJV95]	Smalltalk [Sha97]	Informix [IBM03]	C# [Sta06]	MOF [MOF06]	XPath [BBC+07]
synopsis	—	∼	intent	synopsis	∼	∼	∼	—
description	∼	—	motivation	definition	usage	∼	—	∼
syntax	—[a]	syntax	structure	∼	∼	∼	—	[NN][b]
constraints	—	constraints	applicability	errors	restrictions	∼	constraints	∼
references	—	—	related patterns	—	references	∼	—	∼
relationship	—	—	consequences	return value, refinement	related	return type	—	∼
semantics	—	semantics	collaborations	—	important	∼	semantics	∼
rationale	∼	notes	implementation	rationale	GLS, ES[c]	note	rationale	note
example	examples	examples	sample code, known uses	—	∼	example	—	∼
update	—	—	—	—	—	—[d]	changes	—
default	—	—	—	—	note	default values	—	—
value	—	—	also known as	conforms to	—	—	—	—
list	∼	—	—	messages, parameters	*terminals*	—	properties	∼
section	∼	—	—	—	∼	∼	—	∼
subtopic	—	types	participants	—	fields	parameters, methods	operations	functions
Coverage of LDF								

XPath case study

- Fully mapped:
  - note (<note>) → **rationale**
  - function (<proto>) → **subtopic**
  - productions (<scrap>) → **syntax**
  - ? → **description**

XPath case study

```
<prod id="NT-RelativeLocationPath">

<lhs>RelativeLocationPath</lhs>

<rhs><nt def="NT-Step">Step</nt></rhs>

<rhs>

| <nt def="NT-RelativeLocationPath">RelativeLocationPath</nt>

'/' <nt def="NT-Step">Step</nt>

</rhs>

<rhs>

| <nt def="NT-AbbreviatedRelativeLocationPath">AbbreviatedRelativeLocationPath</nt>

</rhs>

</prod>
```

XPath case study

- Partially mapped:
  - first **description** sentence → **synopsis**
  - **rationale** with “should be” → **constraints**
  - **rationale** with “[not] the same as” → **relationship**
  - **description** bits with “for example” → **example**
  - ...

Document transformations

- ★ Same as grammar transformations
- ★ Language evolution
- ★ Language convergence with documents
- ★ Documentation improvement
- ★ …

Document transformations

```
xldf:add-section(structured-section:((title:"For Expressions",
                  id:"id-for-expressions"),
                  ...));
xldf:move-section(id:"section-Function-Calls",
                   inside:"id-primary-expressions");
xldf:rename-id(from:"section-Function-Calls",
                to:"id-function-calls");
```

Related work: documentation

- LDF vs. DocBook
- LDF vs. DITA
- LDF vs. home-grown XML
- LDF vs. ???

Related work: research

- ★ Verification techniques on documentation
- ★ Wikis, eBooks, interactive tutorials, browsable grammars
- ★ Information retrieval
- ★ Natural language generation
- ★ Knowledge reuse
- ★ …

Conclusion / Future Work

- A unified format: LDF
    - derived from real language document
    - integrated with current research & infrastructure
- Language document engineering
- Round-tripping experiments
- Considerably large case study

Questions?

Thank you!