

Yves Vandewoude¹, Peter Ebraert², Yolande Berbers¹, Theo D'Hondt²

¹ Department of Computer Science, KULeuven, Celestijnenlaan 200A,B-3001 Heverlee, Belgium

² Programming Technology Lab, Vrije Universiteit Brussel, Pleinlaan 2, B-1050 Brussel, Belgium
 {yvesv, yolande}@cs.kuleuven.ac.be, {pebraert, tjdhondt}@vub.ac.be

Type Systems

Type Binding is the process of assigning a type to a variable. A distinction is made between *static and dynamic binding* depending on whether the binding occurs at compile-time or at run-time respectively.

Type Strength refers to the effectiveness with which a type system prevents type errors. A *strongly typed* language prevents any operation on the wrong type of data. In *weakly typed* languages there are ways to escape this restriction: coercion.

Type Checking is the process that verifies whether the operands of an operator have compatible types. Depending on when the type checks occur, the term *static type check* or *dynamic type check* are used.

A Type Error is an error which occurs when an operation is performed on the wrong kind of data.

Coercion: is an implicit conversion of a variable from one type to another.

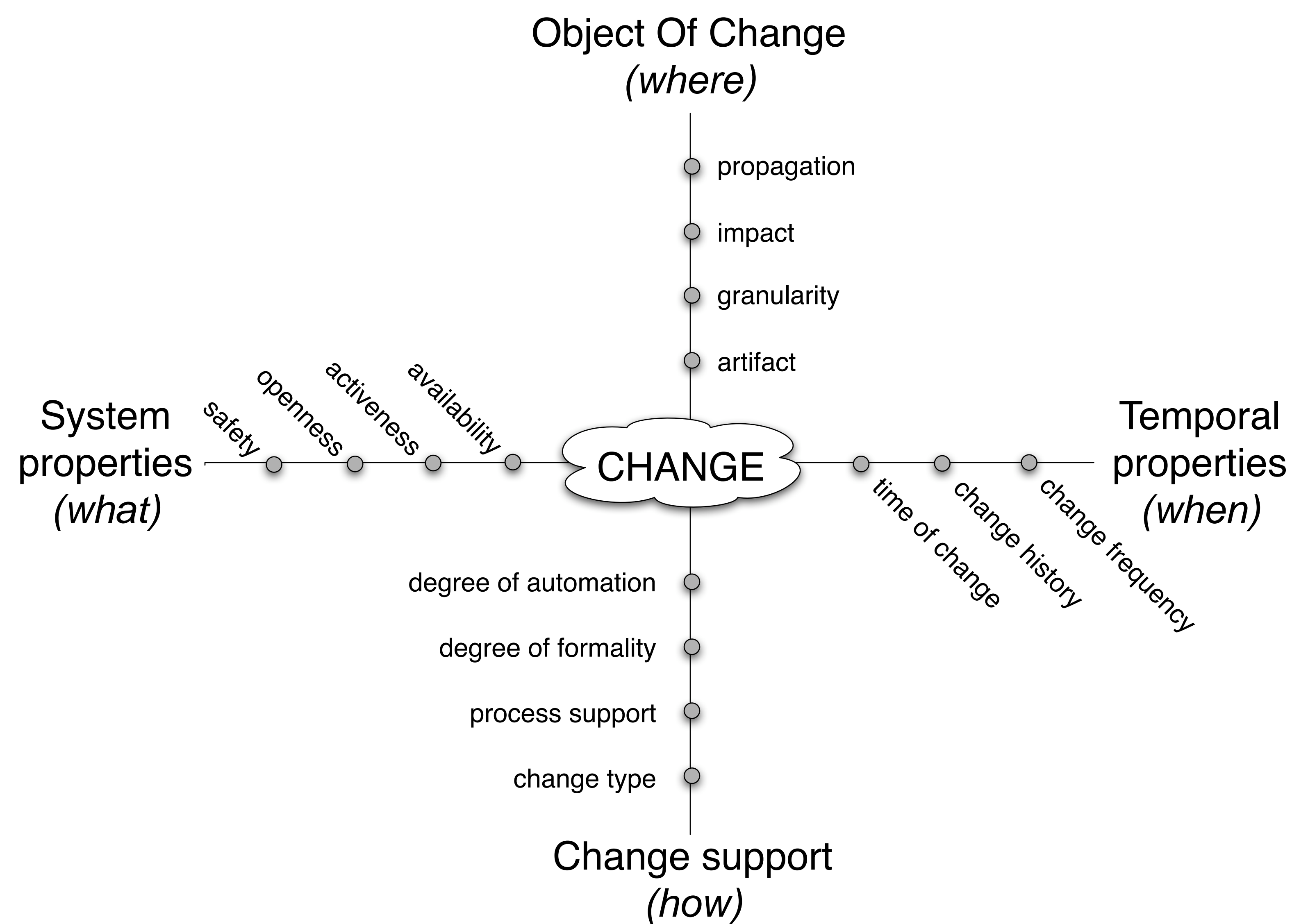
Binding vs. Checking

	Binding		
Dynamic	-	-	
Both			
Static		-	-
	Checking		
	Static	Both	Dynamic

Typing vs. Strength

Dynamic	Cecil	CLOS	Python	Javascript
	Scheme	Smalltalk	E	
Both	Ruby	Java		NewPython
	Sather	C#		
Static	Eiffel			C++
	Pascal			
	Strong			Weak
				C
				Typed Assembly Language

Evolution Taxonomy



Type impact on dynamic software evolution

Group	Dimension	Static		Both		Dynamic	
		Weak	Strong	Weak	Strong	Weak	Strong
Temporal (when)	Time of change						
	Offline changes	++	++	++	++	+	+
	Online changes	--	--	+/-	-	++	+
	Change history	++	+	++	+	+	+/-
Object of change (where)	Change frequency	-	--	-	--	++	+
	Anticipation	-	-	+/-	+/-	+	+
	Granularity						
	Coarse grained	++	++	++	++	+	+
	Fine grained	-	-	-	-	++	++
System properties (what)	Impact	+	++	+/-	+	--	-
	Change propagation	+/-	+/-	+/-	+/-	++	+
	Availability	-	-	+/-	+/-	+	+
	Openness	+/-	-	+	+/-	++	+
Change support (how)	Safety	+/-	+	+/-	+	--	-
	Degree of automation	-	-	+/-	+/-	+/-	+/-
	Degree of formality	+	++	+	++	-	+/-
	Change type	+	+	+	+	+/-	+/-

References:

- J. Buckley, T. Mens, M. Zenger, A. Rashid, and G. Kniesel. Towards a taxonomy of software change. *Journal of Software Maintenance and Evolution: Research and Practice*, 2003.
- Y. Vandewoude, P. Ebraert, Y. Berbers and T. D'Hondt. Influence of type systems on dynamic software evolution. Technical Report CW410, KULeuven, Belgium