Dynamic Software Architectures LTU

Ulysse Rémond

Student
Computer Science Department

ÉNS Paris-Saclay & LTU

31 août 2020

Outline

Context

- Shift from mass production

Context

- Shift from mass production to mass customization

Context

- Shift from mass production to mass customization
- Industrial systems now both move and are reconfigurated on-the-fly

Context

- Shift from mass production to mass customization
- Industrial systems now both move and are reconfigurated on-the-fly
- Additional components can be added or removed at any time

Context

- Shift from mass production to mass customization
- Industrial systems now both move and are reconfigurated on-the-fly
- Additional components can be added or removed at any time
- These systems are called **Dynamic Systems**

Article

- *Organizing definitions and formalisms for dynamic software architectures*, 2004

Article

- *Organizing definitions and formalisms for dynamic software architectures*, 2004
- Autor : Jeremy Bradbury.

Article

- *Organizing definitions and formalisms for dynamic software architectures*, 2004
- Autor : Jeremy Bradbury.
- A survey about Dynamic Software Architectures (DSA) :
  - First reviews the area and notices that software architectures have yet to be assessed in terms of evolutionary change
  - Then evaluates DSA according to three principles

Evaluating Dynamic Architectures

- What type of change is supported?

Evaluating Dynamic Architectures

- What type of change is supported?
- What kind of process implements the change?

Evaluating Dynamic Architectures

- What type of change is supported?
- What kind of process implements the change?
- What infrastructure is available to support the change process?

2 - Dynamic Software Architectures

When is the change known ?

Programmed Dynamism

The change is known at design time and is triggered by the system.

Ad-Hoc Dynamism

The changes are not known before runtime. It is often initiated by the user or by a kind of external, unpredictable event.

. M. Endler, 1994

Middleware

Adaptative Dynamism

A kind of programmed dynamism. The change is known at design time and triggered by predefined events. It is implemented in the middleware by selecting a configuration among a set of predefined configurations.

Constructible Dynamism

A kind of ad-hoc change, triggered by an external event.

- Initial system configuration $\Longleftrightarrow$ Description Language
- Architectural changes $\Longleftrightarrow$ Modification language

. J. Andersson., 2000

Middleware

Adaptative Dynamism

A kind of programmed dynamism. The change is known at design time and triggered by predefined events. It is implemented in the middleware by selecting a configuration among a set of predefined configurations.

Constructible Dynamism

A kind of ad-hoc change, triggered by an external event.

- Initial system configuration $\Longleftrightarrow$ Description Language
- Architectural changes $\Longleftrightarrow$ Modification language
- Changes are enforced via a Dynamic Updating System in the middleware

. J. Andersson., 2000

Constraints on the modifications

Intelligent Dynamism

Very similar to adaptative dynamism, but instead of having a predefined set of configurations, they are created on-the-fly, and the most opportune one is implemented.

. J. Andersson., 2000

Constrained Run-Time Dynamism

Change occurs only when proven "safe", with respect to a set of constraints on the architectural topology, and the state of the program.

. P. Oreizy, 1996, Issues in the Runtime Modification of Software Architectures

Connectors

Permanent connectors

Connectors set or reset interconnections between components according to given conditions of applicability.

The architecture of the system may get cluttered by a high number of connectors.

Transient connectors

Connectors are removed and added according to a boolean interaction condition, so that the architecture only depicts the active connectors.

. Michel Wermelinger & José Luis Fiadeiro, 1998

. Same & José Meseguer, 1999

Distributed Architectures

Self-organising Architectures

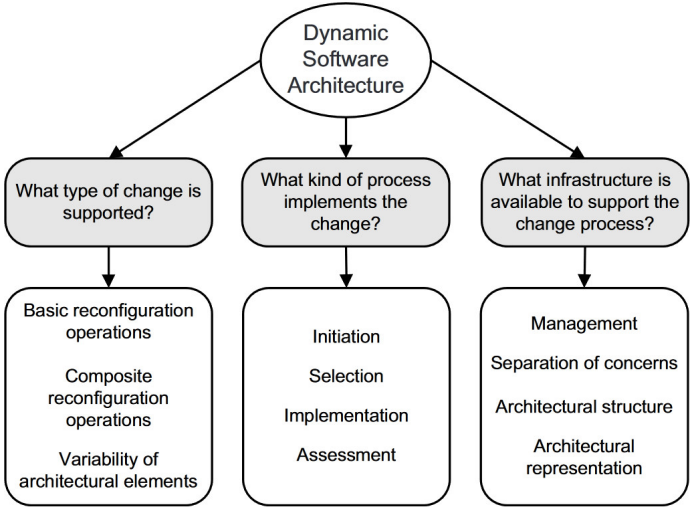
- No centralized control
- Components can add or remove themselves to the architecture
- Constraints of the components describe the system

Self-repairing systems

Very similar to Self-organising Architectures, but not necessarily distributed, and following this control loop :

- Monitoring
- Interpretation
- Reconfiguration

Classification Criteria



Examples of DSA

		Initiation		Selection				Implementation	Assessment		
		Internal	External	Explicit	Pre-defined	Constrained from Pre-defined set	Unconstrained		Execution/ Simulation	Direct Formal Analysis	Formal Analysis after Translation
Graph	Le Métayer approach	●	●	○	●	●	○	graph rewriting rules	○	●	○
	Hirsch et al. approach	○	○	○	○	○	○	graph rewriting rules	○	○	○
	Taentzer et al. approach	?	?	○	●	○	○	graph rewriting rules	○	●	○
	CommUnity	●	●	◉	●	●	○	category theory	○	◉	○
	CHAM	●	●	○	●	○	○	evolution CHAM reaction rules	○	○	○
Process Algebra	Dynamic Wright	●	○	○	●	●	○	CSP	○	●	●
	Darwin	●	○	○	●	○?	○	π-calculus	●	●	●
	LEDA	●	○	○	●	●	○	π-calculus	◉	○	●
	PiLar	●	○	○	●	○	○	CCS	○	○	●
Logic	Gerel	●	◉	●	●	?	○	first order logic	◉	○	○
	Aguirre-Maibaum approach	●	○	○	●	●?	○	first order logic, temporal logic	○	○	●
	ZCL	●	○	○	●	○	○	Z operation schema	◉	◉	○
Other	C2SADEL	○	◉	●	○	○	○	AML	◉	◉	○
	Rapide	●	○	○	●	●	○	where statement, exec. arch. events	●	●	○

Other classifications

- A Classification of Dynamic Reconfiguration in Component and Connector Architecture Description Languages [Wortmann& Al.2017]
- Moving architectural description from under the technology lamppost [Medvidovic & Al. 2006]

3 - Dynamical Systems

Example

- Can be used to model continuous Cyber-Physical Systems (CPS)

Modelisation de performance humaine

- 1996 T Henzinger : *The Theory of Hybrid Automata*

4 - Conclusion

4 - Conclusion

- Software reconfiguration
- Physical Movements

This presentation was only done to proofread what has been done on paper about reviews, having a concrete support on which write what I found and therefore enable the formation on perspectives about it. It is still unfinished.