Context-Aware Software-Intensive Systems
An autonomic approach

Vladimiro Sassone

University of Sussex, UK

Engineering Software-Intensive Systems (22.05.04)

A Forthcoming Computing Paradigm

- Ubiquitous Computing: migrating software executing on networks owned and operated by others.
  Countless `pervasive' devices equipped with limited resources and computing power will support end-to-end applications which far exceed their own capabilities.

Challenges

Scalability, Variability, Context-Awareness

A Forthcoming Computing Paradigm

- Ubiquitous Computing: migrating software executing on networks owned and operated by others.
  Countless `pervasive' devices equipped with limited resources and computing power will support end-to-end applications which far exceed their own capabilities.

> **Challenges**
>
> Scalability, Variability, Context-Awareness

- System Complexity is rocketing beyond our ability to design, comprehend and control. It approaches that of biosystems (e.g. economic systems).
- We do: Understand the behaviour of components in isolation.
  We don't: Understand the global behaviour of interacting components.

A Forthcoming Computing Paradigm

- Ubiquitous Computing: migrating software executing on networks owned and operated by others.
  Countless 'pervasive' devices equipped with limited resources and computing power will support end-to-end applications which far exceed their own capabilities.

Challenges

Scalability, Variability, Context-Awareness

- System Complexity is rocketing beyond our ability to design, comprehend and control. It approaches that of biosystems (e.g. economic systems).
- We do: Understand the behaviour of components in isolation.
  We don't: Understand the global behaviour of interacting components.
- Not likely to change soon: Need to "design for autonomy."

Autonomous Systems

- exhibit context-dependent behaviour to fulfill specific goals, possibly in complete isolation, and based on previously gathered information.

Examples

Complex biosystems
Beagle2 probe on Mars
Wireless ad-hoc networks
Electronic controlled transport systems
Health monitoring systems
...

Autonomous Systems

- exhibit context-dependent behaviour to fulfill specific goals, possibly in complete isolation, and based on previously gathered information.

Examples

Complex biosystems
Beagle2 probe on Mars
Wireless ad-hoc networks
Electronic controlled transport systems
Health monitoring systems
...

- characterised by a degree of independence in making decisions and adapting to unforeseen environmental conditions. Often entail collaborative or competing aspects, self-organisation, emergent behaviour.

Autonomy as a Design Principle

- Need to design systems which build models of the world, gather evidence, learn, and progressively increase confidence in their own autonomous decisions.
- More ambitiously, want to apply concepts and tools from the realm of autonomic systems for the design of systems which must be highly adaptable during their lifetime.

Examples

access control systems
privacy and security systems
traffic control systems . . .

Autonomy as a Design Principle

- Need to design systems which build models of the world, gather evidence, learn, and progressively increase confidence in their own autonomous decisions.
- More ambitiously, want to apply concepts and tools from the realm of autonomic systems for the design of systems which must be highly adaptable during their lifetime.

Examples

access control systems
privacy and security systems
traffic control systems . . .

- Not a bio-inspired approach: Biological systems are fundamentally non-linear, and thus largely unpredictable. They exhibit splendid properties of self-organisation, context-awareness, adaptability and autonomy, but work by trial and error, on evolutionary timescales.

Autonomy as a Design Principle

- Need to design systems which build models of the world, gather evidence, learn, and progressively increase confidence in their own autonomous decisions.
- More ambitiously, want to apply concepts and tools from the realm of autonomic systems for the design of systems which must be highly adaptable during their lifetime.

Examples

access control systems
privacy and security systems
traffic control systems . . .

- A CS approach: Provide solid foundations for autonomic systems, via a system-and-theory integrated approach: make abstract models, use them for predictions, embed the in middleware and programs.

Two paradigmatic examples

Examples

Third-party resource usage

- negotation of lease for resources
- pricing policies based on context-awareness
- code reputation . . .

London congestion charges

- traffic monitoring
- pricing per time
- car reputation . . .

The first steps

- Understanding autonomy at its foundations. Initially, a model for processes to explore their surroundings via risk-assessment techniques (e.g. trust and reputation).

  More generally, we look at context-awareness as the central notion to equip systems with the tools for autonomy.

The first steps

- Understanding autonomy at its foundations. Initially, a model for processes to explore their surroundings via risk-assessment techniques (e.g. trust and reputation).

  More generally, we look at context-awareness as the central notion to equip systems with the tools for autonomy.

Basic Enabling Theories

Concurrency & Mobility
Game Theory & MicroEconomics
Trust & Reputation Theory

The first steps

- Understanding autonomy at its foundations. Initially, a model for processes to explore their surroundings via risk-assessment techniques (e.g. trust and reputation).

  More generally, we look at context-awareness as the central notion to equip systems with the tools for autonomy.

Basic Enabling Theories

Concurrency & Mobility
Game Theory & MicroEconomics
Trust & Reputation Theory

Immediate Challenges

Integrate several theories
Yield models & validation mechanisms
Design languages & middleware

Example: A model of trust

$$N, M ::= \epsilon \quad \text{(empty)} \mid N \mid N \quad \text{(net-par)} \mid a\{ P \}_\alpha \quad \text{(principal)} \mid (\nu n)\, N \quad \text{(new-net)}$$

$$P, Q ::= \mathbf{0} \quad \text{(null)} \mid Z \quad \text{(sub)} \mid P \mid P \quad \text{(par)} \mid (\nu n)\, P \quad \text{(new)} \mid\, !P \quad \text{(bang)}$$

$$Z ::= p \cdot \tilde{u}(\tilde{v}) . P \quad \text{(output)} \mid \phi :: p \cdot \tilde{u}\langle \tilde{v} \rangle . P \quad \text{(input)} \mid Z + Z \quad \text{(sum)}$$

Communication

$$\frac{\beta \vdash \phi \qquad \alpha' = \alpha + [b \cdot \tilde{l} \triangleright \tilde{m}] \qquad b : \tilde{m} \odot p : \tilde{x} = \sigma}{a\{ p \cdot \tilde{l}(\tilde{x}) . P' \}_\alpha \mid b\{ \phi :: a \cdot \tilde{l}\langle \tilde{m} \rangle . Q \}_\beta \to a\{ P\sigma \}_{\alpha'} \mid b\{ Q \}_\beta}$$

Conclusion

Next Step

A lot of work needed

Refine these ideas until they make sense

Find pilot projects, apply for fundings