Trends in Middleware for Ubiquituous Computing : WComp Solution

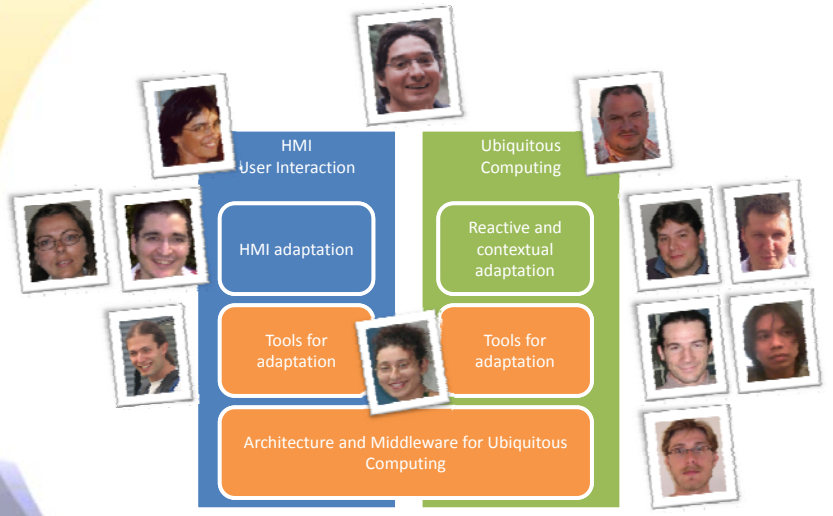
DAAD Summer CTDS 09, 24th - 26th September - Tunis

Ass. Prof. Jean-Yves Tigli – http://www.tigli.fr
at the University of Nice Sophia Antipolis, delegated at INRIA in the team PULSAR

In collaboration with S. Lavirotte and G. Rey

Our experimental platform is as a Sharpdevelop Addon on .Net Framework.
http://rainbow.i3s.unice.fr/wikiwcomp/

Rainbow research group



PART 1 : Requirements, trends, open issues associated with middleware for ubiquitous computing

I.1 What do we mean by Ubiquitous Computing?

- [Mark Weiser 1991]

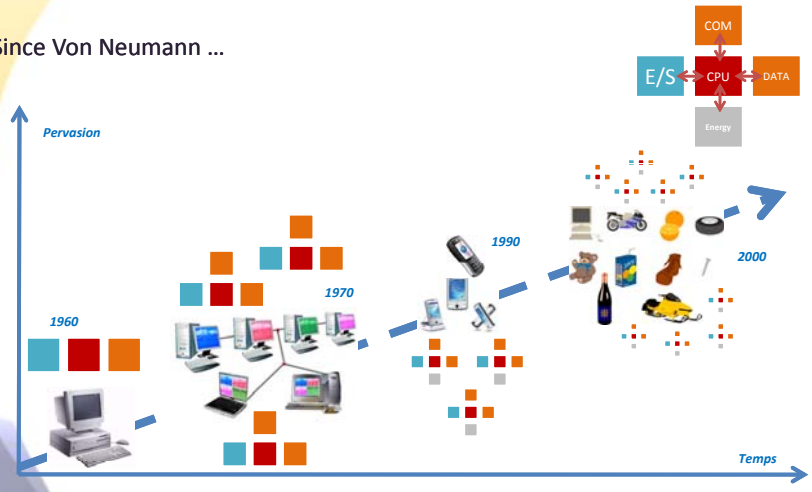
« Silicon-based information technology, is far from having become part of the environment. »

«The most profound technologies are those that dissappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.»

Scientific American,
Vol. 265 N.9, pp. 66-75, 1991

I.1 Evolution of Computing

- Since Von Neumann …



Nanocomputers & Swarm Intelligence, John Wiley & Sons - ISTE, London, 2008, ISBN 9781847040022

I.1 Variations of Ubicomp

- Embedding for smart control
  - Embedded systems for cars, airplanes, etc.
- Creating new computing devices
  - Hi-tech, silicon-based gadgetry, e.g. PDAs, cell phones, mp3 players, active displays
- Connecting the existing physical world to a computational infrastructure
  - Ordinary objects and tasks re-evaluated and extended with computational/communication capabilities

II.2 New challenge due to Mobility and Heterogeneity of devices

- An ubiquitous environment
- Spheres of interaction of devices, from Personal Area Network to World Wide Web

S. Arbanowski, M. Lipka, K. Mössner, K. Ott, R. Pabst, P. Pulli, A. Schieder, M. A. Uusitalo. The **WSI Reference Model** for the Wireless World. *Proceedings of IST Mobile Summit 2003*.

II.2 New challenge due to Mobility and Heterogeneity of devices

- Interoperability
  - Heterogeneity of devices
- Dynamicity
  - Mobility
  - Discoverability

SOA for Device

1.3 Main Ubiquitous Computing Characteristics

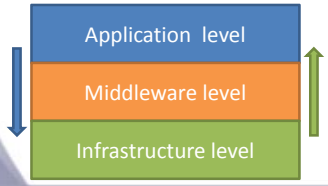
- Three main characteristics are :
  - Use embedded devices in a real environment
  - Deal with Multiple Heterogeneous Devices
  - Deal with Highly Dynamic variation at Runtime

I.4 Main Ubiquitous Computing Requirements

- Main requirements are :
  - *Real Environment =>* Event based interaction from devices
  - *Heterogeneous Devices =>* Discovery of new software entitites and devices
  - *Highly Dynamic at Runtime =>* Deal with dynamic appareance and dispareance of devices
  - *Highly Dynamic at Runtime =>* Deal with dynamic composition (at runtime)
  - *Highly Dynamic at Runtime =>* Deal with dynamic adaptation (self- adaptation)

I.5 New challenge and open issues

- Ubiquitous Computing applications are continuously interacting with a real world, partly unknown at design time and, always changing at runtime in uncountable manner
- We witness to a kind of inversion in the classical software methodology where the software applications levels are much more stable and stationary than the software infrastructure level.



I.6 Multi-Domain adaptation as open issue

- Ubiquitous Middleware must continuously adapt at runtime, application requirements to changing computing environment (due to mobility) in multiple domains :
  - HMI,
  - Power,
  - Network bandwidth,
  - Devices availability, …

I.7 Reactive adaptation as open issue

- Reactive adaptation is defined the ability for the Ubiquitous applications to perceive the environment and adapt to changes in that environment in a timely fashion.
- Ubiquitous Middleware must provide reactive adaptation mecanism to changing operational environment.

I.8 Semantic adaptation as open issue

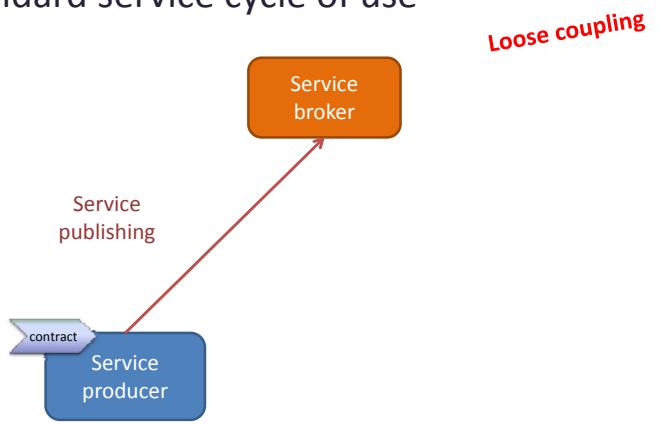
- Ubiquitous Middleware must match at run-time the current operational environment and application requirements.

PART II : Our solution, called *WComp*

1. Software Infrastructure based on Web services for Device
2. Local composition (LCA model),
3. Distributed composition (SLCA model) and
4. Reactive adaptation using Aspects of Assembly (AA)

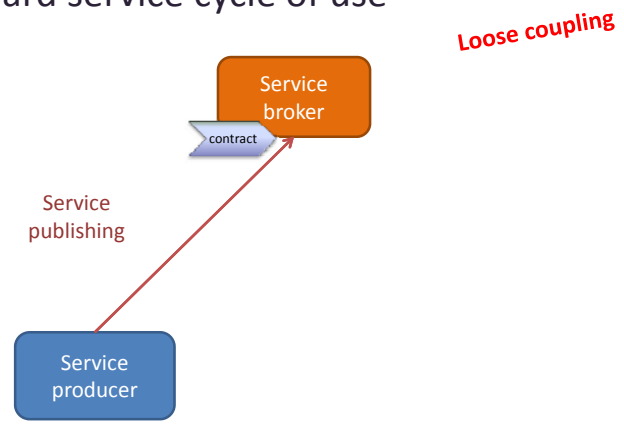
II.1 Service Oriented Architecture

- Standard service cycle of use



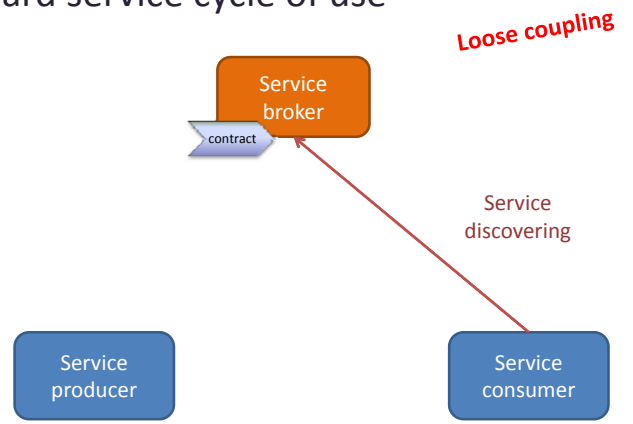
II.1 Service Oriented Architecture

- Standard service cycle of use



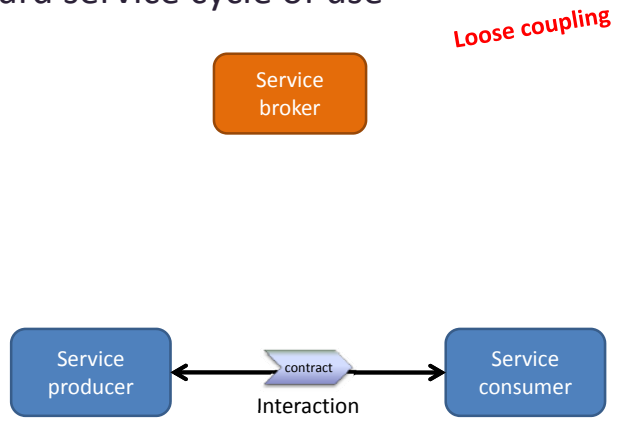
II.1 Service Oriented Architecture

- Standard service cycle of use



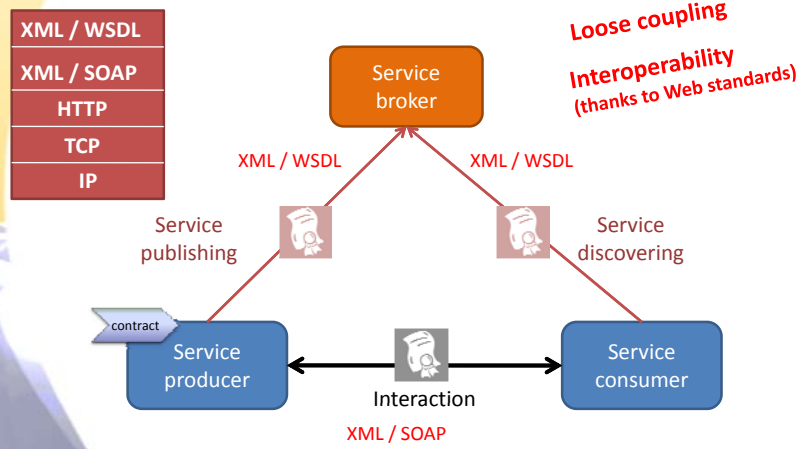
II.1 Service Oriented Architecture

- Standard service cycle of use



II.1 Web Service Oriented Architecture

- Web Services using Web technologies

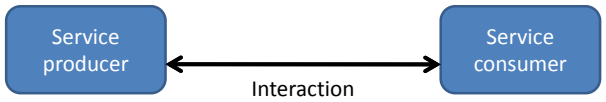


II.1 Web Services for Devices

- New challenges for multi-device and mobile sytems

Loose coupling

Interoperability



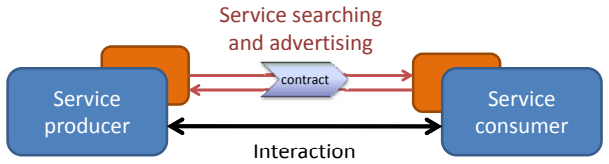
II.1 Web Services for Devices

- New challenges for multi-device and mobile sytems
  - Decentralized Dynamic Discovery

Loose coupling

Interoperability

Contextual discovery, high dynamicity



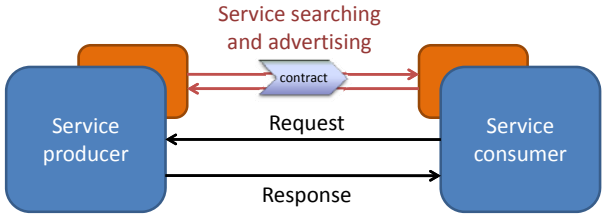
II.1 Web Services for Devices

- New challenges for multi-device and mobile sytems
  - Traditional interactions: invocations

Loose coupling

Interoperability

Contextual discovery, high dynamicity



II.1 Web Services for Devices

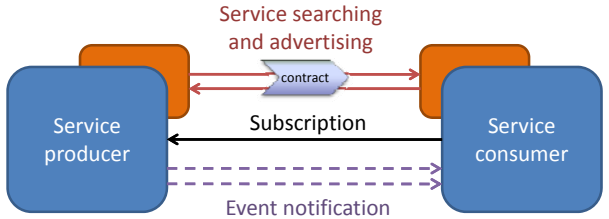
- New challenges for multi-device and mobile sytems
  - New ways of interacting: Eventing (Event-Driven Architecture)

Loose coupling

Interoperability

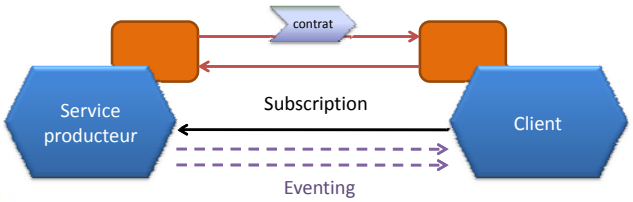
Contextual discovery, high dynamicity

Reactivity

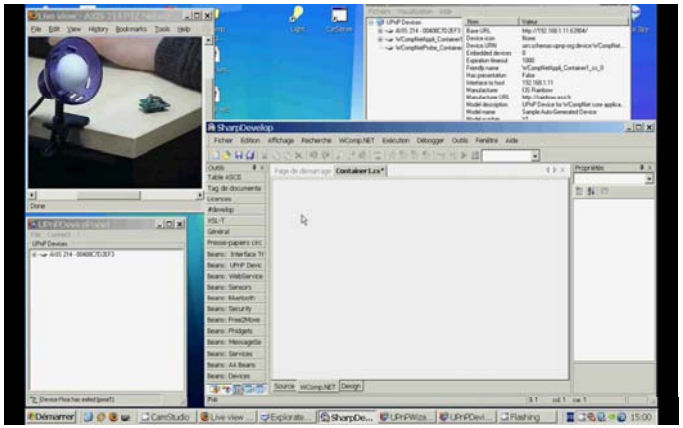


II.1 Software Infrastructure based on Web services for Device

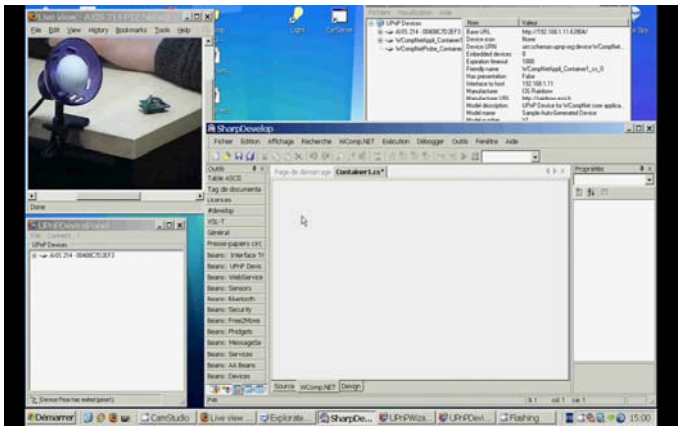
- Meet Service for Device characteristics
  - Example : UPnP and DPWS standards
- Three main evolutions from Web Services :
  - Distributed Service Publication / Discovery
  - Appareance / Disappearance Management
  - Eventing interaction model



II.1 Demo : Services for Physical Devices in WComp



II.1 Demo : Services for Virtual Devices in WComp



II.2 Both Composition levels

- Contrarly to most middleware approaches, *distribution must be explicit* to deal with the evolution of the infrastructure
- We need to distinguish between always available components and appearing / disappearing components

We distinguish :

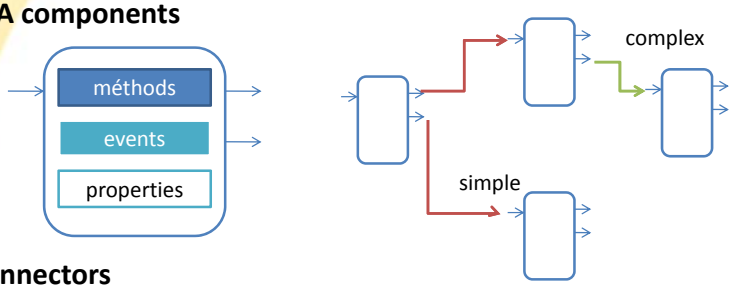
- Local Composition : LCA (Lightweight Component Model) for each application execution node.
- Distributed Composition : SLCA (SLCA (Service Lightweight Component Model) to enable application execution node to communicate between them.

II.2 Main Features of LCA Model :

- Goal :
  - Allow to compose Services for Device between them towards a multiple devices ubiquitous application.
- Principles
  - LightWeight Components Approach :
    - Like OpenCom [204], JavaBeans [96], PicoContainer [205]
  - On the same execution node
  - For each execution node, a container dynamically manage the assembly of components
  - Event-based interaction between components
  - Blackbox LightWeight Components

II.2 LCA components, ports and connectors

LCA components



Connectors

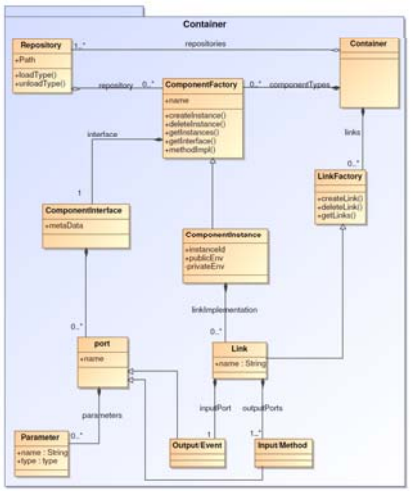
Simple Event based Connector

C1.Event (param) → C2.Method (param)

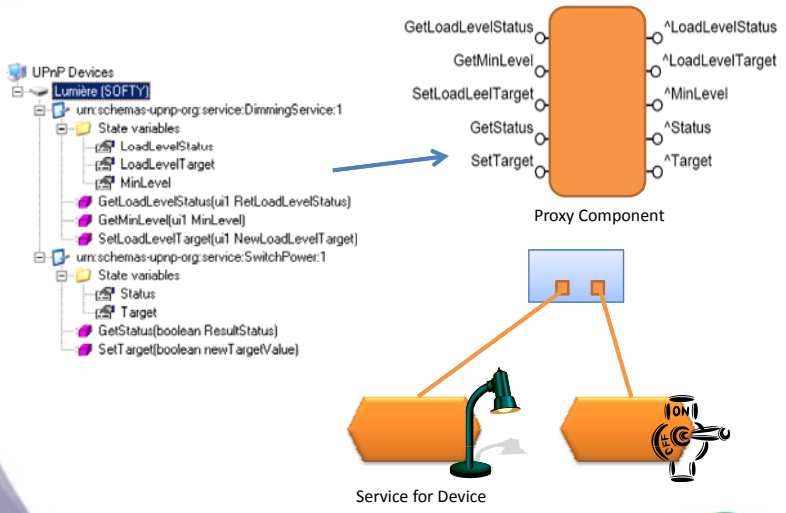
Complex Event based Connector

C1.Event (param) → C2.Method (C1.GetAProperty())

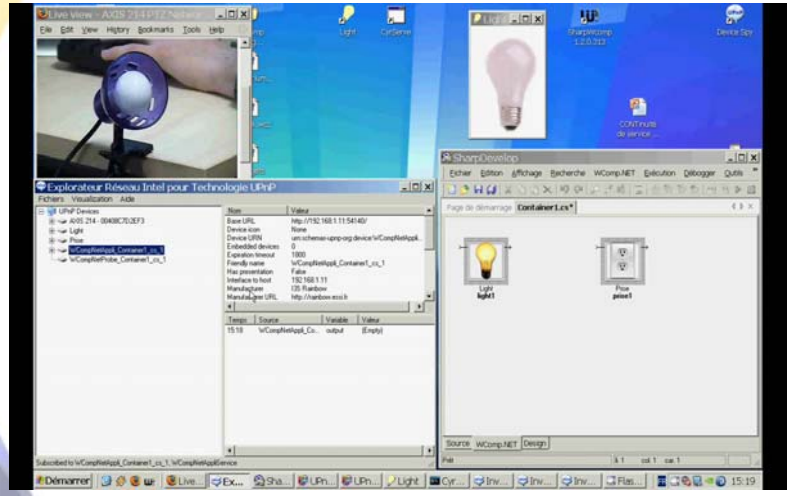
II.2 MetaModel of LCA



II.2 LCA Proxy components to access to Services for Devices

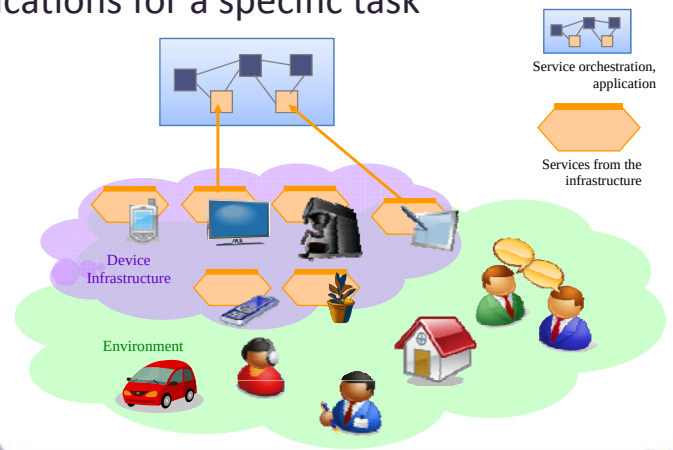


II.2 Demo : LCA in WComp



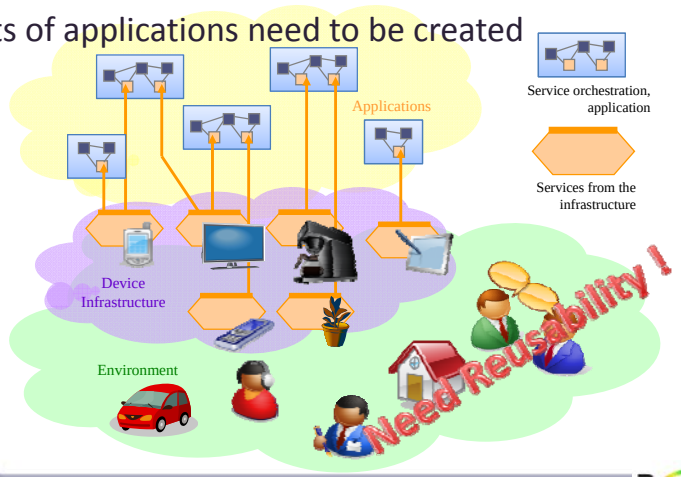
II.2 Application using Web Services for Devices

- Service orchestrations create service-based applications for a specific task

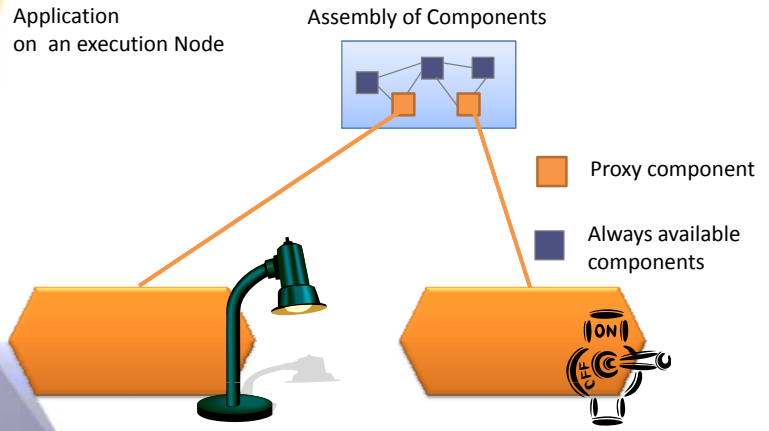


II.2 Applications using Web Services for Devices

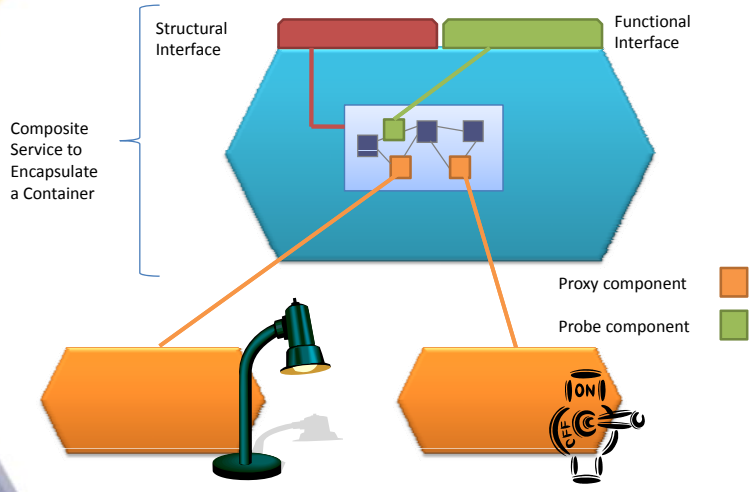
- Applications are specific and not reusable
  - Lots of applications need to be created



II.2 Distributed Compistion : From LCA to SLCA

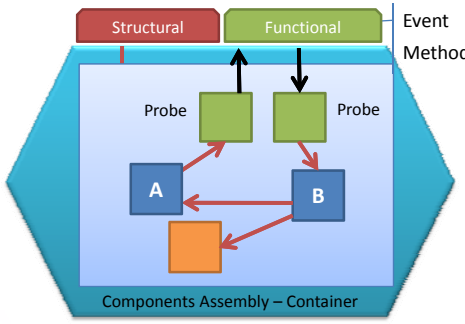


II.3 Distributed Compistion : From LCA to SLCA



II.3 Composite Service : SLCA

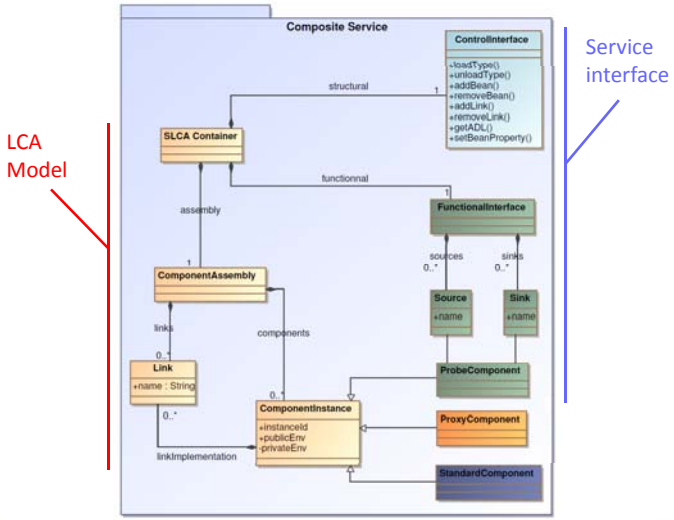
- Probe Components



Probe emitter :
Local event → Service Event

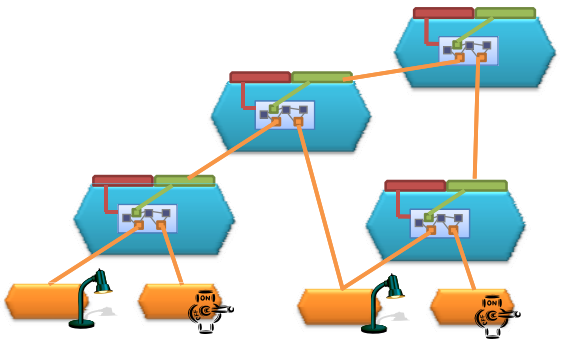
Probe receiver :
Service Method Invocation →
Local Event (Request)

II.3 MetaModel of SLCA

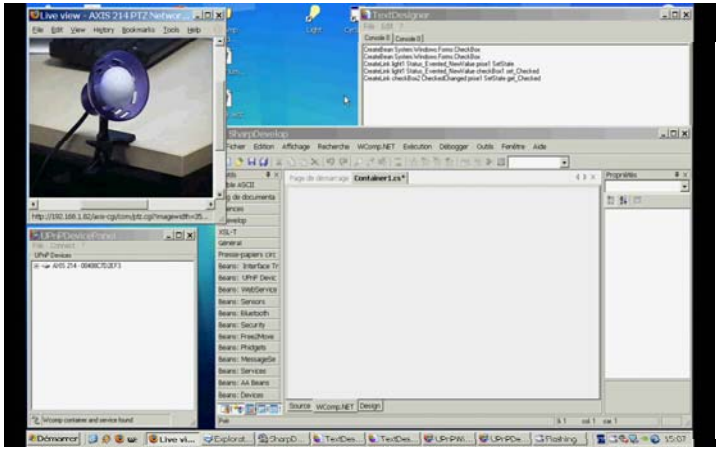


II.3 Distributed and Dynamic composition with SLCA, Demo in WComp

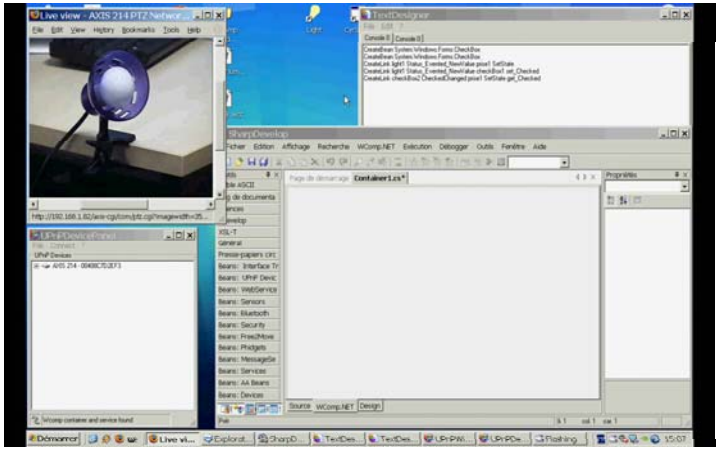
The Ubiquitous Applications are spreading in a graph of Composite Services for Devices



II.3 SLCA Demo in WComp : Control Interface



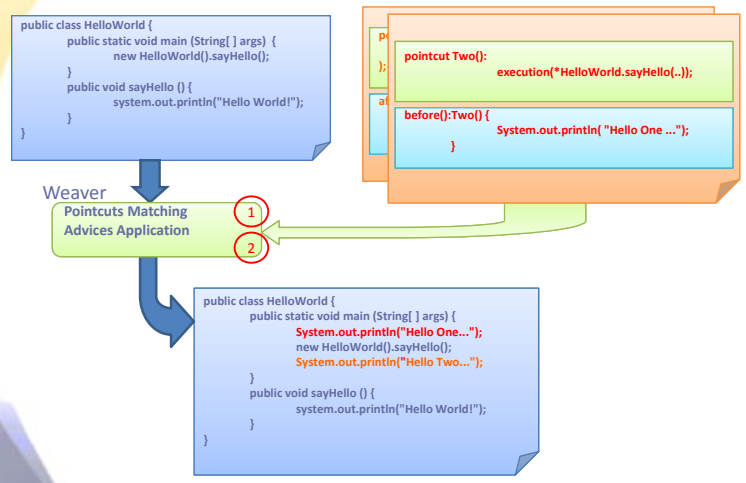
II.3 SLCA Demo in WComp : functionnal Interface



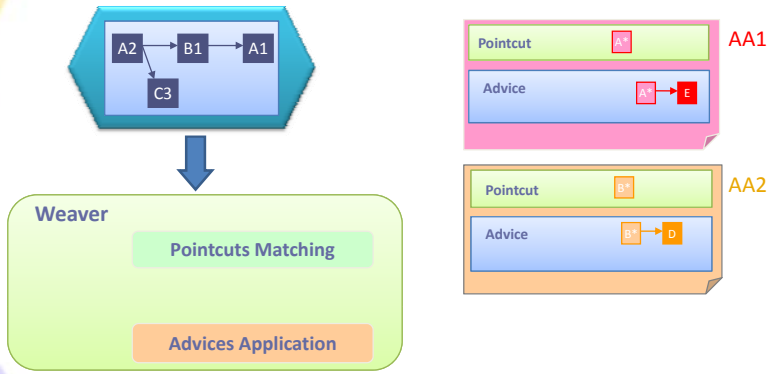
II.4 Reactive Adaptation: Aspect of Assembly

- *Aspect of Assembly*
- *Demo : AA in WComp*
- *Experiments and Results*

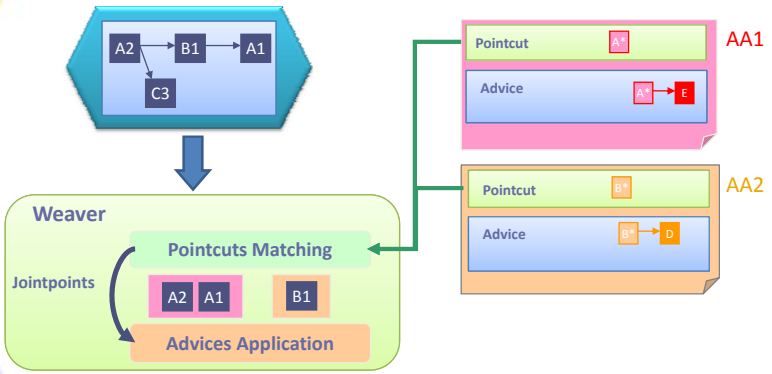
II.4 reminder : AOP Principles



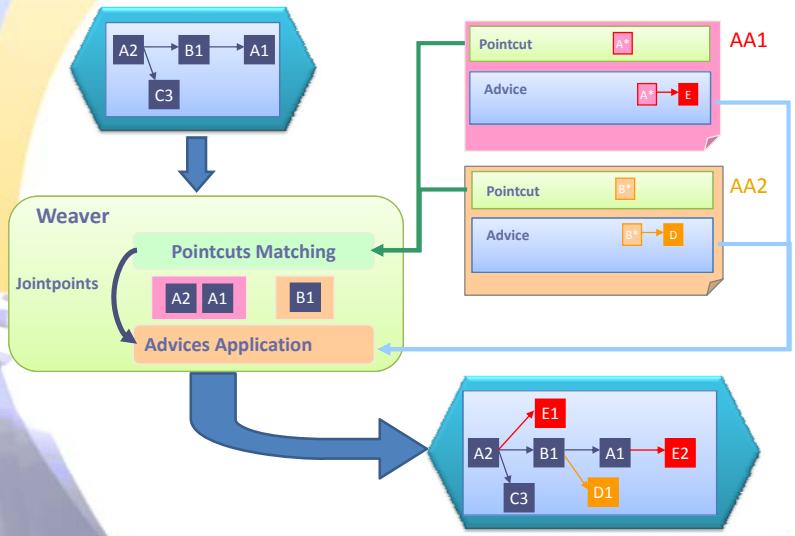
II.4 Aspect of Assembly Principles



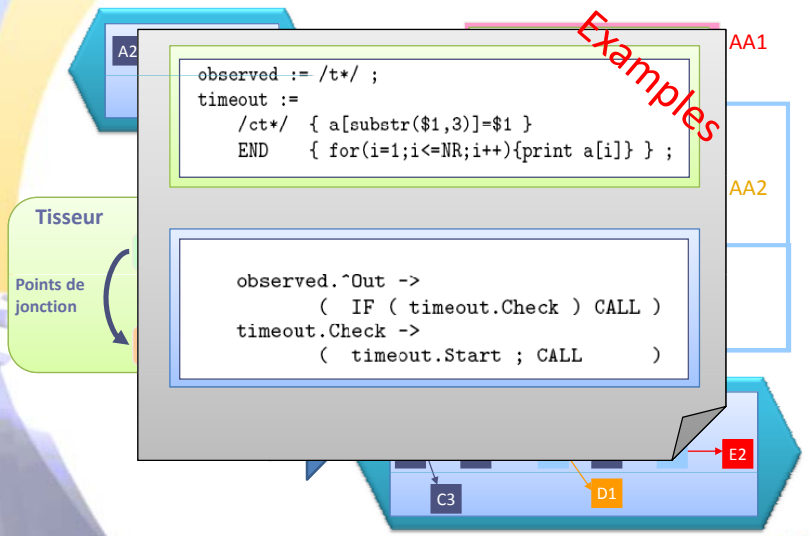
II.4 Aspect of Assembly Principles



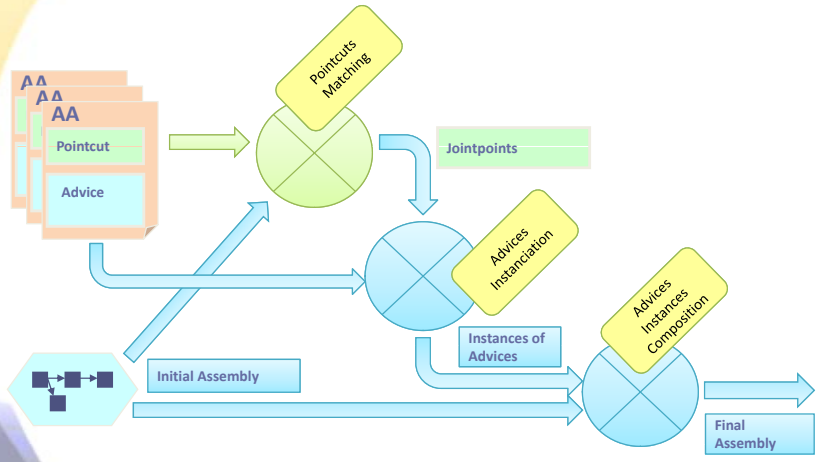
II.4 Aspect of Assembly Principles



II.4 Aspect of Assembly Principles



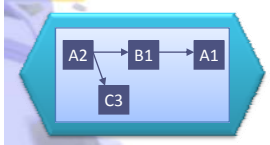
II.4 Internal Architecture of the AA Weaver



II.4 AA: A first Pointcut Language


```
observed := /t*/ ;
timeout :=
    /ct*/   { a[substr($1,3)]=$1 }
    END     { for(i=1;i<=NR;i++){print a[i]} } ;
```

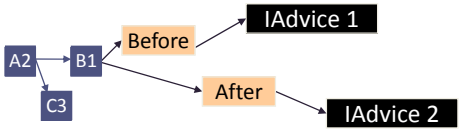
ct3, ct2, ct1, …



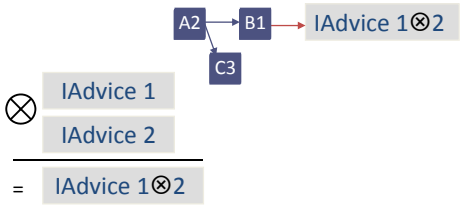
Regular expression based Syntactic Matching

II.4 AA: I-Advices, Composition, and Conflits

- External Composition :
  - I-Advices are « blackbox »
  - I-Advices are scheduled
  - Before, After, Around …



- Internal Composition with Merge :
  - I-Advice are « whitebox »
  - Conflicted I-Advices can be merged according to a specific logic and its properties (ex. ISL [Berger 01] , ISL4WComp, BSL [Cheung 09] …)



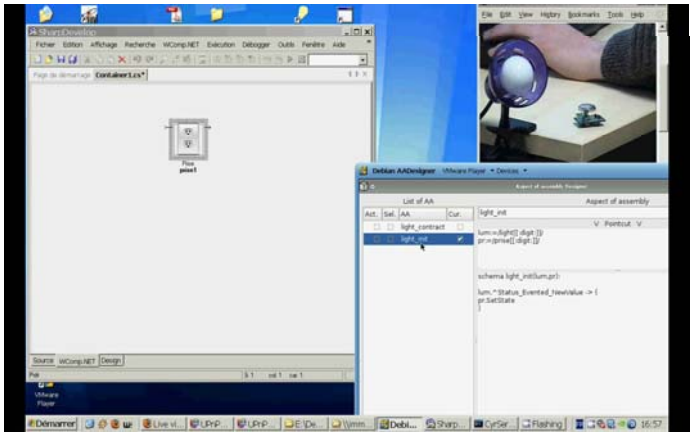
AA : Example of specific merging logic and its properties

- Merging logic is based on rules modified according to the Advice language
- example of prooved properties in the composition / merging logic :

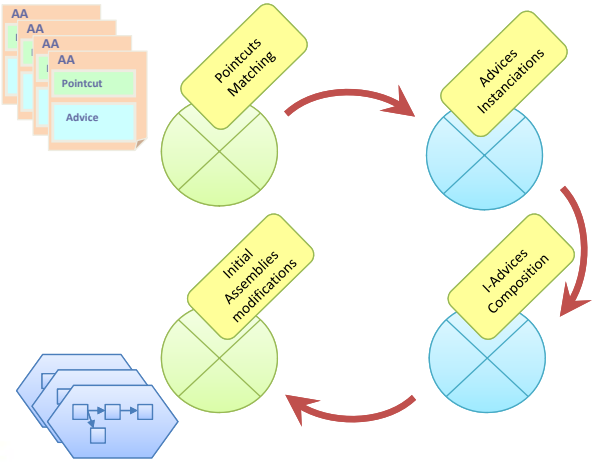
Commutativity : AA0 ⊗ AA1 = AA0 ⊗ AA1
Associativity : (AA0 ⊗ AA1) ⊗AA2 = AA0 ⊗ (AA1 ⊗ AA2)
Idempotence : AA0 ⊗ AA0 = AA0

- Weaving mecanism became « Symmetric »
- Designer can apply a set of AA without caring of the their order.

II.4 Demo : AA in WComp

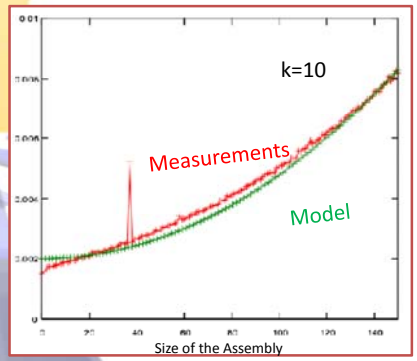


II.5 Reactivity and Weaving Cycle



II.5 Cost of the weaving cycle : Pointcut Matching

$$D = a_1 . \sum_{i=1}^{k} (\delta_i + 1) . c^2 + a_2$$



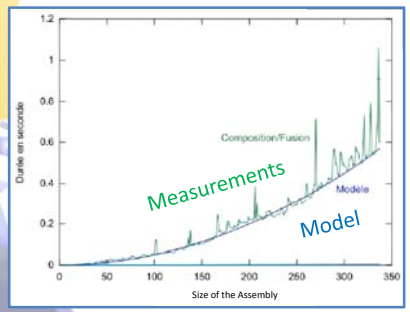
- Parameter identification
  - δ_i : nb of applications of the advice *i*
  - *c* : nb of components

$$a_1 = 28010^{-9}$$
$$a_2 = 2.10^{-3}$$

- Depend on the size of the initial assembly and the number of AA

II.5 Cost of the weaving cycle : Weaving and modification

$$K = b.n^0 . \sum_{i=1}^{N} n^i (1 + p_i . C(g_0, g_i))$$



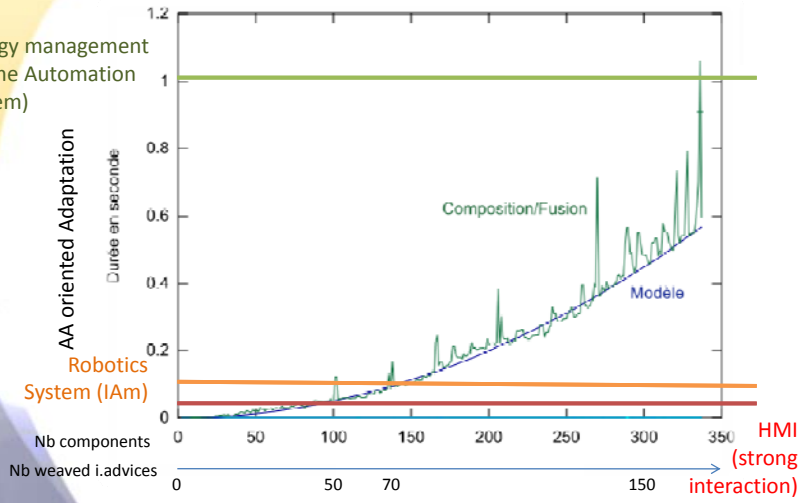
- Parameter Identification
  p_i : proba fusion
  C : merging cost
  n : number of rules
  N : number of I-advices

$$b = 2{,}6.10^{-6}$$

- Only depend on the number of weaved AA

Nb components => Nb weaved AA

II.5 Reactivity and Adaptation : Experiments and Results



III Conclusion and Future Works

III.1 Conclusion

	Reactive adaptation	Semantic adaptation	Multi-Domain adaptation
AA composition	✓	low	✓

WComp	Event based interactions	Dynamic composition (at runtime)	Dynamic publication (at runtime)
LCA composition	✓	✓	
SLCA composition	✓	✓	✓

WComp	Event based interactions	Discovery of devices at runtime	Deal with appareance and dispareance of devices at runtime
Software Infrastructure	✓	✓	✓

III.2 Future Works in WComp

- Multi-Domain weaving for AA to adapt Mobile Workers applications (Cf. CONTINUUM project of the French National Research Agency towards « Continuity of Service »)
- From AA to AOM (Aspect oriented Modelling) : a way to generalize Aspect to Adapt target architectures according to their model
- Improving of Pointcut Matching algorithms from Ontology-Based Metadata and mapping between ontologies (Cf. CONTINUUM project of the French National Research Agency towards « Continuity of Service »)

III.3 Questions ?

Ass. Prof. Jean-Yves Tigli
www.tigli.fr
tigli@polytech.unice.fr

III.4 Historical References

- Mark Weiser. "The Computer for the 21th Century." Scientific American, September 1991.
- Mark Weiser. "Some computer science issues in ubiquitous computing." Communications of the ACM, 36(7):75-85, July 1993.
- Mark Weiser, John S. Brown. "The Coming Age of Calm Technology." 1996.
- M. Satyanarayanan. "Fundamental Challenges in Mobile Computing." Fifteenth ACM Symposium on Principles of Distributed Computing, May 1996.
- M. Satyanarayanan. "Pervasive Computing: Vision and Challenges." IEEE Personal Communications, August, 2001.

III.5 WComp References

- J.-Y. Tigli, S. Lavirotte, G. Rey, V. Hourdin, D. Cheung, E. Callegari, M. Riveill "WComp middleware for ubiquitous computing: Aspects and composite event-based Web services" in the journal *Annals of Telecommunications*, Springer Paris editor, ISSN 0003-4347 (Print) 1958-9395 (Online), Vol. 64, No 3-4, March-April 2009
- Vincent Hourdin, Jean-Yves Tigli, Stéphane Lavirotte, Gaëtan Rey, Michel Riveill, "SLCA, Composite Services for Ubiquitous Computing", in International Conference on Mobile Technology, Applications and Systems, Sep 2008.
- Daniel Cheung-Foo-Wo, Jean-Yves Tigli, Stéphane Lavirotte et Michel Riveill. « Self-adaptation of event-driven component-oriented Middleware using Aspects of Assembly ». Dans 5th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC), California, USA, novembre 2007.