



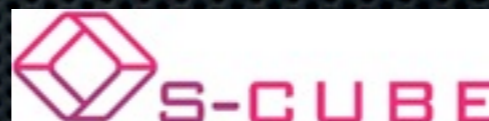
EnTiMid

A flexible component model
to integrate smart devices
in service-based applications
for ambient assisted living

Advisor : Jean-Marc Jézéquel
Co-Advisor : Olivier Barais

Reviewer : Elisabetta Di Nitto
Reviewer : Didier Donsez

Member : Daniel Thouroude
Member : Romain Rouvoy



- ⚠ Aging of the European population
- ☎ More needs in health care and assistance
- ⚠ Limited resources in Hospital and care centers
 - ⚠ Rooms and Places
 - ⚠ Manpower

- ✓ Promote cares at home
 - ✓ People prefer to stay at home
 - ✓ Shorter stays



 How to secure and help people at home ?

- ✓ Home automation: automation of tasks, support of chores
- ✓ Huge number of technologies

Media

Gateways

Transport Protocols

Devices

Bus
Radio
Infrared
PLC

RS232
RS484
TCP/IP

IOBL X2D BackNet X10 6lowPan
KNX DALI OWN ZigBee Zwave
io-Homecontrol LonWork

iPod WiiMote
SunSpot Temper
Android Nabztag
3G modem (SMS)

 How to support carers in their work ?

- ✓ Services through Internet
- ✓ New services created regularly

Skype(SMS) UPnP Vigicrue Google Talk Flickr REST Google Agenda
Google Document RTFE Weather.com Mail XMPP Facebook DPWS

- ⚠ 1 person = 1 solution = customization
 - ⚠ Pathologie(s)
 - ⚠ Disabilities
 - ⚠ User preferences
 - ⚠ Technical specificities



Human in the loop



Accessibility, Acceptability



Confidence in the system



Picture from <http://www.appt.gc.ca>



Human in the loop



Accessibility, Acceptability



Confidence in the system



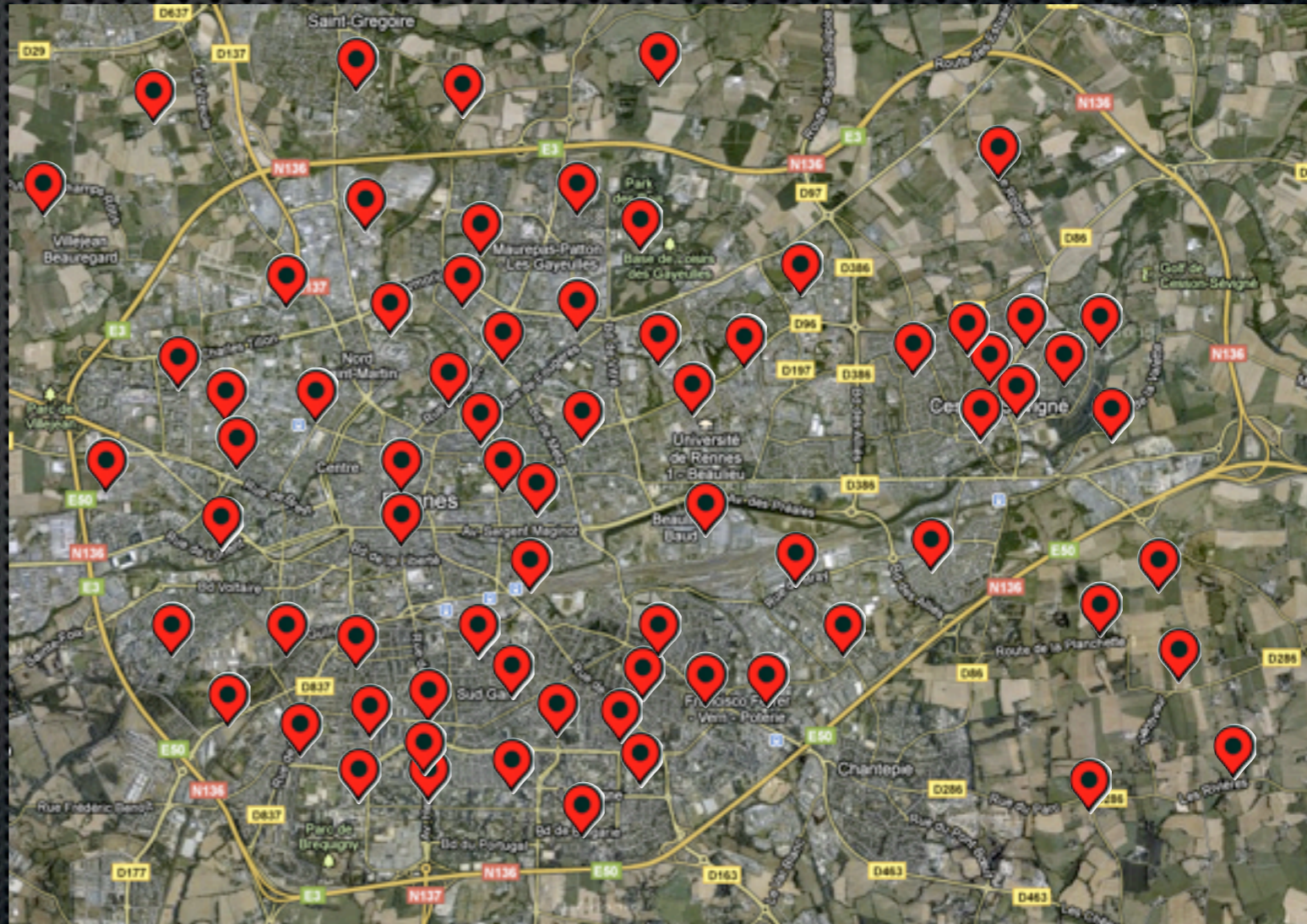
Space in the loop



Variability management tools
(versions, products, etc.)



Remote control and
administration





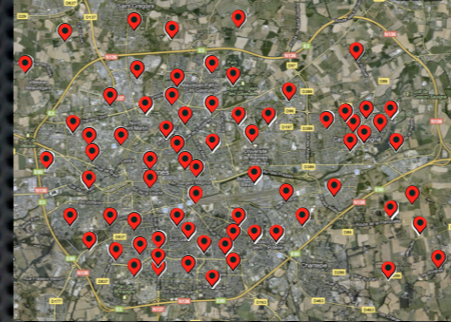
Human in the loop



Accessibility, Acceptability



Confidence in the system



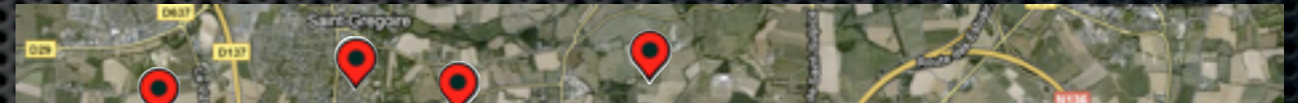
Space in the loop



Variability management tools (versions, products, etc.)



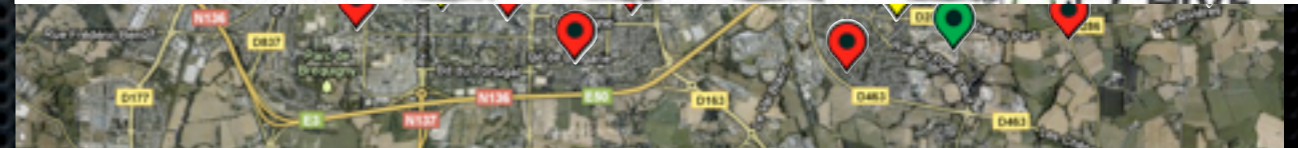
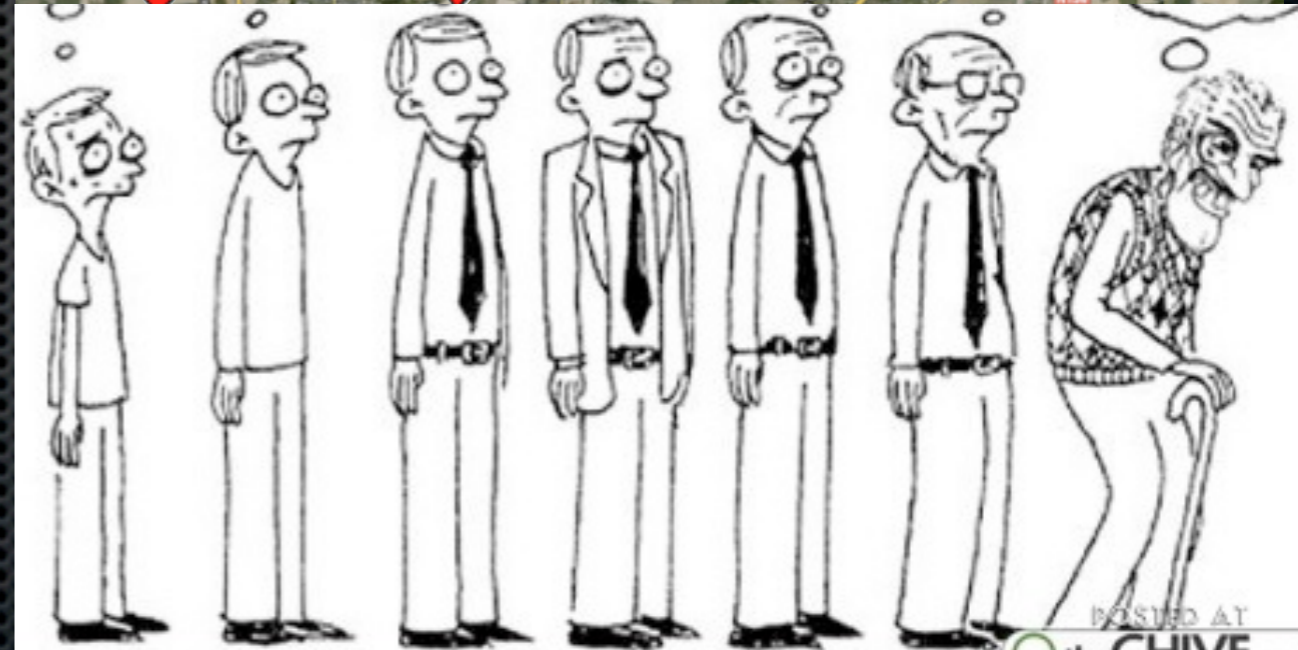
Remote control and administration



Time in the loop



Adaptation to evolution of pathology, of technology, of user preferences





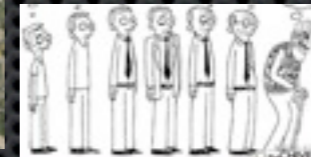
Human in the loop



Accessibility, Acceptability



Confidence in the system



Space in the loop



Variability management tools (versions, products, etc.)



Remote control and administration



Time in the loop



Adaptation to evolution of pathology, of technology, of user preferences



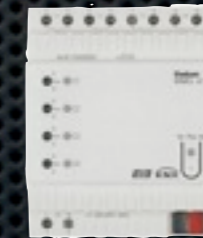
Variability in the loop





Interoperability, device-service integration



Openness, connection of heterogenous systems



-  Hardware not generic enough and/or not customizable for each demand.
-  Customization at the software level ?

Requirements

-  A software engineering approach to create on-demand customized solutions
 -  Very flexible for Interoperability and Openness
 -  Supports adaptation and evolution, even after deployment
 -  Eases the management of Variability


Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

Survey - CBSE - SOA - CM4SOA - Synthesis

		Interoperability	Openness	Adaptation	Evolution	Variability Management	Safety & Security
Generic Approaches	Darwin [GMK02]			+	+		+
	Koala [RvdLKM00]				+	+	+
	Fractal [BCL+06]			+			
	OSGi [All11]		+	+	+		
	ESB [Cha04]	+	+		+		
	SCA [sca]		+		+		+
	FraSCAti [MRRS10]	+	+	+	+		+
	iPojo [EHL07]		+	+	+		
Domain-Specific Approaches	uMiddle [NT07]						
	Soprano [WSO+10]			+	+		
	Gaïa [RHC+02]	+		+	+		
	Dia Suite [CBC10]	+	+			+	+
	Habitation [JRS+09]	+				+	
	WADL [CDT08]	+		+	+		
	PervML [MPC06]	+	+	+	+	+	
	AutoHome [BDLM11]		+	+	+		
	WComp [FHL+11]		+	+	+		
	Niagara [Tri08]	+	+				

Component-Based Software Engineering

Darwin [GMK02], Koala [RvdLKM00], Factal [BCL+06]

-  Great level of abstraction for Home Automation devices
-  Explicit architecture
-  Reconfiguration
-  Verifications on architecture consistency

 Interfaces of components (services, ports, etc.) are strongly linked to components' implementations.






 Independence between model / runtime

 Openness for third party connexions

 Dynamic adaptations with new component types

Service-Oriented Architecture

OSGi [All11], ESB[Cha04]

-  Adaptation and evolutions while running
-  Openness through the paradigm of service
-  Distribution
-  Lack of architectural description / reflexion
-  Services are defined by fixed interfaces.
No flexibility.

Component Model for Service-Oriented Architecture

SCA [SCA], FraSCAti [MRRRS10], iPojo [EHL07]

 Adaptation and evolution while running

 Architecture description

 Openness

 Distribution

 Independence of the model (reflexion)


 Ports/Services defined by fixed interfaces.
No flexibility.

Component Models for SOA

 How to improve the flexibility ?

 How to let designers choose between strict and relaxed checkers ?

Electronic components

 Can always be connected, regardless of their technical constraints

 Assemblies are modeled and simulated before welding, to prevent predictable failures



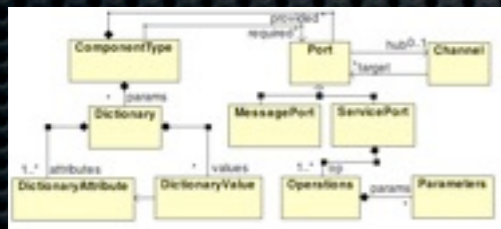
Contribution

- ❓ Decouple the typing of a component and move it from the implementation to the model
 - ❓ Mapping between model and implementation
- ❓ A set of tools to ease the component management, and check assemblies conformance
 - ❓ Adaptable checkers, flexible type verifications
- ❓ A runtime to execute this new component model
 - ❓ Independent model : Model@Runtime

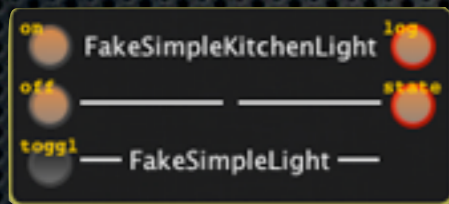
? Software components

Component.class

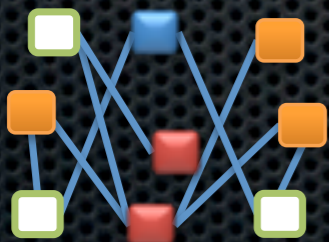
Component : Elementary piece of software that provides functionalities to other components, and can use other components' functionalities



Component Model : defines how a component must be described, and how components can be assembled (may include constraints)



Model of a Component : abstractly describes a type of component; conforms to the component model



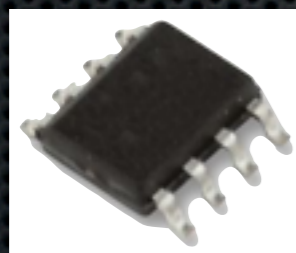
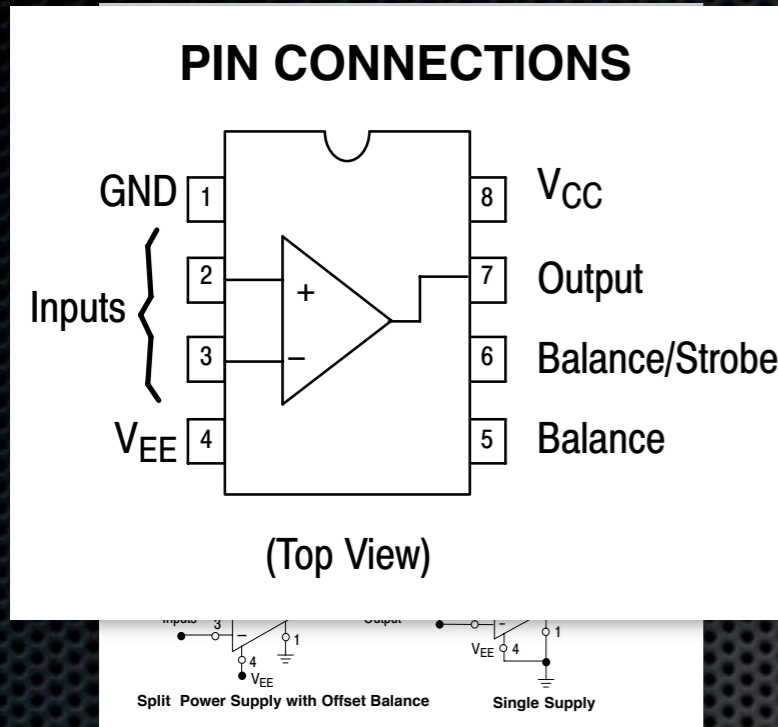
Model of a System : describes the component instances involved in the system and their connections/interactions

Electronics

Software

Component Model

Component Implementation



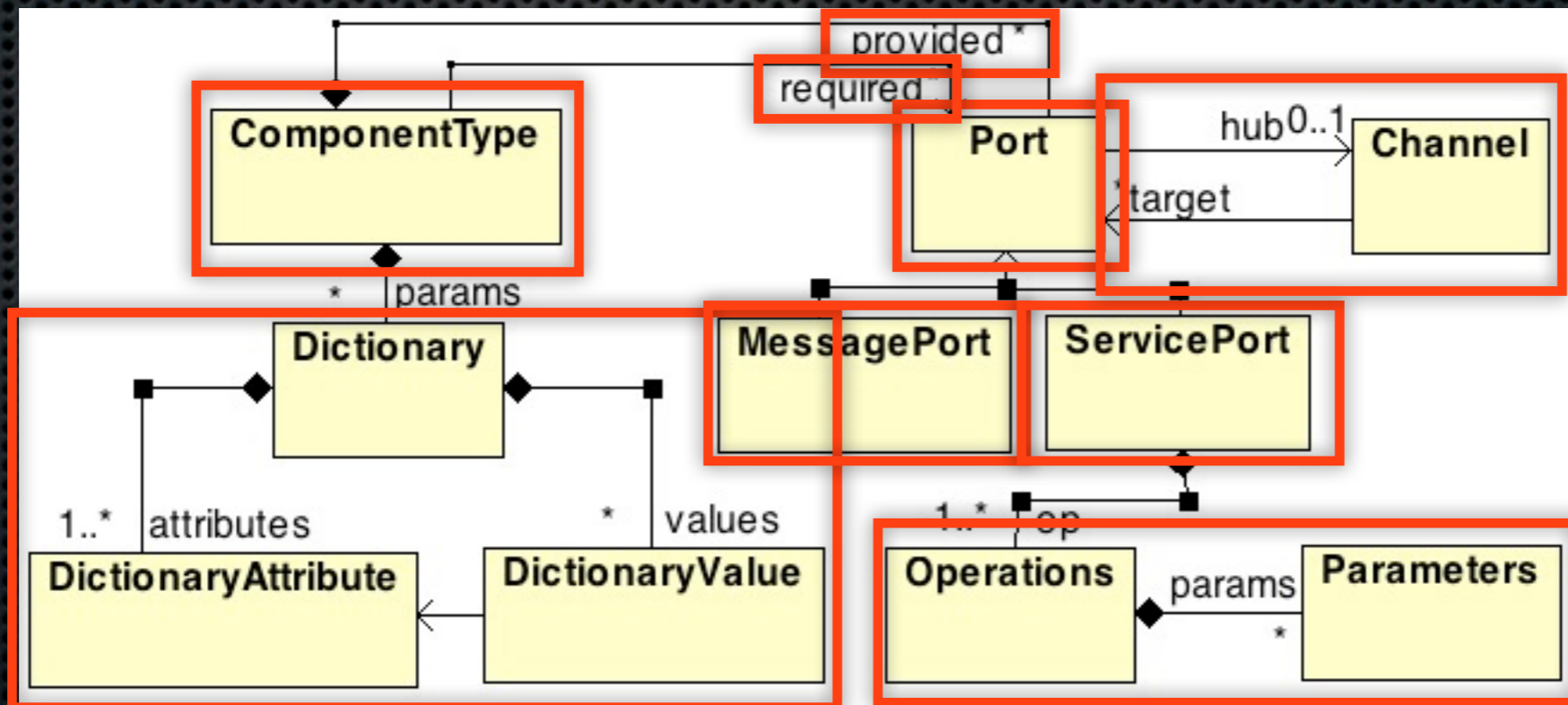
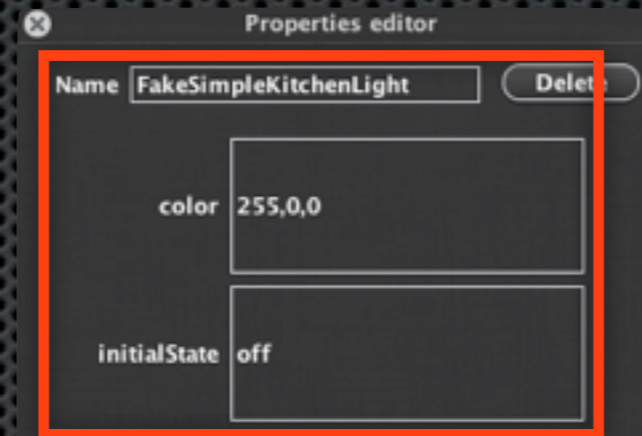
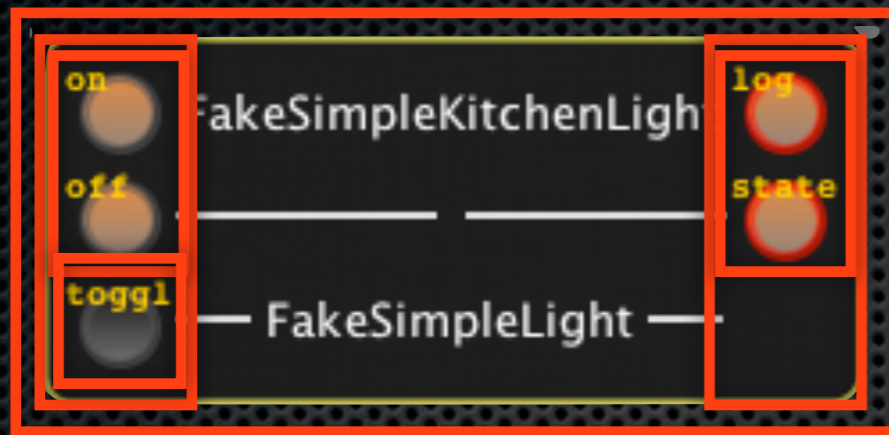
LM211

```
<?xml version="1.0" encoding="ASCII"?>
<art2:ContainerRoot xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:art2="http://art/
2.0">
  <typeD
  deployU
  factoryB
  bean="o
  @deploy
  <requ
  <requ
  <requ
  </type
  </type
  <typeDefinitions xsi:type="art2:MessagePort Type"
  name="org.kermeta.art2.framework.MessagePort" />
  <typeDefinitions xsi:type="art2:ServicePort Type"
  name="org.entimid.framework.service.OnOffService">
    <operations name="on" returnType="//@dataTypes.0" />
    <operations name="off" returnType="//@dataTypes.0" />
  </typeDefinitions>
  </art2:ContainerRoot>
```

```
oleanvalueOf(Z)Ljava/lang/Boolean;getDictionary()Ljava/util/HashMap;java/util/
HashMapkeySet()Ljava/util/Set;java/util/Setiterator()va/util/
Iterator;hasNext()Znext()Ljava/lang/Object;java/lang/SystemoutLjava/io/
PrintStream;append-(Ljava/lang/String;)Ljava/lang/StringBuilder;get&(Ljava/lang/
Object;)Ljava/lang/Object;-(Ljava/lang/Object;)Ljava/lang/
StringBuilder;toString()Ljava/lang/String;java/io/rintStreamprintln(Ljava/lang/
String;)Vjava/lang/ObjectgetClass()Ljava/lang/Class;java/lang/ClassgetNamejava/
util/logging/Logger getLogger(Ljava/lang/String;)Ljava/util/logging/Logger;java/
awt/ColorREDLjava/awt/Color;(Ljava/awt/Color;)V
```

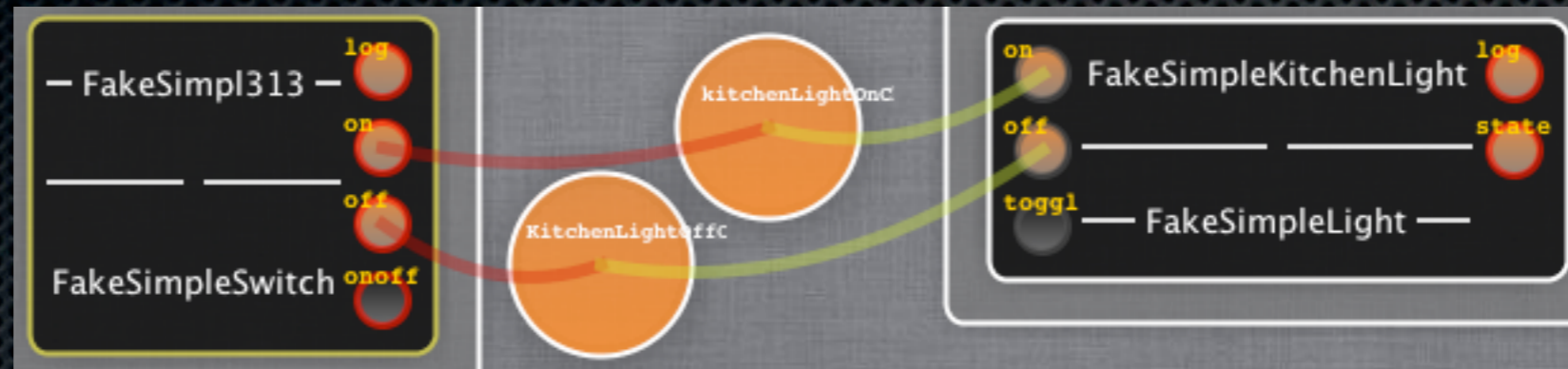
FakeSimpleLight.class

Classical structure



Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

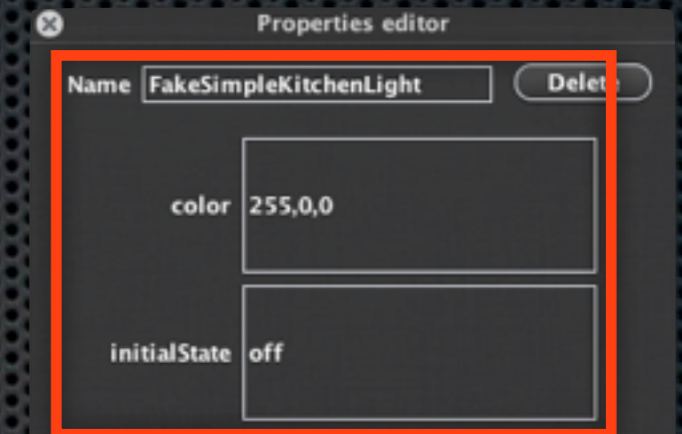
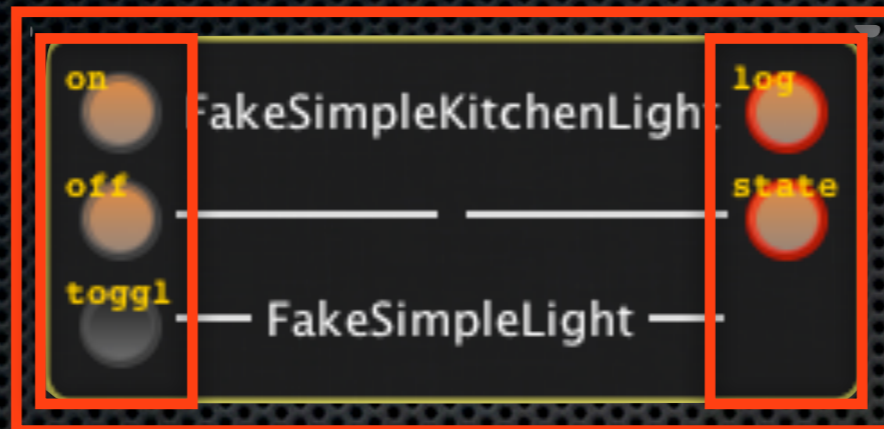
Overview - Component Model - Software tools - Outcomes



Channels : explicit and manages the communications between components

- ✓ More flexibility (can be changed)
- ✓ Media-independent
- ✓ Synchronous/Asynchronous
- ✓ Can perform some parameters alignments

Decoupling



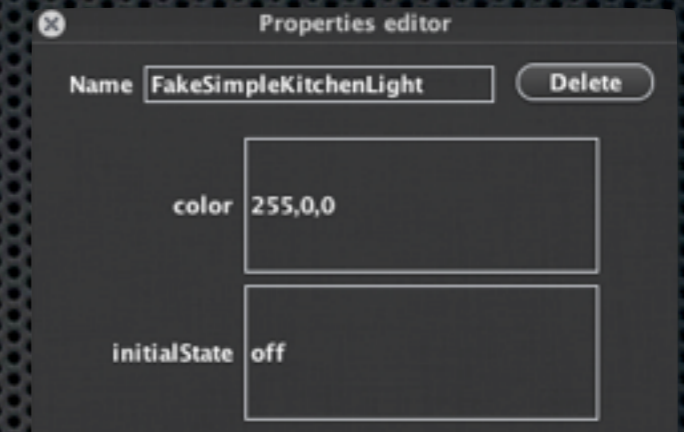
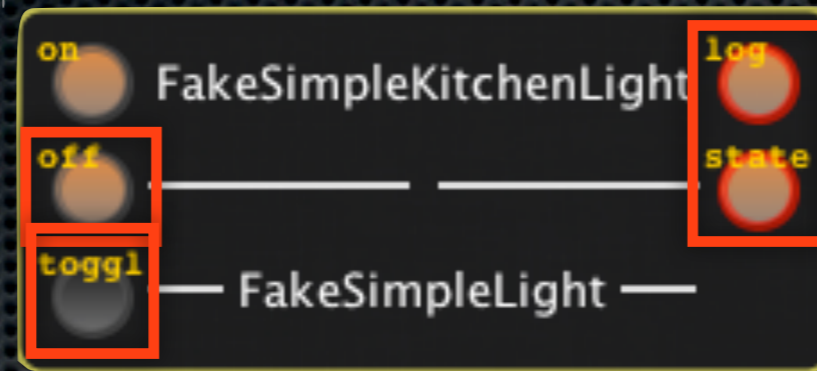
```
@Provides({
    @ProvidedPort(name = "on", type = PortType.MESSAGE),
    @ProvidedPort(name = "off", type = PortType.MESSAGE),
    @ProvidedPort(name = "toggle", className = OnOffService.class)
})
```

```
@Requires({
    @RequiredPort(name = "state", type=PortType.MESSAGE, optional = true)
})
```

```
@DictionaryType({
    @DictionaryAttribute(name = "color"),
    @DictionaryAttribute(name = "initialState", defaultValue = "off", optional = true)
})
```

```
@ComponentType
public class FakeSimpleLight extends AbstractFakeStuffComponent {
```

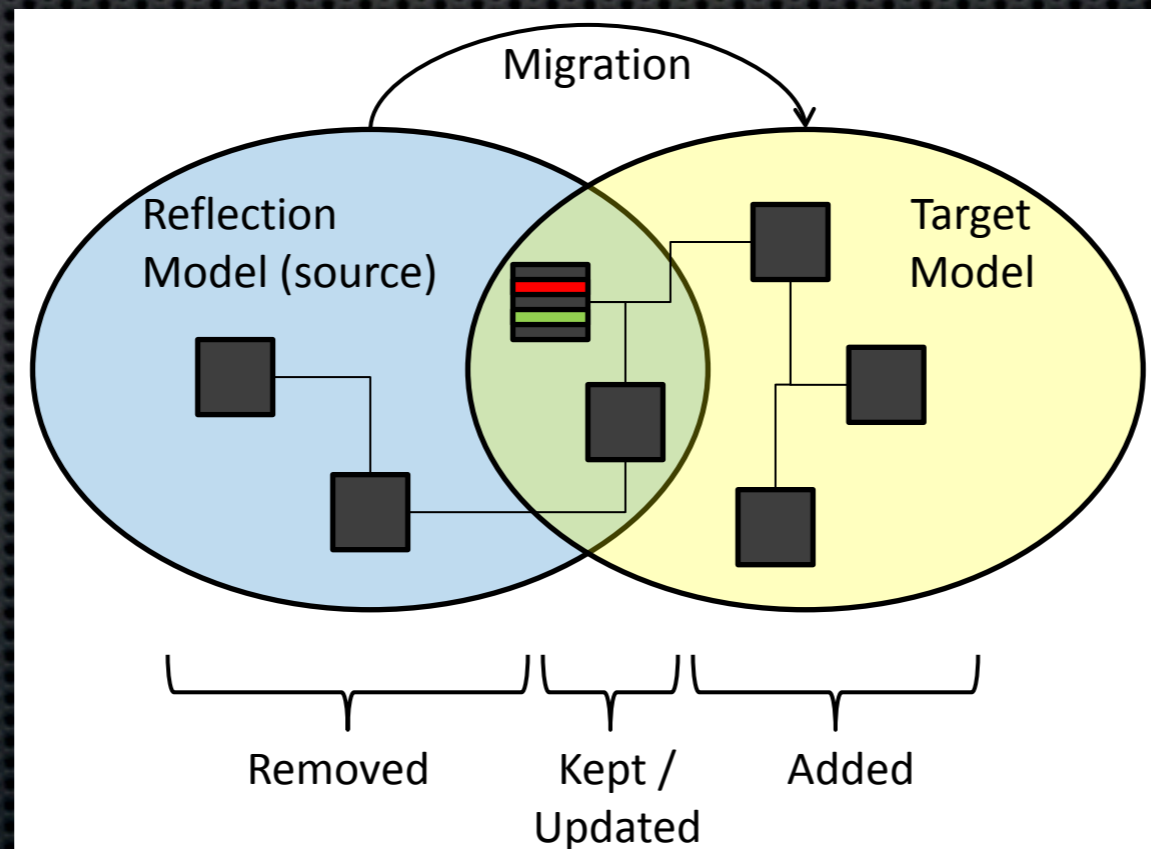
Mapping



```
@Ports({
|   @Port(name = "off", method = "process"),
|   @Port(name = "toggle", method = "off")
| })
public void lightOff(Message o) {
|   o.put(getName() + ":state", "off");
|   MessagePort log = getPortByName("log", MessagePort.class);
|   if (log != null) {
|       |   log.process(o);
|   }
|   MessagePort statePort = getPortByName("state", MessagePort.class);
|   if (statePort != null) {
|       |   statePort.process(o);
|   }
| }
```

- ✓ Good practice : prefer using Message Ports
- ✓ Dictionnaires of <key,value> easier to align than fixed interfaces of services
- ✓ Benefits [NFM+10]
 - ✓ Model of the component different from the actual implementation => flexibility
 - ✓ A MessagePort and a method of a service can be mapped on the same implementation method => reuse, capitalization

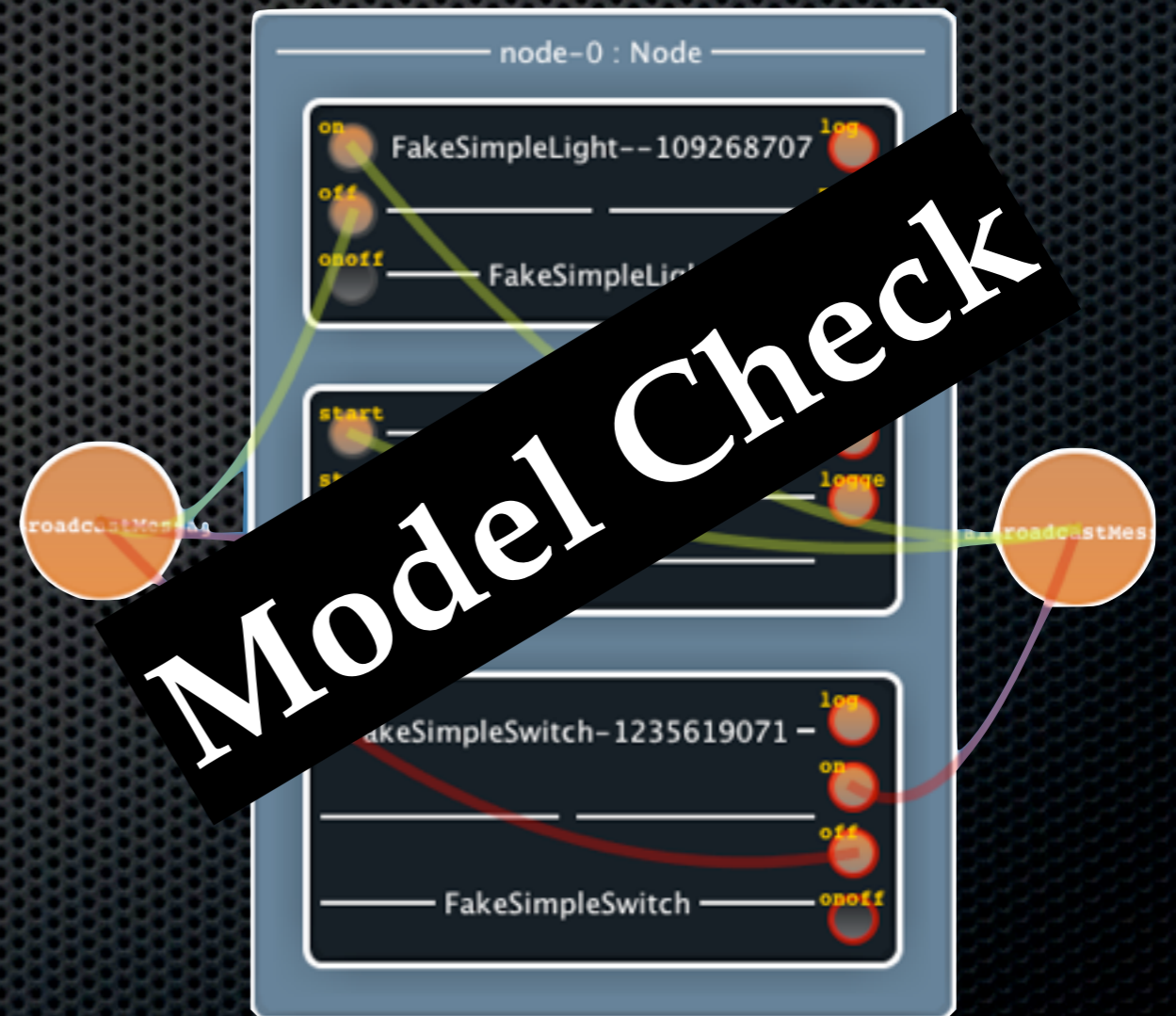
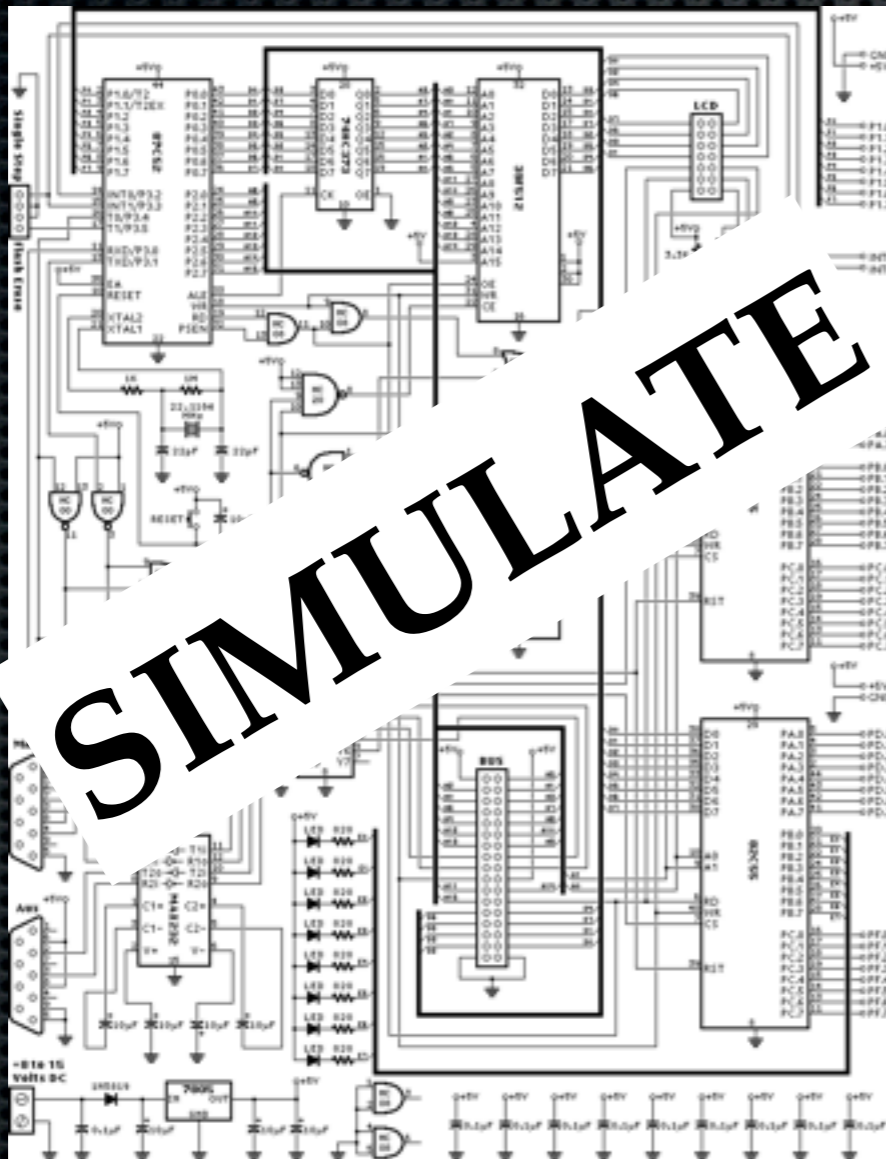
- ✓ Model @ Runtime [ICSE'09]
- ✓ Keep synchronized an independent model of the system
- ✓ Reasoning, checks can be performed at runtime with no impact on the running system
- ✓ Drives the adaptations at runtime



Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

Overview - Component Model - Software tools - Outcomes

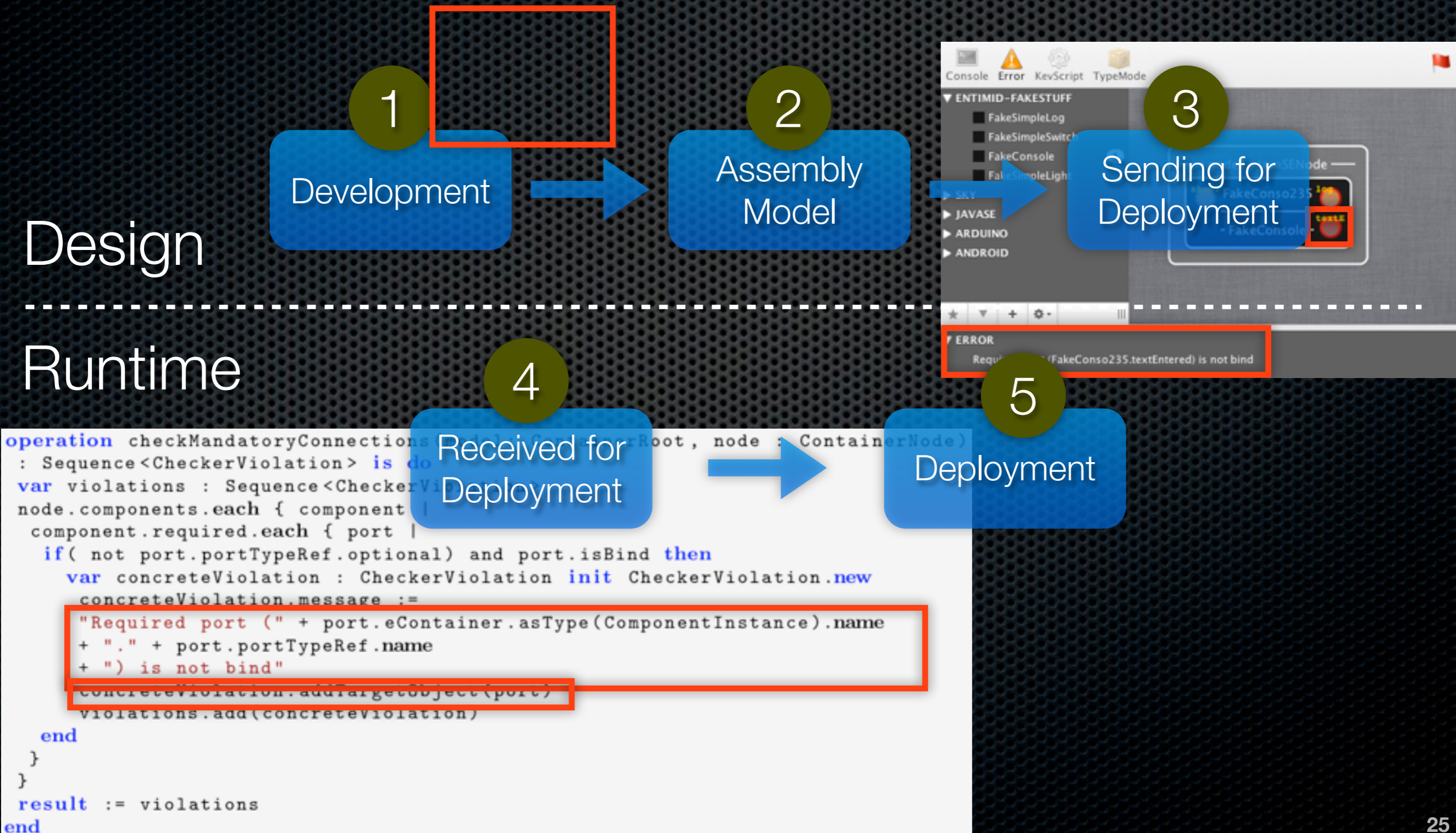
Pluggable checkers



Pluggable checkers

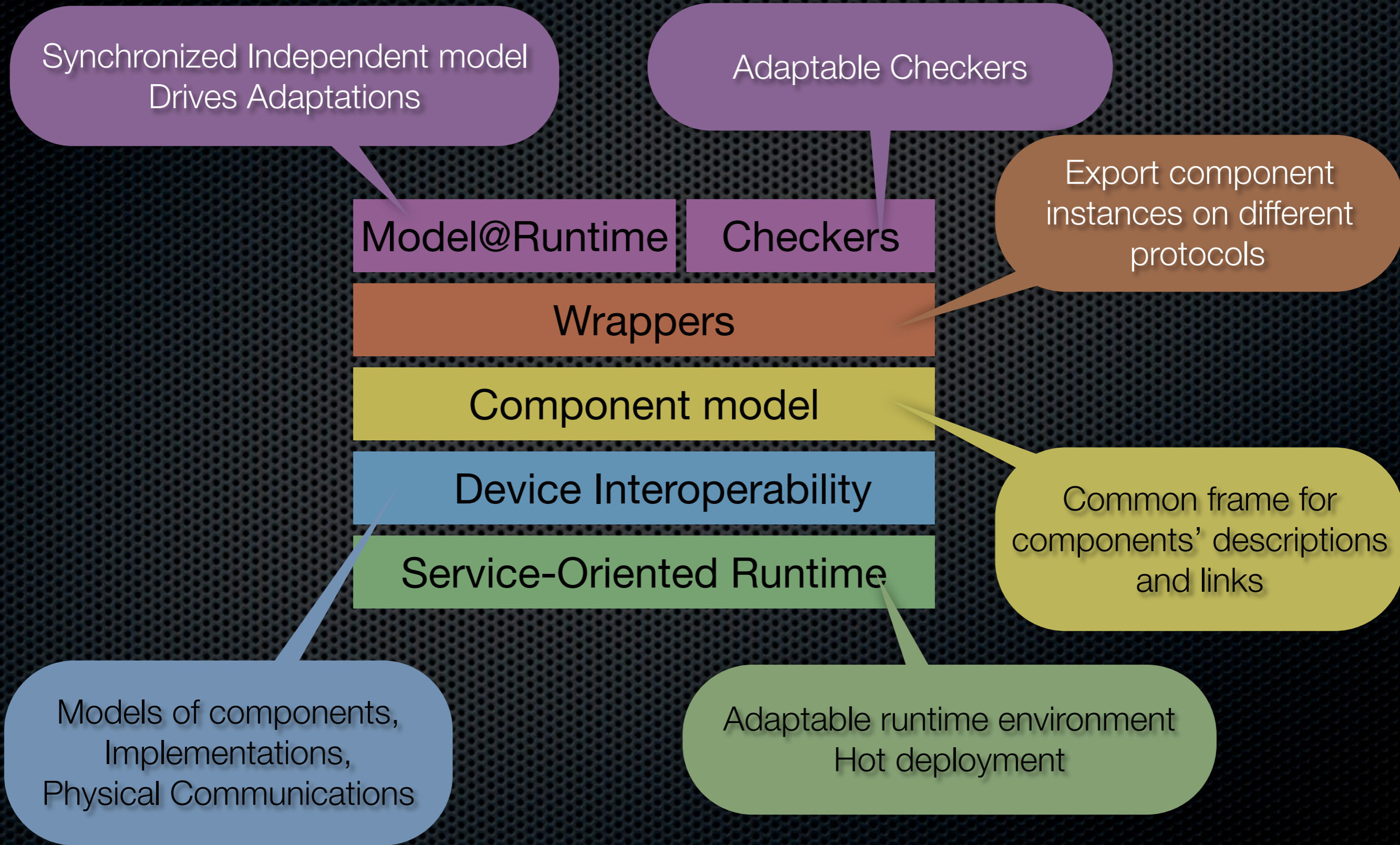
Design

Runtime



- ✓ Extraction of model from the code (CodeToModel)
 - ✓ Java Annotation Parser process
 - ✓ Integrated in the compilation phase (Maven)
 - ✓ Model embedded with compiled code

- ✓ From Model to Code
 - ✓ Java AST Parser/PrettyPrinter (Javaparser project on Google Code)
 - ✓ Generation of the component skeleton
 - ✓ Update of annotations and mappings



- ✦ A runtime has been developed on top of an OSGi platform
- ✦ EnTiMid, specific for Home Automation and AAL, generalized in Kevoree to be usable in different contexts
- ✦ ~22% de participation sur Kevoree, ~83% sur EnTiMid

Project	Scala	Java	Total	Part	Personal Total
Kevoree-Core	8143	344	8487	11,57 %	979,3
Kevoree-Tools	8483	6522	14738	22,22 %	3274,65
Kevoree-Extra (Ecore)	1275	126	1401	50 %	700,5
EnTiMid-Core	0	445	445	85 %	378,25
EnTiMid-Tools	82	315	3197	75 %	2397,75
EnTiMid-Library	0	2065	2065	80 %	1652
EnTiMid-Extra	0	1091	1091	85 %	927,35

Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

Context - Experimental Setup - Tests - Conclusion


✓ Members : Industrials, Carers, Academics/Laboratories

✓ The case study : Definition

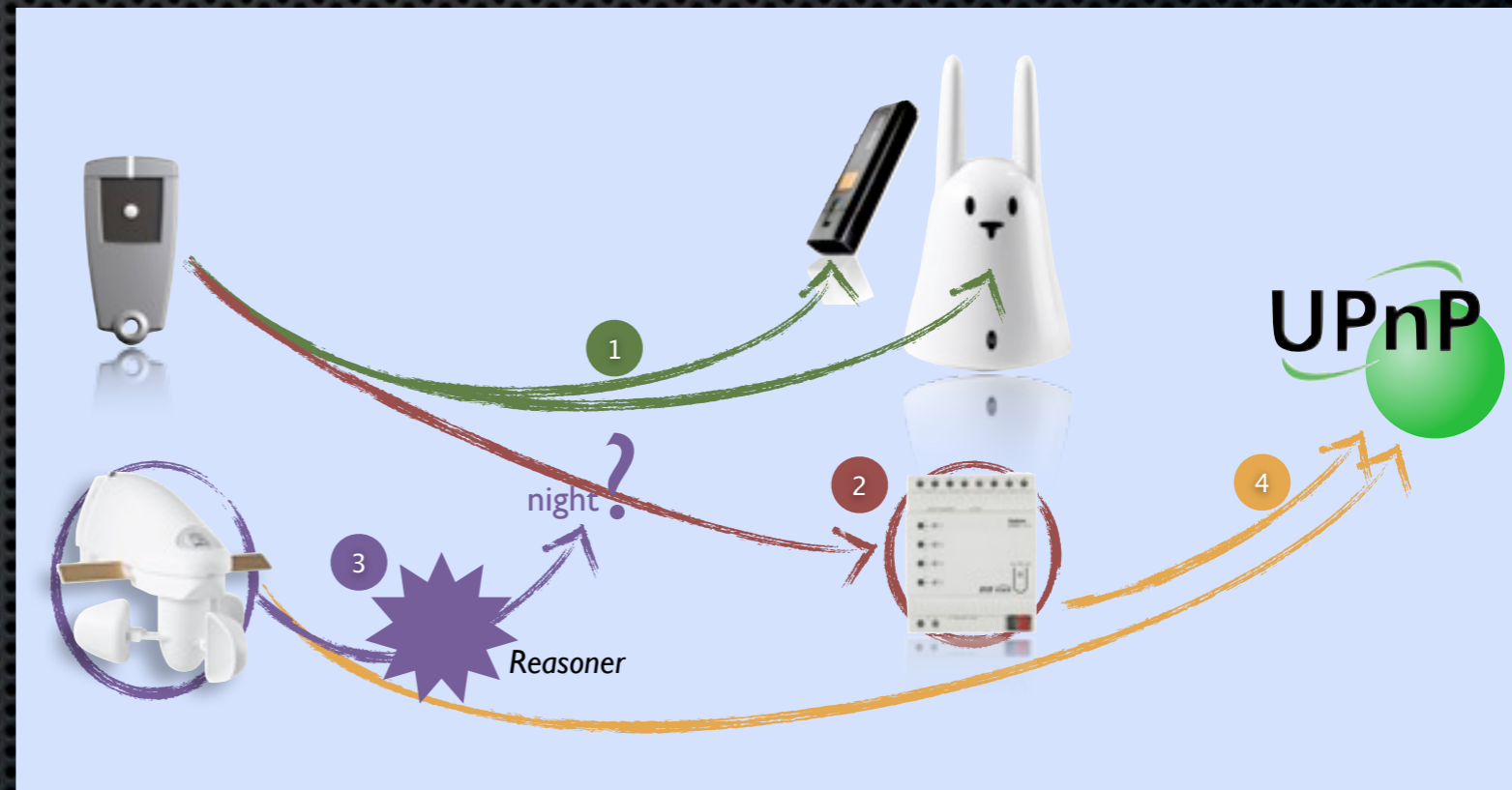


1  Interoperability

2  Evolution of user preference

3  Adaptation to the situation

4  Openness



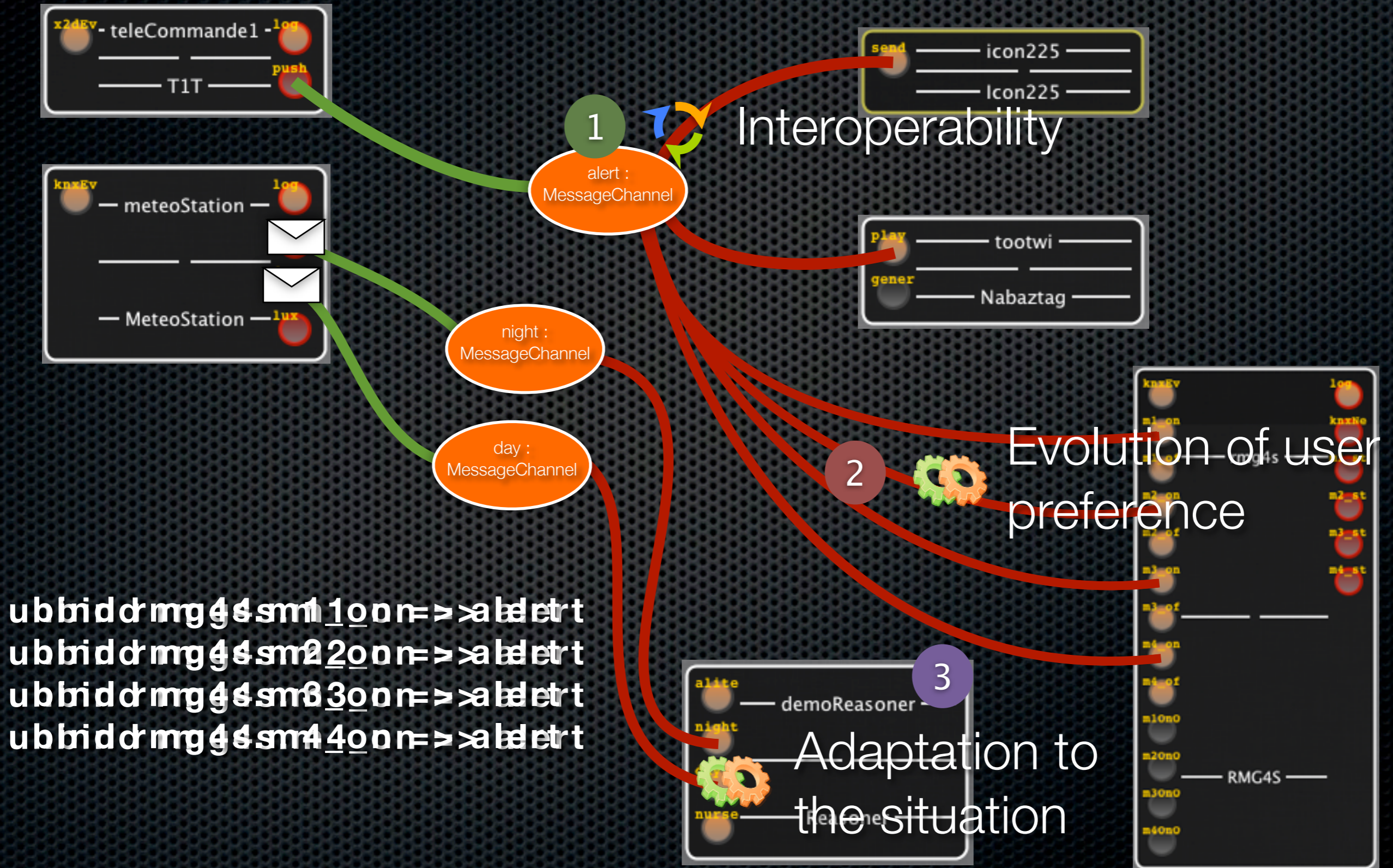
Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

Context - Experimental Setup - Tests - Conclusion

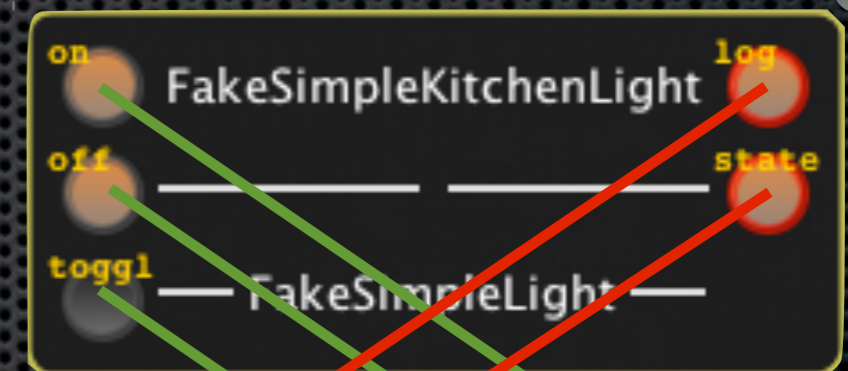
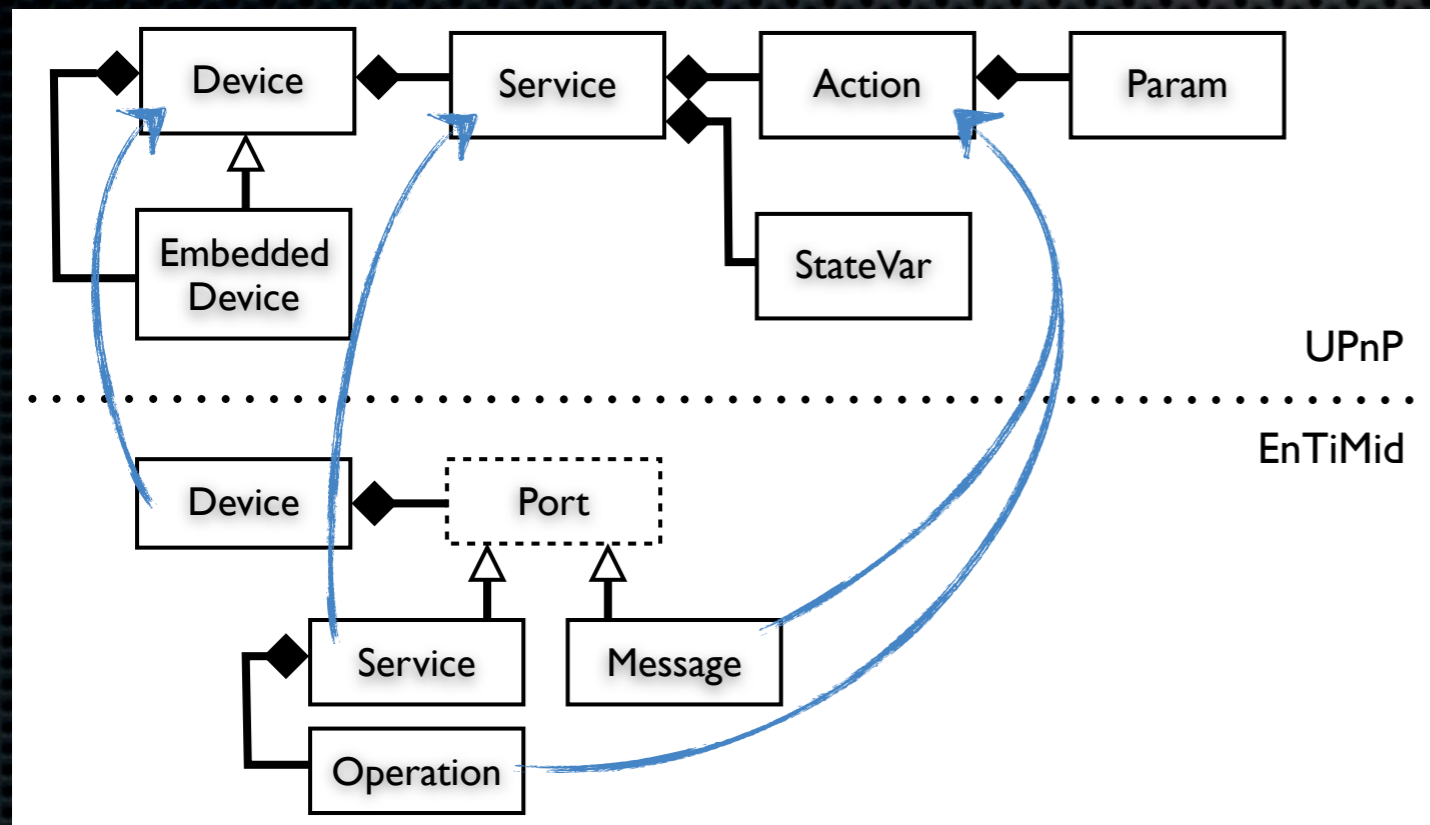
✓ Test environment

- ✓ Delta Dore
- Remote Control
- ✓ 3G Modem
- ✓ Nabaztag Rabbit
- ✓ Light controller (KNX)
- ✓ Meteo Station
- ✓ Touch Screen





4 Openness - UPnP Wrapper [NDBJ08]



- | UPnP-Actions | UPnP-StateVars |
|-----------------|----------------|
| - on | - state |
| - off | - log |
| - toggle-toggle | |


Introduction :: State-Of-The-Art :: Contribution :: Validation :: Conclusion

Context - Experimental Setup - Tests - Conclusion


Limitations

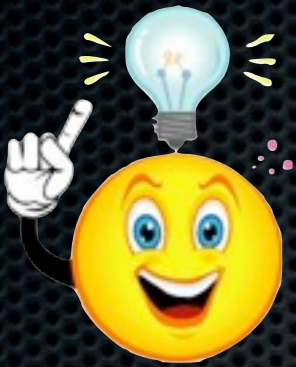
-  Safety & Security (partially addressed)
-  Scalability (cloud)
-  Validity of the scenario
-  Communication with smart devices
(requires a gateway)

 Decouple the component type and move it from the implementation to the model





 A set of tools to ease the component management, and check assemblies conformance

 A runtime to execute this new component model

 Approach validated on a scenario defined with caring professionals of the AAL



Research

-  Complete the tools with DSPL [INPJ09], Decision helping tools, Architecture Sythesis (CVL, etc.)
-  Distribution, Model@Runtime for Cloud infrastructure management (PhD Fouquet, PhD Daubert, PhD Mattos)
-  Automatic generation of Graphical User Interface [BBB⁺11]
-  End-User programing [CC09]

✓ Projects / Colaborations

- ✓ Daum : Improve and Support on-field information system of Firefighters
- 🔍 Collaboration projects with Kerpape, acceptability/accessibility, Resources/Interactions/Process models (UBS-Lorient)

✓ Business

- ✓ EnTiMid received fundings from the Britany region for maturation (FEDER)
- ✓ Neociti : Project of Spin-off (currently incubating), should start in January

Questions ?

- Using MDE to Build a Schizophrenic Middleware for Home/Building Automation
Grégory NAIN, Olivier BARAIS, Erwan DAUBERT, Jean-Marc JEZEQUEL
In ServiceWave'08: Networked European Software & Services Initiative (NESSI) Conference, December 2008, Madrid, Spain
- EntiMid : un middleware aux services de la maison
Grégory Nain, Olivier Barais, Régis Fleurquin, Jean-Marc Jézéquel
In Conférence francophone sur les architectures logicielles : CAL 2009, March 2009, Nancy, France
- Taming Dynamically Adaptive Systems with Models and Aspects.
Brice Morin, Olivier Barais, Grégory Nain, Jean-Marc Jézéquel
In 31st International Conference on Software Engineering (ICSE'09), Vancouver, Canada, May 2009.
- Dynamic Software Product Lines for Service-Based Systems
Paul Istoan, Grégory Nain, Gilles Perrouin, Jean-Marc Jézéquel
In 9th IEEE International Conference on Computer and Information Technology, Xiamen, China, Oct 2009.
- Leveraging Models From Design-time to Runtime. A Live Demo.
Brice Morin, Grégory Nain, Olivier Barais, and Jean-Marc Jézéquel.
In 4th International Workshop on Models@Run.Time (at MODELS'09), Denver, Colorado, USA, Oct 2009.
- Integrating IOT and IOS with a component-based approach.
Grégory Nain, François Fouquet, Brice Morin, Olivier Barais, and Jean-Marc Jézéquel.
In Proceedings of the 36th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2010), Lille, France, 2010.
- F4Plan: An Approach to build Efficient Adaptation Plans
Francoise Andre, Erwan Daubert, Grégory Nain, Brice Morin, and Olivier Barais
In Proceedings of the 7th International ICST Conference on Mobile and Ubiquitous Systems (MobiQuitous 2010), Sydney, Australia
- Combining Aspect-Oriented Modeling with Property-Based Reasoning to Improve User Interface Adaptation
Arnaud Blouin, Brice Morin, Olivier Beaudoux, Grégory Nain, Patrick Albers and Jean-Marc Jézéquel
EICS'11: Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems