PROS

Centro de Investigación en Métodos de Producción de Software

Centro de Investigación ProS Grupo TATAMI

La Computación Autónoma desde la Perspectiva de los m@rt

Joan Fons

Febrero de 2018

Red de Excelencia en ingeniería del Software Dirigida por Modelos (MDE)

UNIVERSIDAD POLITECNICA DE VALENCIA

PROS
Centro de Investigación en Métodos de Producción de Software

Presentación

UNIVERSIDAD POLITECNICA DE VALENCIA

https://tatami.dsic.upv.es/group

TaTAmI Group

Techniques and Tools for Ambient Intelligence

- Dr. Vicente Pelechano -
Director of TaTAMi Research Group, Professor

- Dr. Joan Fons -
Assistant Professor

- Dr. Manoli Albert -
Assistant Professor

- Dr. Pedro Valderas -
Assistant Professor

- Dr. Victoria Torres -
Associate Professor

- Dr. Miriam Gil -
Postdoc. Researcher

- Nacho Mansanet -
Researcher & Software Engineer

tatAmi

Presentación

- El grupo tatAmI ha liderado el desarrollo de **propuestas metodológicas** para modelado y desarrollo automático de Sistemas de Información, Aplicaciones Web, Aplicaciones Móviles, Sistemas Inteligencia Ambiental (Hogares, Edificios y Ciudades Inteligentes), en ámbito **Internet de las Cosas** y **Computación Autónoma**

- Hemos sido impulsores y '*early adopters*' (punto de vista práctico) soluciones en ámbito del **MDE/MDD**, los **Modelos en Tiempo de Ejecución** y la **IoT**

PROS
Centro de Investigación en Métodos de Producción de Software

Presentación

- Proyectos Nacionales y Europeos

  https://tatami.dsic.upv.es/group/projects.php

- Tesis Doctorales dirigidas

  https://tatami.dsic.upv.es/group/thesisSupervised.php

UNIVERSIDAD POLITECNICA DE VALENCIA

Presentación

- Herramientas (¿nuestra debilidad? 🤔)
  - Industriales

MOSKITT
MOdeling Software Kitt

  - Académicas (https://tatami.dsic.upv.es/group/tools.php)

mFM

MOSKITT 4ME

MOSKITT 4SPL

PervML
be pervasive

Model-based Reconfiguration Engine for Web Services

AdaptIO

EUCaliptool

VIVACE

PROTeus

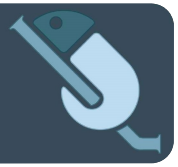
Objetivos del Curso

- Objetivo de este curso:

  Mostrar un ejemplo práctico de cómo un enfoque MDE usa M@RT para desarrollar sistemas con capacidades de computación autónoma

Centro de Investigación en Métodos de Producción de Software

Agenda

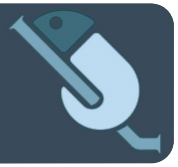


- Los Modelos en Tiempo de Ejecución
- La Computación Autónoma
- Ejemplos/Ámbitos de Aplicación

UNIVERSIDAD POLITECNICA DE VALENCIA

PROS
Centro de Investigación en Métodos de Producción de Software

Agenda



- Los Modelos en Tiempo de Ejecución
  - Principios
  - ¿Qué son?
  - ¿Qué nos aportan?
  - Rol en el Ciclo de Desarrollo de Software
  - Ejemplos de Aplicaciones

- **Los Modelos en Tiempo de Ejecución**
- La Computación Autónoma
- Caso de Estudio: Los vehículos autónomos
- Ejemplos y Ámbitos de Aplicación

UNIVERSIDAD POLITECNICA DE VALENCIA

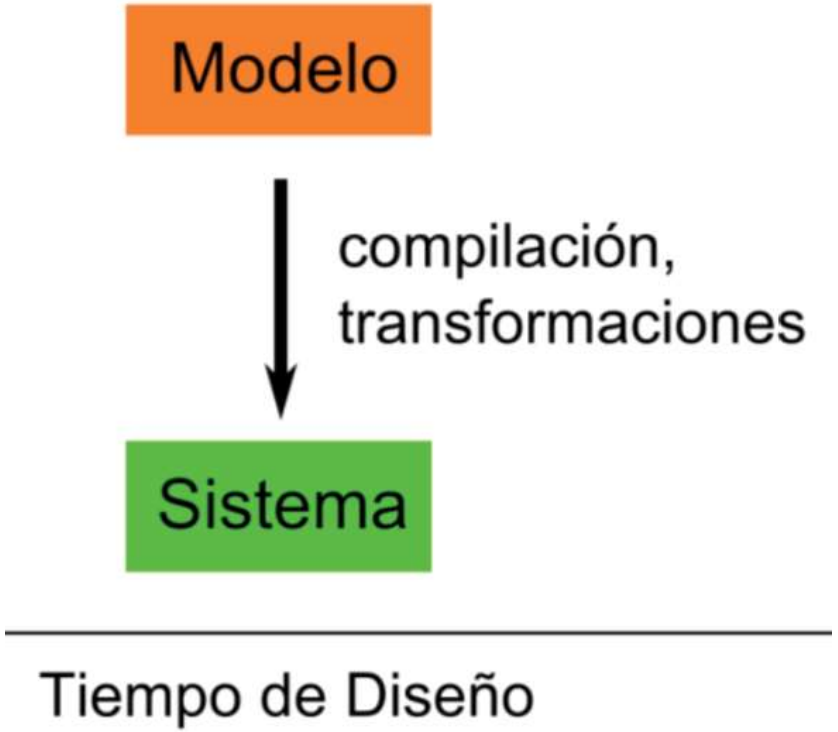
Los Modelos en Tiempo de Ejecución (m@rt)

Principios

- **Estrategia MDE**: construcción de software a partir de la conceptualización de sistemas mediante modelos y manipulación/refinamiento de éstos (transformaciones) hasta obtener sistema final
- Los **modelos** juegan un papel **clave** en este enfoque, y **dirigen** el proceso / progreso del **desarrollo** del software
- La utilización de modelos permite focalizar / **abstraer** (en cada momento) de detalles innecesarios (en este momento)
- Cada **transformación** nos acerca al producto final ...

Los Modelos en Tiempo de Ejecución (m@rt)

Principios



Los Modelos en Tiempo de Ejecución (m@rt)

Principios

- Tras un conjunto (variable) de transformaciones y refinamientos (M2M y M2T)*, de los modelos que describen el sistema ...
- ... obtenemos el producto final que es consecuencia de este proceso de manipulación de estos modelos ...
- ... pero que finalmente, este producto final pierde la conexión con estos modelos que lo han producido

* Las transformaciones pueden ser automáticas, (semi)automáticas o manuales

Los Modelos en Tiempo de Ejecución (m@rt)

Principios

- Entonces, si modelos son artefactos más valiosos ...
  - nos describen los diferentes aspectos de un sistema
  - permiten expresar un sistema como un conjunto de perspectivas
- ... con los cuáles se construyen los sistemas ...
  - a través de estas transformaciones y refinamientos
- ... ¿por qué deshacerse de ellos?
  - ¿es que no pueden aportar nada más allá de las fases de análisis, diseño, testing y despliegue?

Los Modelos en Tiempo de Ejecución (m@rt)

Principios

- La idea clave detrás de los Modelos en Tiempo de Ejecución es ...

... usar los Modelos

TAMBIÉN

en Tiempo de Ejecución

Los Modelos en Tiempo de Ejecución (m@rt)

¿Qué son?

- En la literatura encontraremos este término como

Models at runtime

models@runtime

o en su forma abreviada

m@rt

Los Modelos en Tiempo de Ejecución (m@rt)

¿Qué son?

- En un enfoque MDE, un modelo es ...

  una abstracción o representación simplificada de un sistema que se construye con propósitos específicos

Los Modelos en Tiempo de Ejecución (m@rt)

¿Qué son?

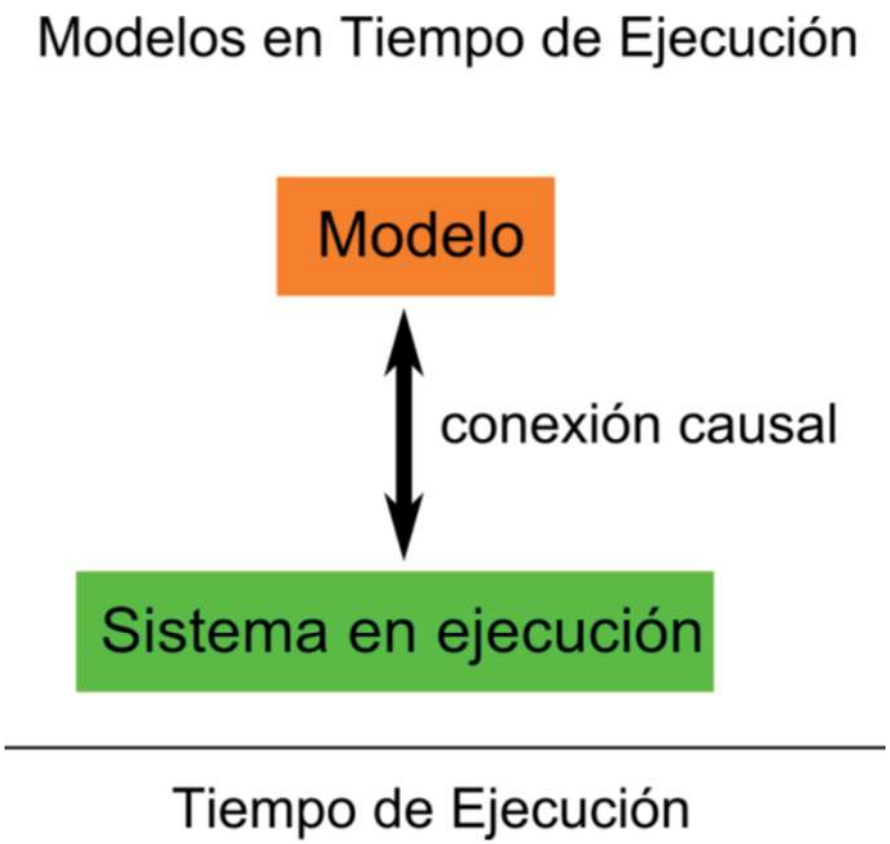
- En m@rt, a modelos se les asigna roles en ejecución

auto-representaciones ***causalmente conectadas**** *al sistema asociado que enfatizan la estructura, el comportamiento y los objetivos del sistema desde una perspectiva del espacio del problema*

* Si el sistema cambia, los modelos cambian (y viceversa)

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué son?

- Esta conexión causal entre modelo y sistema es clave:
  - Si hay *causa* en el sistema, hay un *efecto* en modelo
  - Si hay una *causa* en modelo, hay un *efecto* en el sistema



Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué son?

- Otro aspecto importante es que en m@rt, los **modelos** usados en tiempo de ejecución ...

  ... están **intrínsecamente ligados** a los modelos producidos como artefactos del proceso MDE

- Es decir, m@rt promueve **reúso** (tiempo ejecución) de modelos usados en ingeniería del sistema

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué son?

- ¿Qué constituye Modelo en Tiempo de Ejecución?

Un modelo puede ser cualquier ***representación útil***
del sistema que se pueda ***consultar y actualizar***
en ***tiempo de ejecución***

- Representación 'física' misma que modelos diseño
  - XML/XMI, CSV, RDF, Ficheros (Key/value), ...

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué son?

- Categorías de Modelos usados en *runtime*:
  - Estructural vs Comportamiento
  - Procedural vs Declarativo
  - Funcional vs No Funcional
  - Formal vs Informal

- En la práctica:
  - múltiples modelos coexisten para diferentes aspectos
  - pueden requerirse varios estilos de modelos para capturar un mismo aspecto

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué nos aportan?

- ¿Cuáles son los **objetivos** de los m@rt?
  - Dotar sistema en ejecución nuevas posibilidades
    - Proporcionar información sobre el propio sistema
    - Facilitar tareas de análisis y razonamiento
    - Verificar correctitud
    - Asistir en la toma de decisiones
    - ...

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué nos aportan?

- (Auto-) Conciencia: permiten dotar al software capacidades de **introspección** sobre sus fundamentos, de sus orígenes y objetivos sistémicos
  - Por ejemplo, el sistema podría 'argumentar' porqué se comporta de cierta manera en estos momentos

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué nos aportan?

- Semántica: permiten ofrecer un marco (conceptos, significado) que habilita el **análisis**, **razonamiento** y **entendimiento** en ejecución sobre el propio sistema
  - Por ejemplo, el sistema podría auto-monitorizarse y observar el comportamiento en ejecución con el fin de entender patrones de conducta, incluso plantear alguna acción

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué nos aportan?

- Correctitud: permiten validar la ejecución contra un esquema conceptual
  - Por ejemplo, estos modelos podrían ayudar a identificar errores de diseño, para que el sistema tratara de corregirlos, acorde a los objetivos sistémicos

Los Modelos en Tiempo de Ejecución (m@rt)
¿Qué nos aportan?

- Dinamismo: asisten al **cambio** (adaptación / evolución) del sistema, a través de cambios en estos modelos 'vivos' en tiempo de ejecución
  - Por ejemplo, estos modelos pueden servir para cambiar la configuración actual del software para tratar de adaptarse mejor a las condiciones de contexto (auto-adaptación)
  - O incluso podrían permitir la (auto-)generación automática de nuevos artefactos de implementación en tiempo de ejecución (auto-evolución)

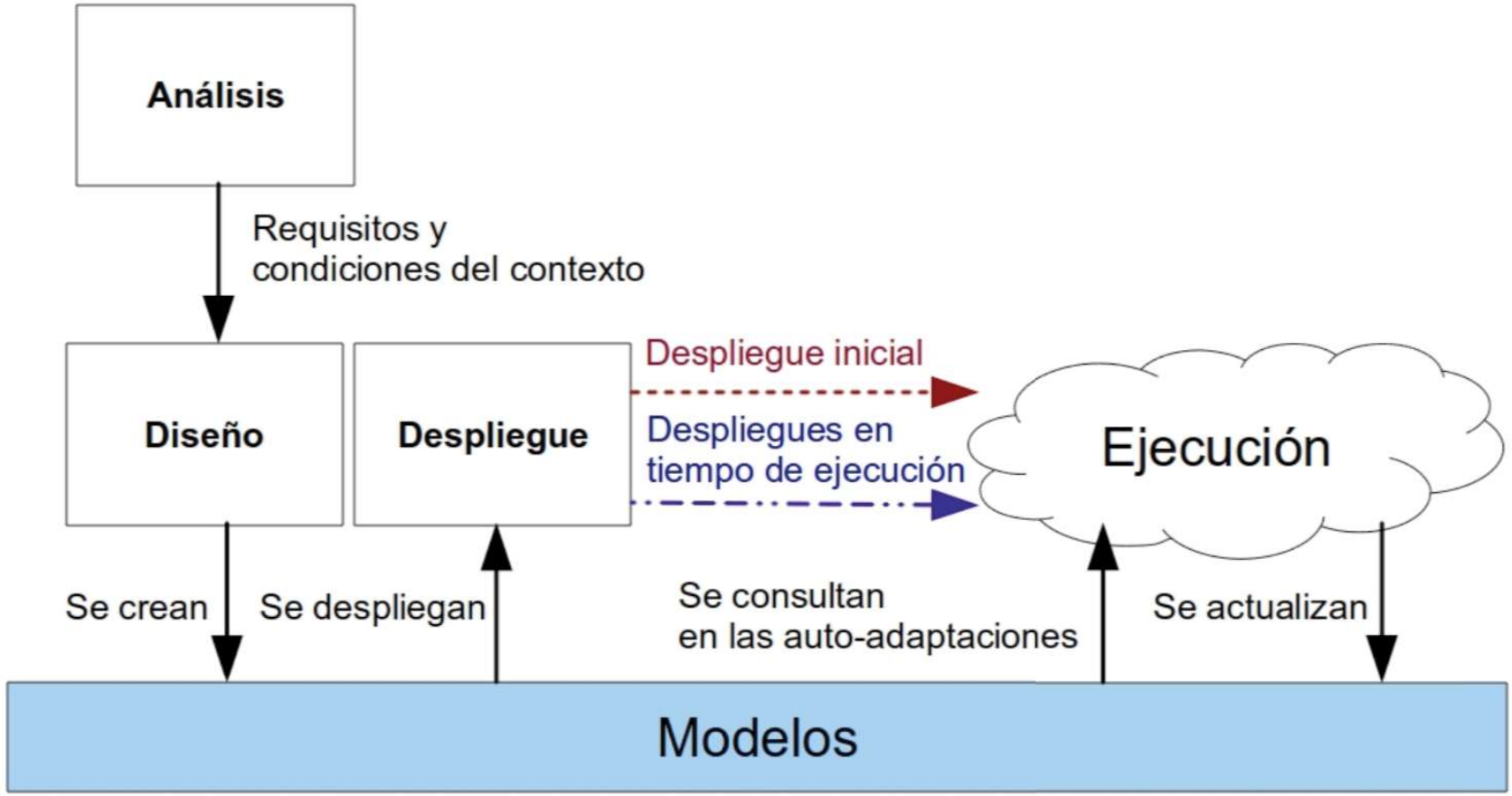
(Aspecto clave en la Computación Autónoma!!)

Los Modelos en Tiempo de Ejecución (m@rt)
Rol en el Ciclo de Desarrollo Software

- En una estrategia **m@rt**, la división entre las fases de diseño, despliegue y ejecución se difumina
  - Una vez un sistema está 'desplegado', pueden abordarse cambios y actualizaciones sobre el sistema en ejecución
  - Podemos mantener el sistema en ejecución en un bucle de refinamiento y actualización de sus modelos y componentes de implementación (¿evolución en runtime?)

Los Modelos en Tiempo de Ejecución (m@rt)

Rol en el Ciclo de Desarrollo Software



- Ejemplos de Aplicaciones
  - Hogares Inteligentes
  - Procesos de Negocio Físicos Móviles
  - Composición Autónoma de Servicios Web
  - Adaptación de Interfaces de Usuario

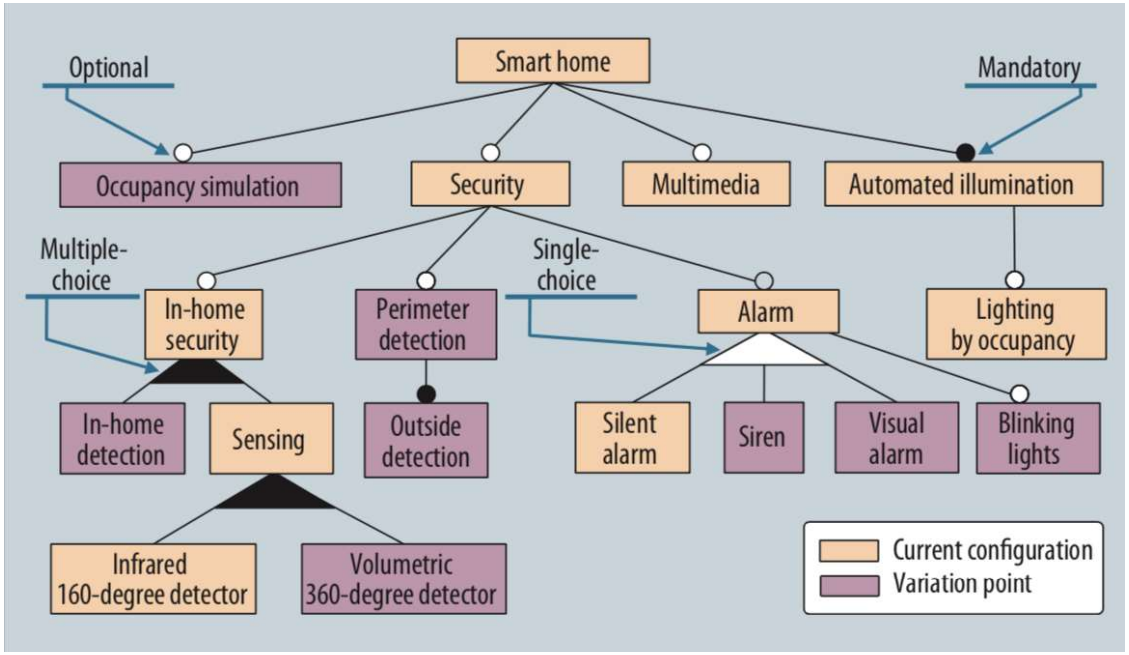
Los Modelos en Tiempo de Ejecución (m@rt)

Ejemplos de Aplicaciones

- Hogares Inteligentes
  - *PhD 'Achieving Autonomic Computing Through the Use of Variability Models at Run-Time' (Cetina, 2010)*
  - Un hogar inteligente reconfigura sus servicios y dispositivos en función de las actividades del usuario, aplicando políticas de eficiencia energética, seguridad y confort (entre otras)
  - Usa modelos/DSLs (PervML) que describen viviendas inteligentes, y Modelos de Variabilidad (MV) para representar potenciales variantes
  - Reutiliza estos modelos en tiempo de ejecución para dirigir la adaptación autónoma, cambiando la configuración actual (modelo de arquitectura de componentes PervML) de la vivienda, y causalmente, el sistema

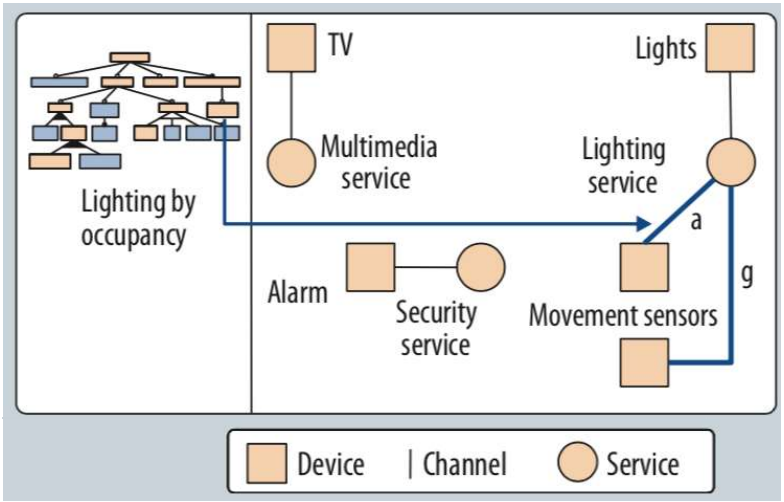
Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones

- Hogares Inteligentes



Descripción de Variantes del Sistema

Modelo Arquitectónico (PervML) de la Vivienda Inteligente. (Instancia) Configuración Actual (el usuario está en casa)

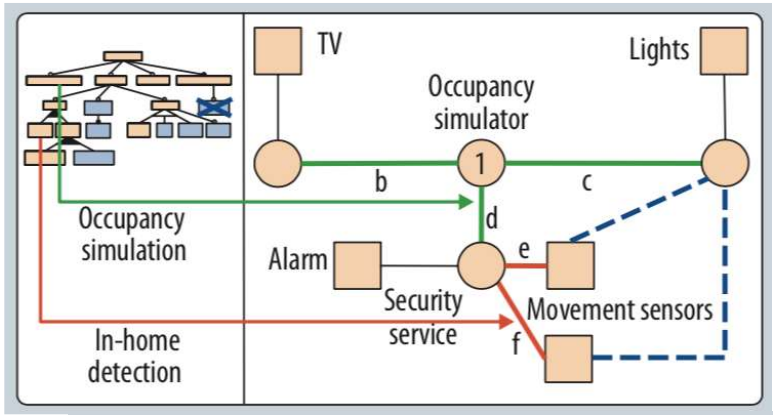


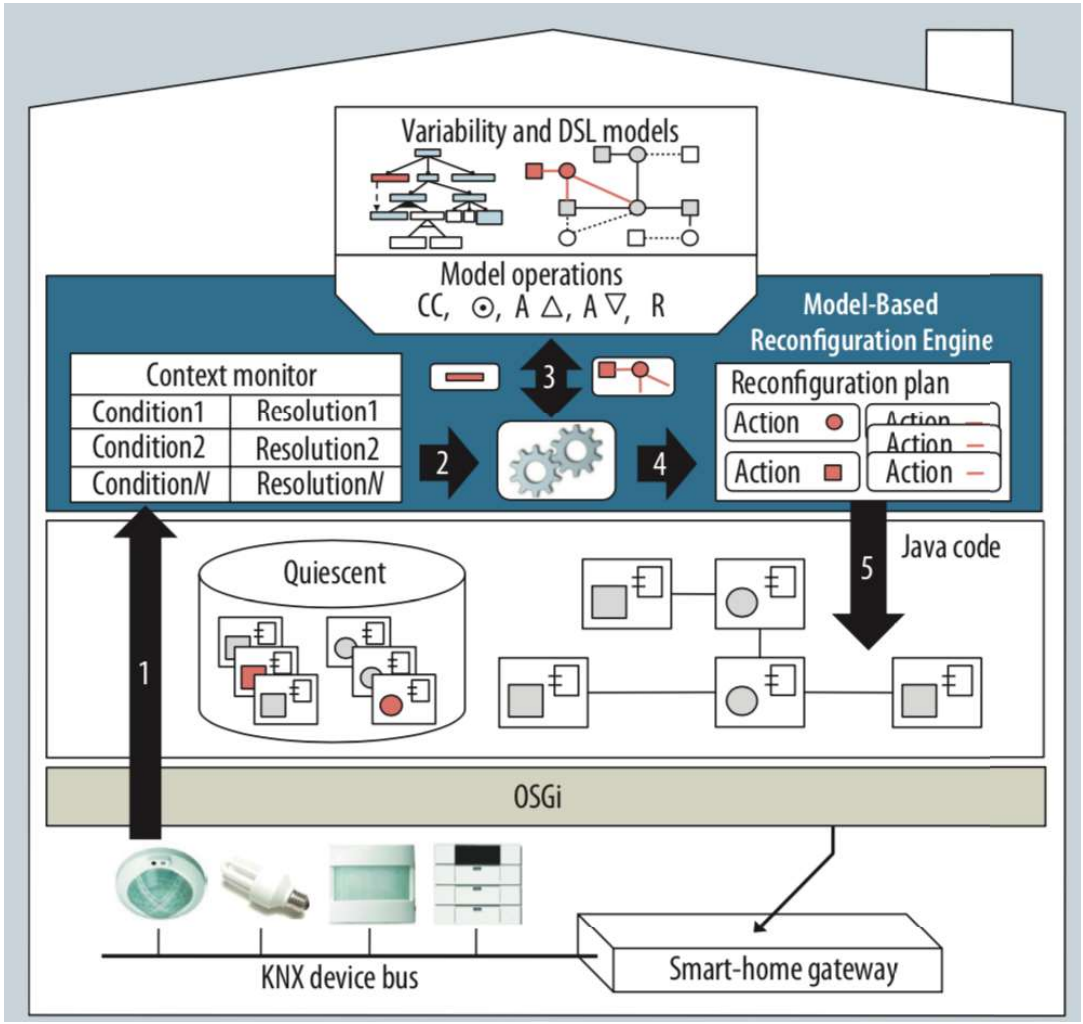
Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones

(1) Cuando ocurre un evento relevante en la vivienda *(el usuario sale de casa)* ...

(2) los modelos de variabilidad y DSLs son consultados, (3) manipulados y transformados para obtener la nueva

(4-5) configuración del sistema





nueva versión del modelo arquitectónico PervML

(la vivienda se configura en modo seguridad)

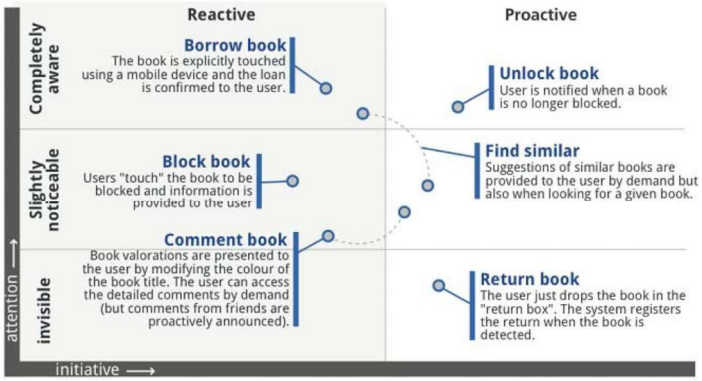
Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones

- Procesos de Negocio Físicos Móviles
  - *PhD 'Automating the Development of Physical Mobile Workflows' (Giner, 2010)*
  - Método Desarrollo de sw de soporte a Procesos de Negocio que integran e interactúan con objetos del mundo real/físico
  - Usa descripciones de Procesos BPM extendidos con interacciones no obtrusivas y físicas con objetos (AutoID/IoT)
  - Implementa Preso, un fw que usa estas descripciones de procesos y del nivel de obtrusividad, para ofrecer mecanismos de interacción adecuados al contexto de interacción con los objetos físicos

Los Modelos en Tiempo de Ejecución (m@rt)

Ejemplos de Aplicaciones

PROS
Centro de Investigación en Métodos de Producción de Software



Controller Component

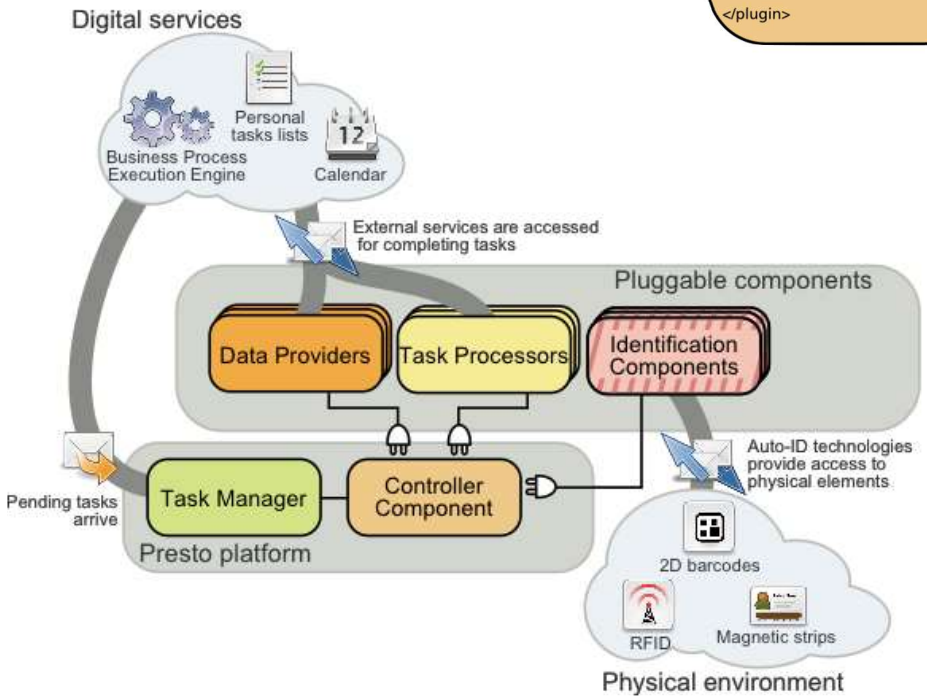
```
<?xml version="1.0" ?>
<!DOCTYPE plugin PUBLIC "-//JPF//Java Plug-in Manifest 1.0"
          "http://jpf.sourceforge.net/plugin_1_0.dtd">

<plugin id="es.upv.pros.presto.controller" version="1.0.0"
          class="es.upv.pros.presto.controller.PluginCore">

 <runtime>
    <library id="core" path="/" type="code">
       <export prefix="*" />
    </library>
 </runtime>

 <extension-point id="TaskProcessor">
    <parameter-def id="class" />
    <parameter-def id="name" />
    <parameter-def id="description" multiplicity="none-or-one" />
    <parameter-def id="initiator" type="boolean"/>
    <parameter-def id="silent" type="boolean"/>
    <parameter-def id="confirmation" type="boolean"/>
</extension-point>

<extension-point id="DataProvider">    ...    </extension-point>

<extension-point id="IDComponent">    ...    </extension-point>

</plugin>
```

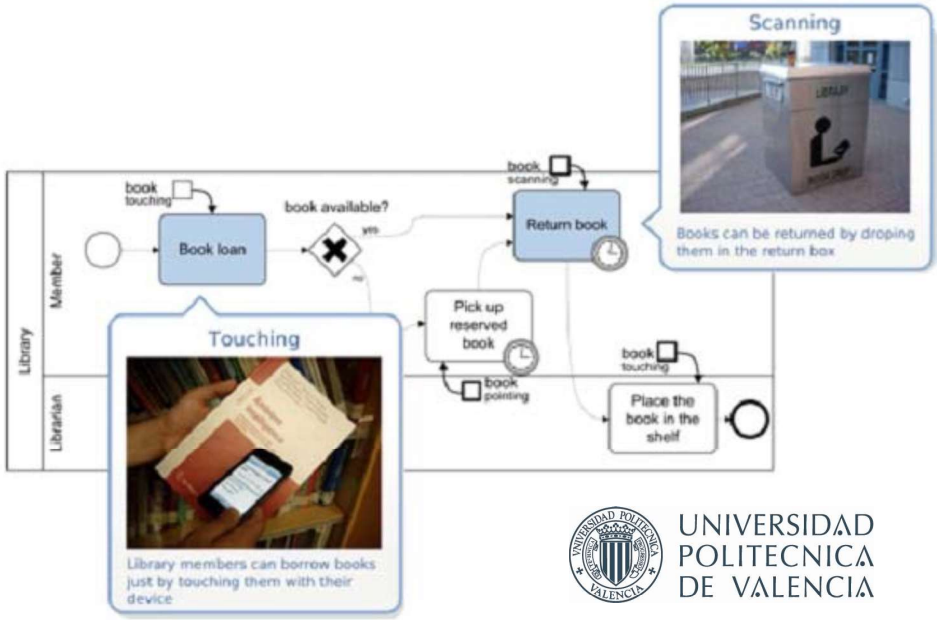
Task Processor

```
<?xml version="1.0" ?>
<!DOCTYPE plugin PUBLIC "-//JPF//Java Plug-in Manifest 1.0"
          "http://jpf.sourceforge.net/plugin_1_0.dtd">

<plugin id="es.upv.dsic.library.return_book" version="1.0.0">

   <requires>
      <import plugin-id="es.upv.pros.presto.controller" reverse-lookup="true"/>
   </requires>

 <runtime>
  <library id="return_resource" path="/" type="code">
   <export prefix="*" />
  </library>
 </runtime>

 <extension plugin-id="es.upv.pros.presto.controller" point-id="TaskProcessor"
    id="return_book">
    <parameter id="class"          value="es.upv.dsic.library.ReturnBook" />
    <parameter id="name"           value="Return Book" />
    <parameter id="description"    value="Supporting the return ..." />
    <parameter id="initiator"      value="false" />
    <parameter id="silent"         value="false" />
    <parameter id="confirmation"   value="true" />
 </extension>
</plugin>
```

Data Provider

Identification Component





UNIVERSIDAD POLITECNICA DE VALENCIA

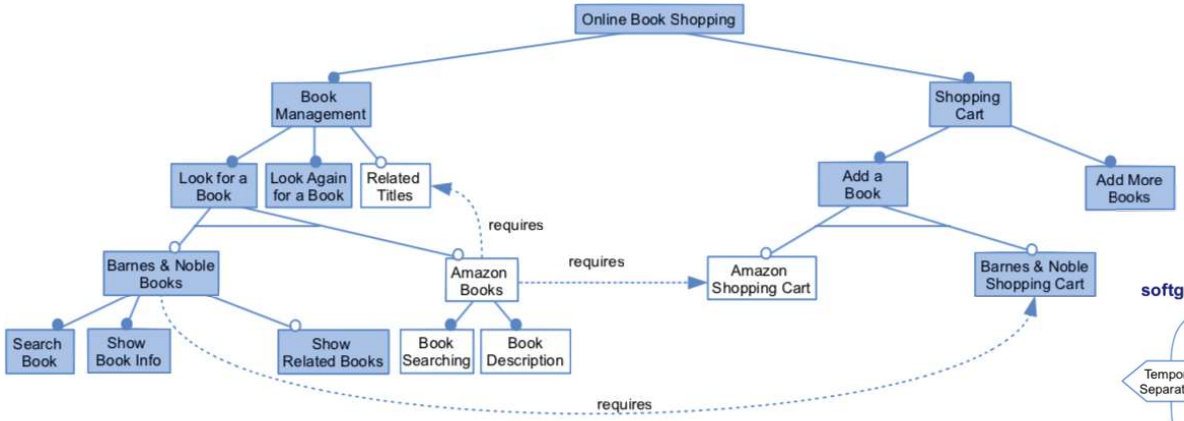
Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones

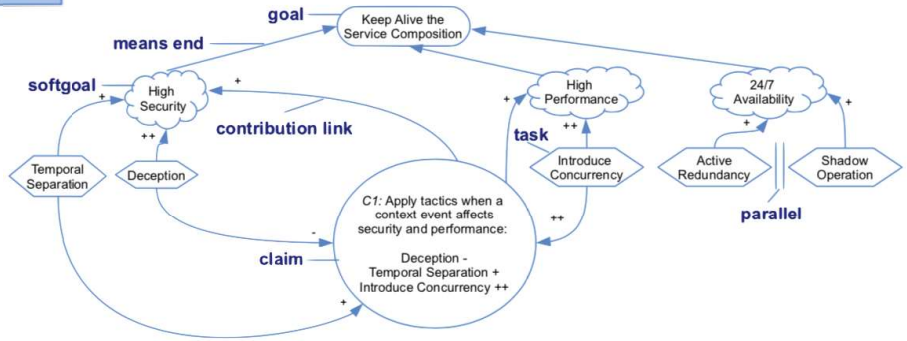
- Composición Autónoma de Servicios Web
  - *PhD 'Achieving Autonomic Web Service Composition with Models at Run-time' (Alférez, 2013)*
  - Diseño de soluciones SOA capaces de cambiar su comportamiento para adaptarse al contexto aplicando estrategias y tácticas para atender a situaciones inesperadas (no previstas)
  - Usa DSPLs para describir la variabilidad de las soluciones y guiar las reconfiguraciones en tiempo de ejecución (m@rt)

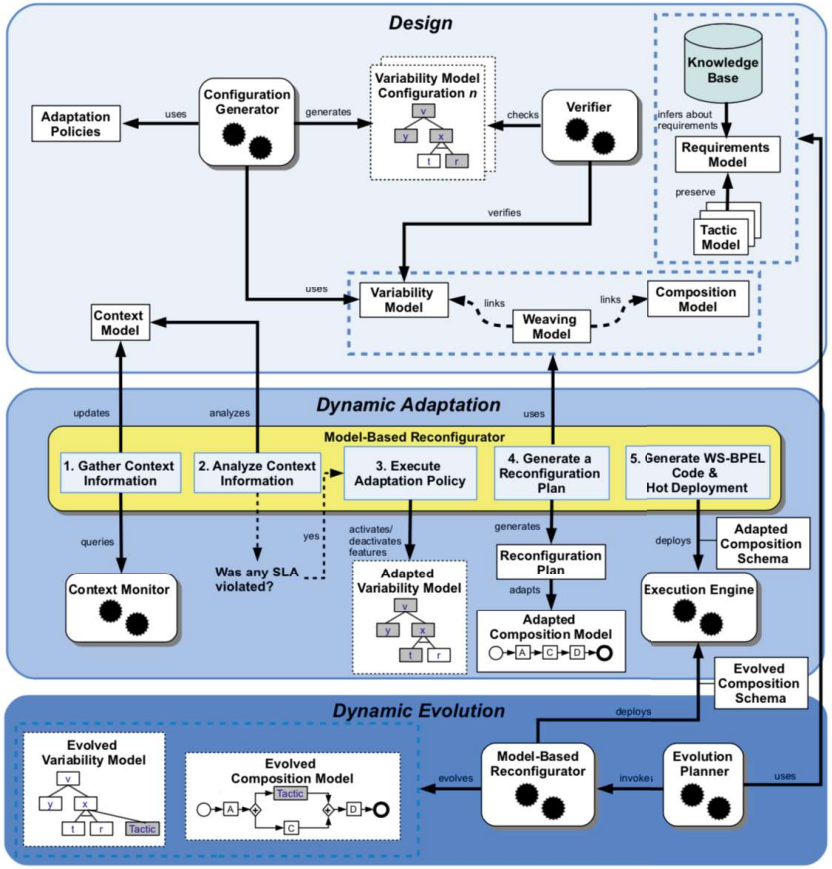
Los Modelos en Tiempo de Ejecución (m@rt)

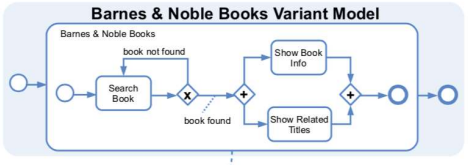
Ejemplos de Aplicaciones

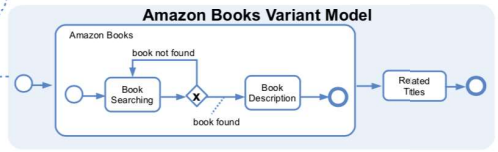
PROS
Centro de Investigación en Métodos de Producción de Software

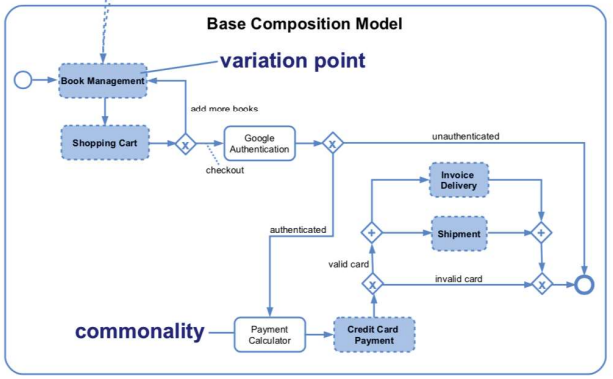










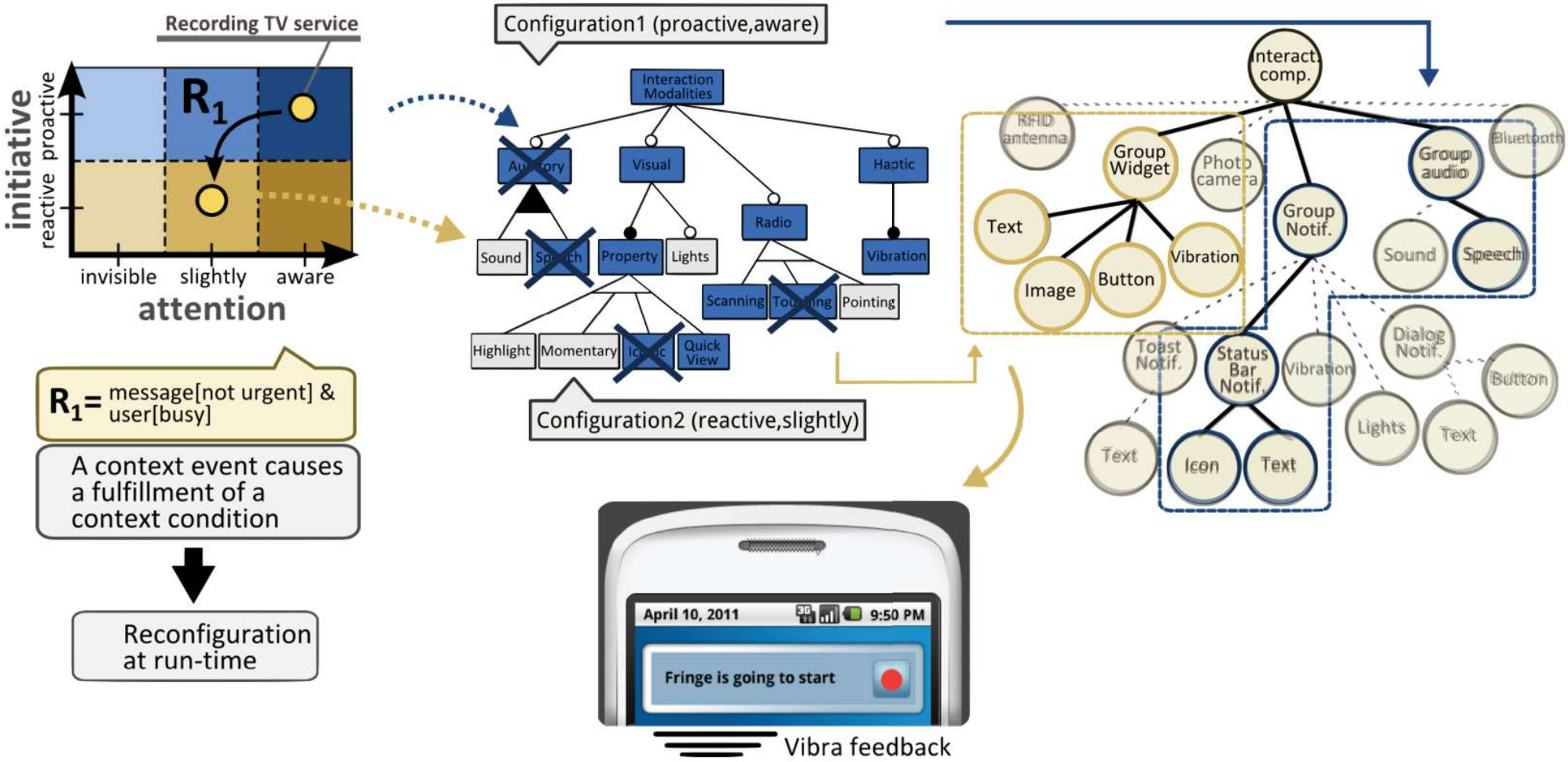


UNIVERSIDAD POLITECNICA DE VALENCIA

Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones

- Adaptación de Interfaces de Usuario Móviles
  - *PhD 'Adapting Interaction Obtrusiveness: Making Ubiquitous Interactions Less Obnoxious. A MDE Approach' (Gil, 2013)*
  - Aproximación que utiliza m@rt para conseguir servicios móviles que son capaces de auto-adaptarse en función del grado de atención que el usuario requiere en cada momento
  - Especifica mediante modelos tanto el nivel de obtrusividad (molestia) asociado a servicios de interacción móvil, como modelos descriptivos de las características de interacción de los dispositivos
  - En tiempo de ejecución, estos modelos se usan para razonar sobre el nivel de molestia del usuario y reconfigurar los mecanismos de interacción concretos para dicho nivel de molestia

Los Modelos en Tiempo de Ejecución (m@rt)
Ejemplos de Aplicaciones



UNIVERSIDAD POLITECNICA DE VALENCIA