



**Obra Abierta**  
CONICET  
UNR  
IMAGE CAMPUS S.A

Directora: Dra. Patricia San Martín

sanmartin@conicet.gov.ar

Becario: Lic. Alejandro Sartorio

sartorio@fceia.unr.edu.ar

# Aspectos Informáticos de un Dispositivo Hipermédial context-aware Dinámico -DHc-aD-

# Contenido

- El proyecto I&D “Obra abierta” (CONICET)
- Concepto de DHc-aD
- Caracterización de una propuesta de **evolución** desde una Aplicación e-learning Web a un DHc-aD.
- Aspectos informáticos de un DHc-aD.
- Modelo compresivo para el diseño de procesos educativos e investigativos en un DHc-aD.



# “Obra abierta” (CONICET-UNR-IC)

## Ejes de estudio:

- ✓ El **taller físico-virtual** como modalidad para educar e investigar.
- ✓ El estudio del **modo interactivo** del dispositivo hipermedial.
- ✓ Los **“contratos”** como **pieza de software** para el modelado de las relaciones entre los participantes del sistema informático. **-Contex Aware dinámico-**

**Bec: A. Sartorio, G. Guarnieri, G. Rodriguez, M.R. de la Riestra.**



## Dispositivo Hipermedial contex-aware Dinámico

Un DHc-aD en un marco organizacional educativo, investigativo y de producción es:

*Una red social mediada por las TIC en un nuevo Contexto presencial físico-virtual.*

*Donde los sujetos:*

*Investigan, enseñan, aprenden, dialogan, confrontan, evalúan, producen y realizan procesos de transformación sobre objetos, regulados según el caso, por una coordinación de contratos integrados a la modalidad participativa del Taller.*

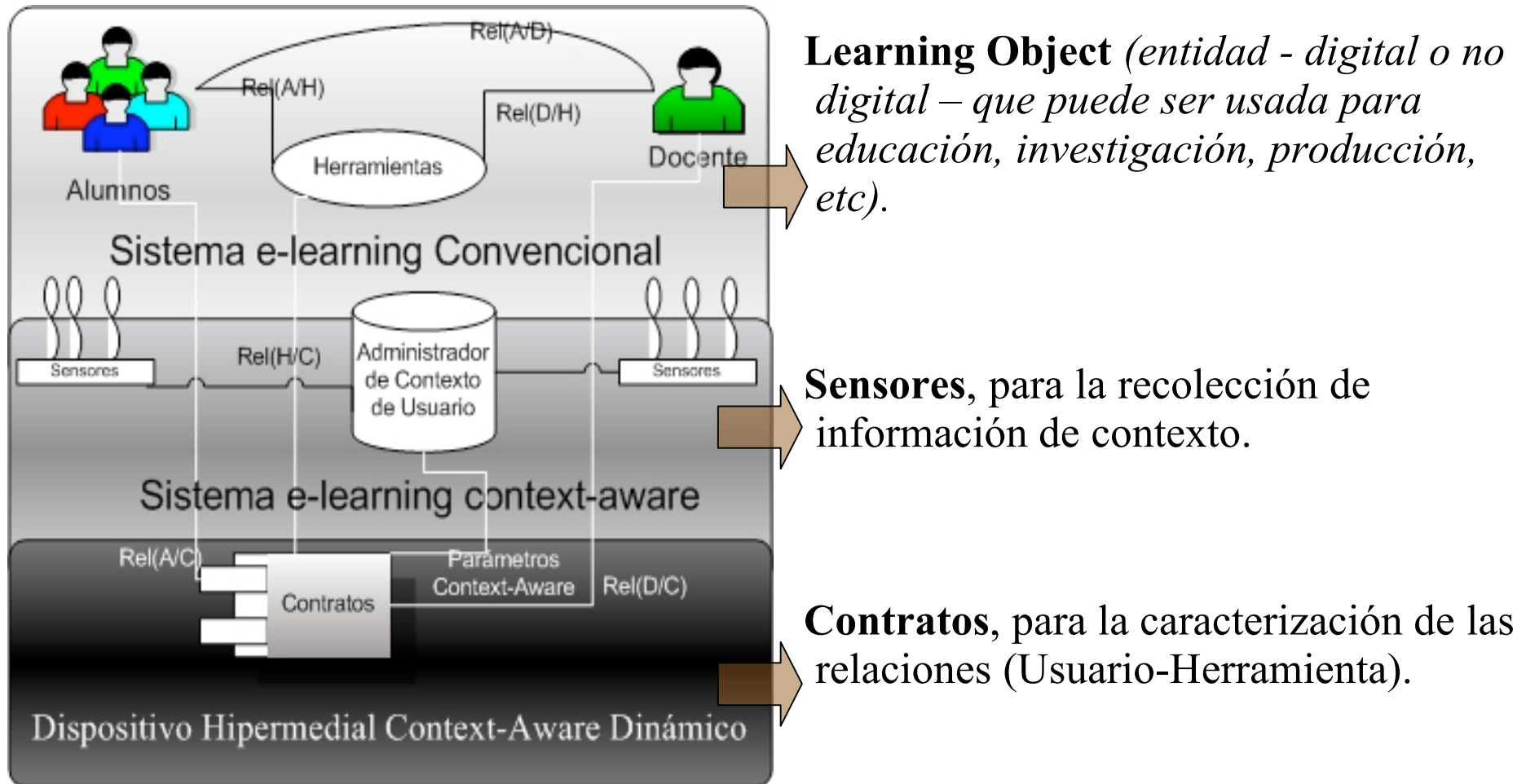




# La evolución por capas

## Componentes conceptuales

## Componentes concretas



# Requerimientos de un DHc-aD

- Reflejar en la arquitectura del sistema los diferentes niveles de **cambios** (run-time) que se producen en las relaciones del dominio del **DH**.
- Soportar **evolución** a través de **reconfiguración dinámica**, sin la interrupción de servicios, minimizando el impacto del sistema global.
- Lograr **adaptación** Hipermedial/**Relacional** con aspectos context awareness.



# Característica del Requerimiento

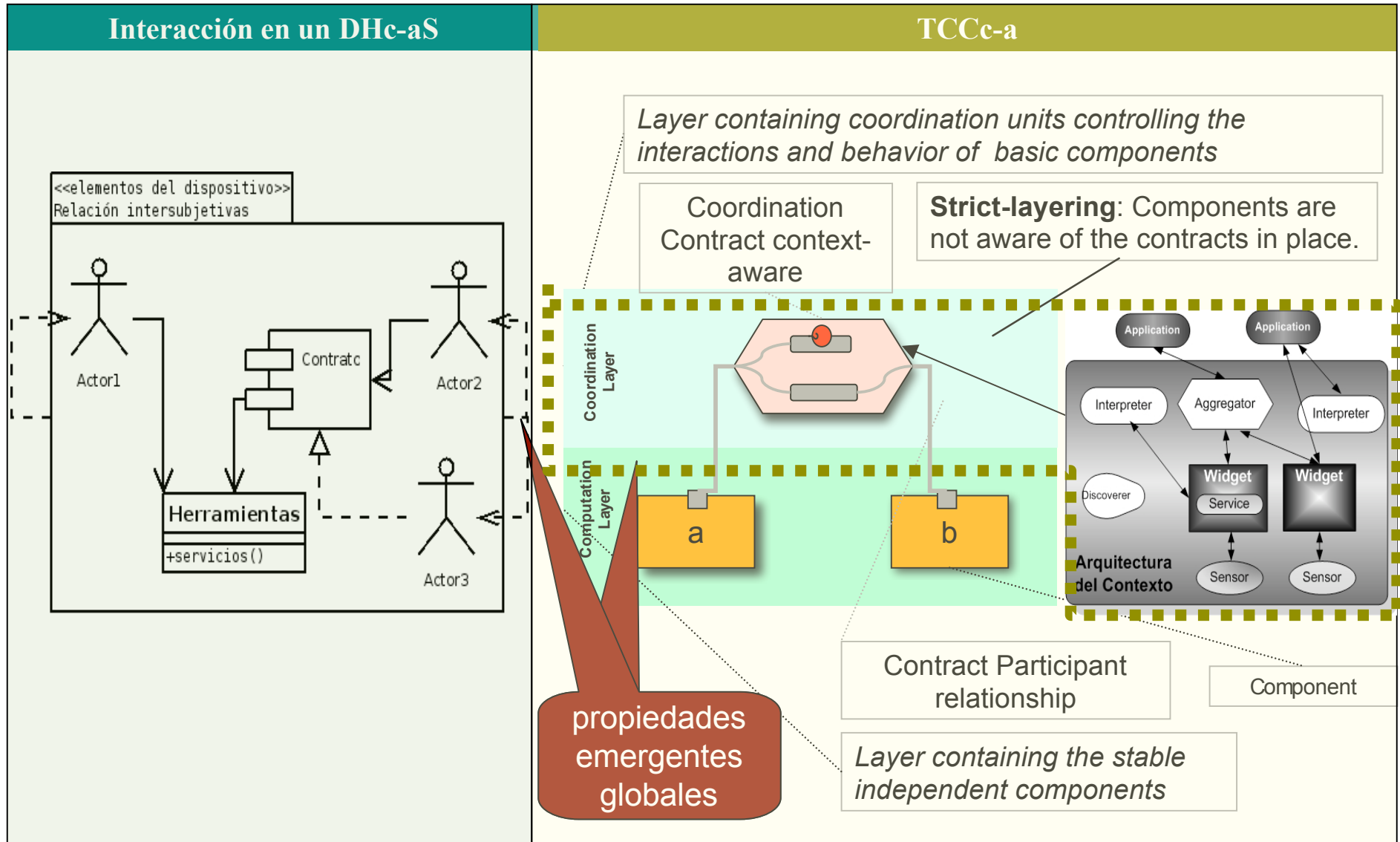
## Cambio vs Creación

"... the ability to **change** is now more important than the ability to **create** [e-commer] systems in the first place. Change becomes a first-class design goal and requires **business and technology architecture** whose components can be added, modified, replaced and reconfigured".

*P. Finger, "Component-Based Frameworks for E-Commerce",  
Communications of the ACM 43(10), 2000, 61-66.*



# Perspectivas de TCCc-a



# Externalización...

- **Superposition** captured through morphisms and universal constructions (colimits)
- **Configurations** modelled as diagrams, and reconfiguration through graph-rewriting
- **Separation** between computation and coordination captured through functors that map systems to coordination interfaces





# TCCc-a: Qué & Por qué

## Qué

- **conexión** establecida entre grupos de **objetos** participantes (con contexto implícito).
- a través de los contratos, **reglas y restricciones** son **superpuestas** al comportamiento de los participantes.
- el contrato define una relación del tipo de **asociación** de clases de UML.
- la **interacción** establecida entre los participantes es más “poderosa” que la que se puede lograr con **UML** y lenguajes **OO**, debido al mecanismo de superposición que elimina los métodos de invocación directa y explícita, reemplazándolo por interacciones del tipo **trigger / reaction**

## Por qué

- Elementos que componen el contrato:
  - colección de clases de roles
  - restricciones
  - atributos y operaciones
  - reglas de coordinación
    - eventos, contexto, condiciones y acciones (pre-condición, post-condición)



# TCCc-a: Cómo

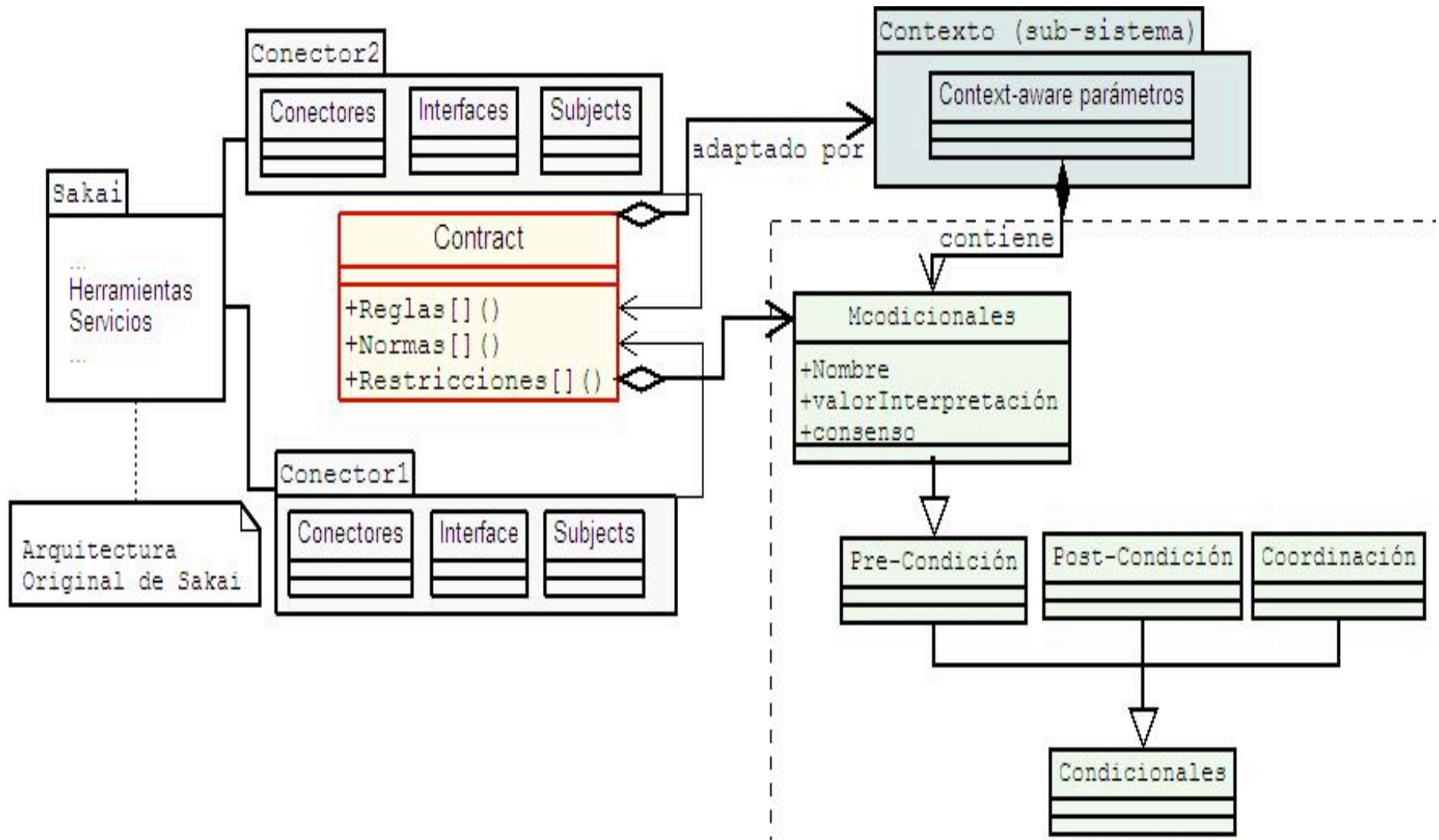
## Cómo

La micro-arquitectura de los contratos context-aware permite su implementación y adaptación (ej. Sakai y JFCA) a través de lenguajes OO, brindando las siguientes ventajas:

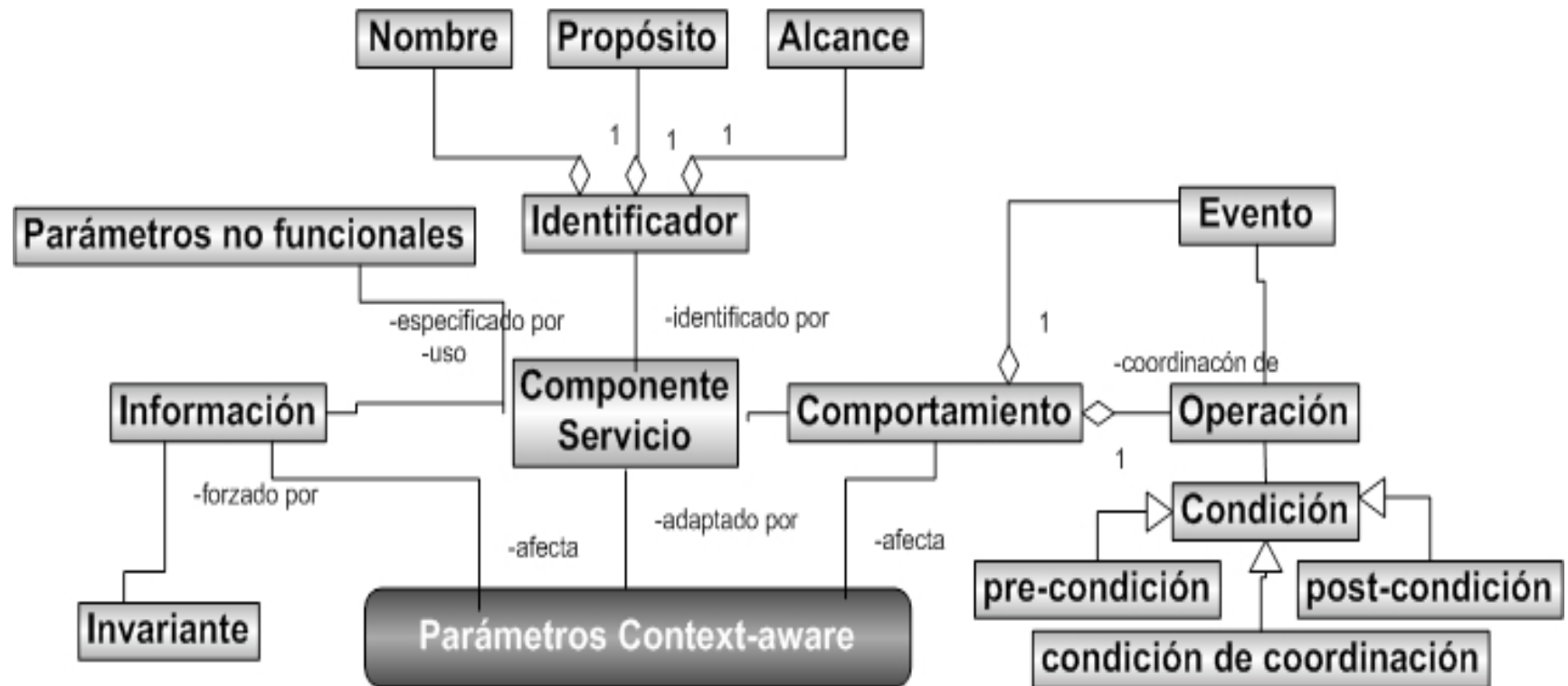
- Las componentes son **independientes** del contratos, no necesitan saber sobre su existencia.
- Los Contratos pueden ser **agregados/borrados** de modo "plug and play", en tiempo de ejecución.
- Las componentes existentes (ej. servicios de las herramientas) pueden ser fácilmente **adaptadas** para la inclusión de los contratos.



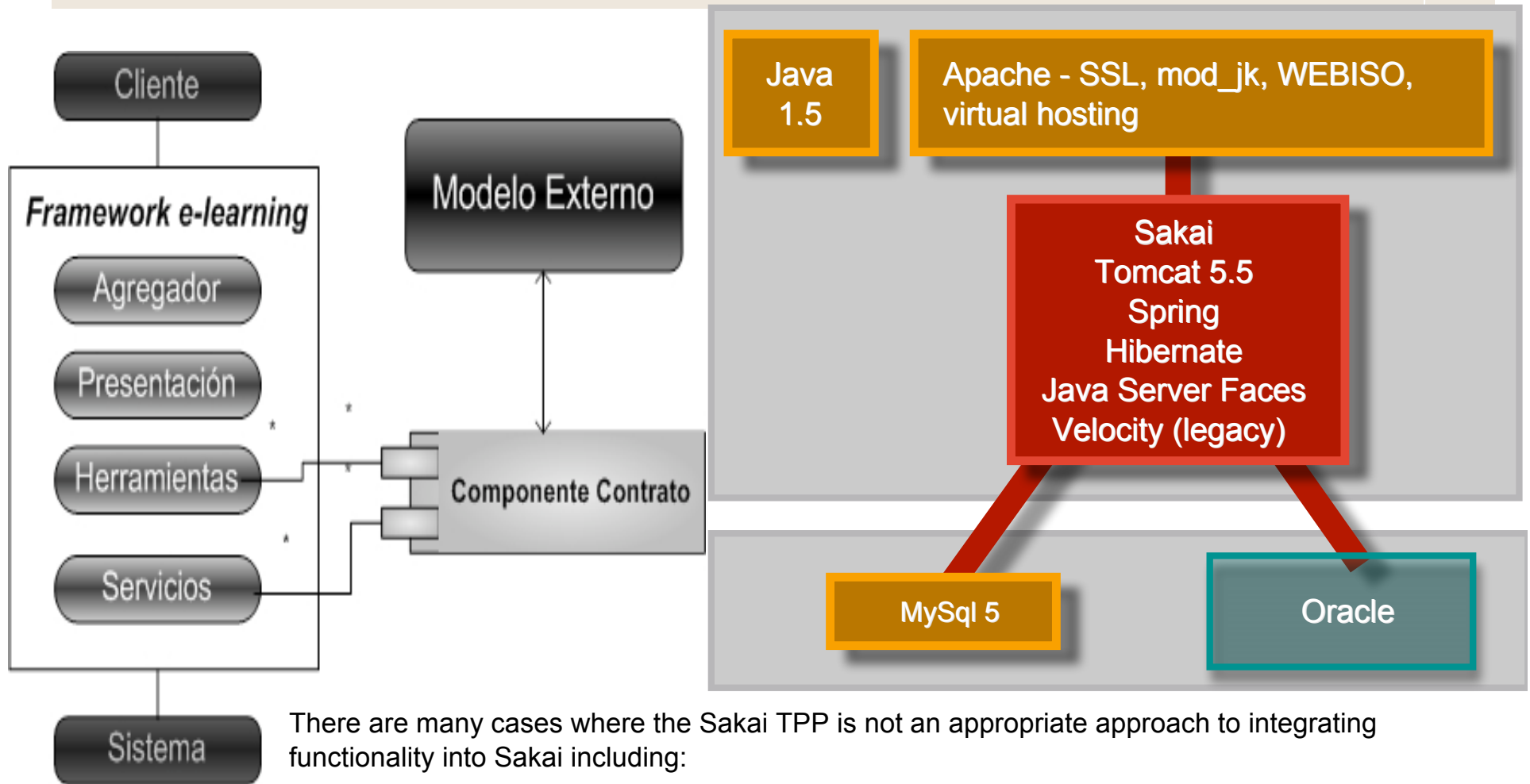
# Macro arquitectura de un DHc-aD



# Arquitectura del contrato c-a



# Arquitectura Sakai



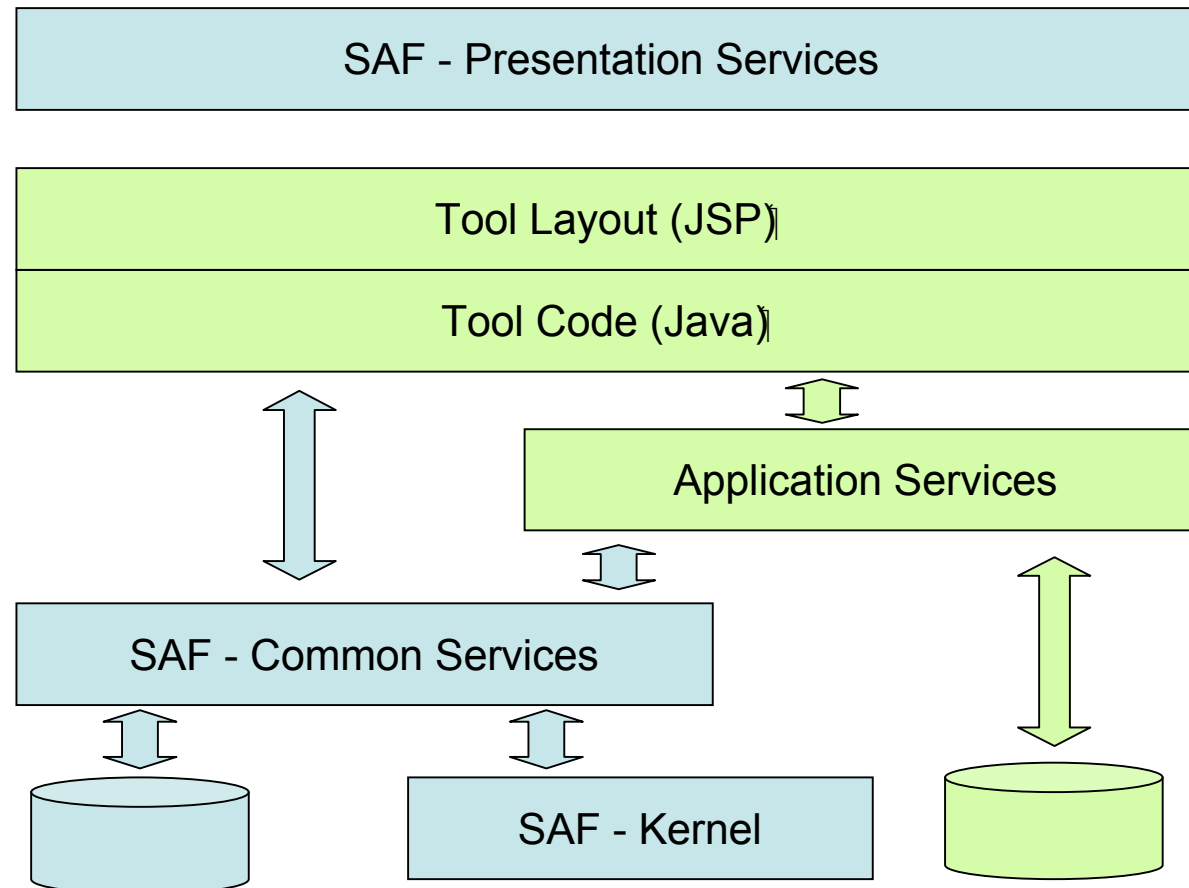
There are many cases where the Sakai TPP is not an appropriate approach to integrating functionality into Sakai including:

- An application that needs to operate both within Sakai and independent of Sakai
- A large application using presentation technology other than JSF, or is using JSF in ways which are not compatible.





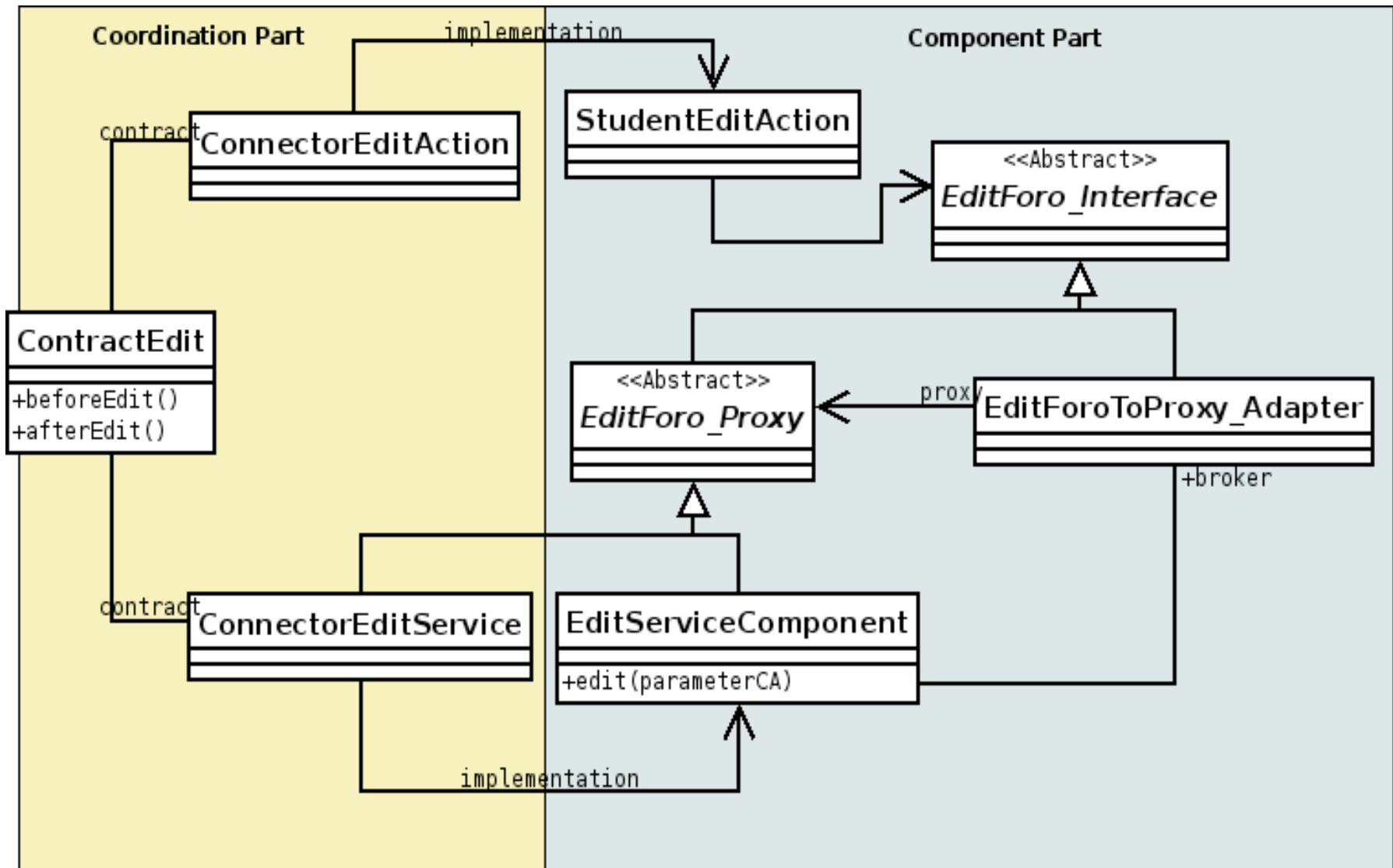
# Sakai Application Framework (SAF)



JCC 2007



# Micro arquitectura de TCCc-a



# Componente Servicio

```
package org.sakaiproject.discussion.impl;
..
import org.sakaiproject.discussion.api.DiscussionChannel;
import org.sakaiproject.discussion.api.DiscussionChannelEdit;
import org.sakaiproject.discussion.api.DiscussionMessage;
import org.sakaiproject.discussion.api.DiscussionMessageEdit;
import org.sakaiproject.discussion.api.DiscussionMessageHeader;
import org.sakaiproject.discussion.api.DiscussionMessageHeaderEdit;
import org.sakaiproject.discussion.api.DiscussionService;
...
public class DbDiscussionService extends BaseDiscussionService
{
...
public Edit newContainerEdit(Element element)
{
    BaseDiscussionChannelEdit rv = new BaseDiscussionChannelEdit(element);
    rv.activate();
    return rv;
}
public Edit newResourceEdit(Entity container, String id, Object[] others)
{
    BaseDiscussionMessageEdit rv = new BaseDiscussionMessageEdit((MessageChannel)
container, id);
    rv.activate();
    return rv;
}
public MessageEdit editMessage(MessageChannel channel, String id)
{
    return (MessageEdit) super.editResource(channel, id);
}
```



# Componente Servicio cc-a

```
package org.sakaiproject.discussion.impl; import java.util.*; import cde.runtime.*; import obab.ca.*;
public abstract class BaseDiscussionService extends BaseMessageService implements DiscussionService,
ContextObserver,
    EntityTransferrer, ForoInterface
{
...
public messageEdit editMessage(MessageChannel channel, String id)
    {    new ComponentOperationEvent(this , "Edit").fireEvent();
      return (MessageEdit) super.editResource(channel, id);
    }
// Code added by the contract generator
protected CrdIProxy _proxy;
// Class Ids definiton
private static Class _classId= Sakai.Discussion.class;
public static Class GetClassId() {return _classId;}

public CrdIProxy GetProxy() { return _proxy; }
public void SetProxy( Object p ) { if ( p instanceof CrdIProxy && p instanceof DiscussionInterface) _proxy =
(CrdIProxy)p; } //else exception!!
public void SetProxy(CrdIProxy p) { _proxy = p; }

AccountInterface GetProxy_Account() {if ( _proxy == null ) return null; return (DiscussionInterface)
_proxy.GetProxy(_classId);}

// These methods implement the Proxy Client part of the Subject

public long getMessage ()

{    new ComponentRequestEvent(this , "getMenssage").fireEvent();
  AccountInterface proxy = GetProxy_Discussion();
  if ( proxy==null || proxy==this ) return _getMessage(); else {long value = proxy.getNumber();return value; } }
```

JCC 2007



# Conector c-a

```
package org.sakaiproject; import java.util.*; import cde.runtime.*; import obab.ca.*;
public abstract class IDiscussionPartner extends CrdContractPartner implements CrdIProxy,
DiscussionInterface {
    /** Attributes ****/
    protected Discussion subject;
    /** Constructor ****/
    public IDiscussionPartner(CrdContract ct, Class c)
    { super(ct, c); }
    /** Definition of abstract methods in CrdContractPartner ****/
    public void SetProxy(Object p) {subject.SetProxy(p);}
    protected Object GetSubject_Object() { return subject; }
    public void ResetProxy() { subject.SetProxy(null); }
    /** Typed Access Methods ****/
    protected Discussion GetSubjectDiscussion(){return (Discussion) subject;}
    protected IDiscussionPartner GetNextPartner_Discussion() {return
(IDiscussionPartner)GetNextPartner(Discussion.GetClassId());}
    protected IDiscussionPartner GetNextPartner_Discussion( CrdPartnerRulesListIterator _it) {return
_it.hasNext() ? (IDiscussionPartner)((CrdPartnerRules) _it.next()).partner :null;}
    /** Default Implementation of Discussion Interface **/
    public void messageEdit (double amount, Customer c) throws DiscussionException
    {
        IDiscussionPartner next = GetNextPartner_Discussion();
        if (next != null) next.editMessage(amount,c); else GetSubjectDiscussion()._editMessage(amount,c);
    }
    /** Default Implementation for condition checkings **/
    ...
    public CrdPartnerRules messageEdit_rules(string texto, Student c) throws DiscussionException,
CrdExFailure
    { return new CrdPartnerRules (this);
    }
    ...
}
```

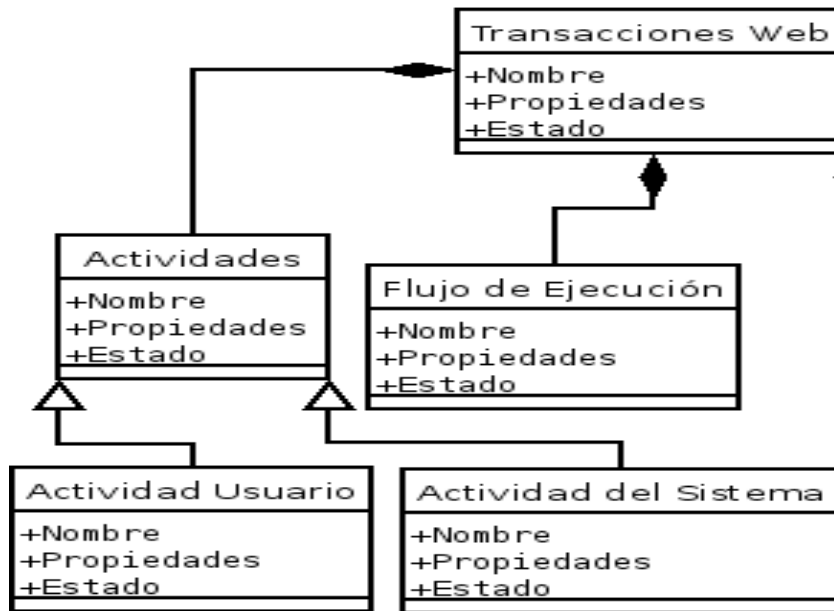
JCC 2007



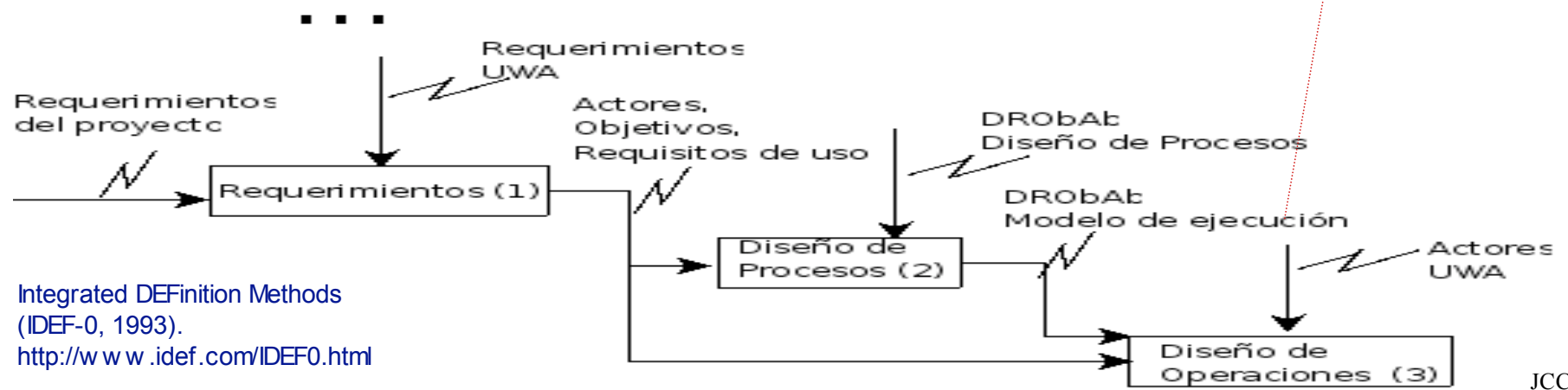
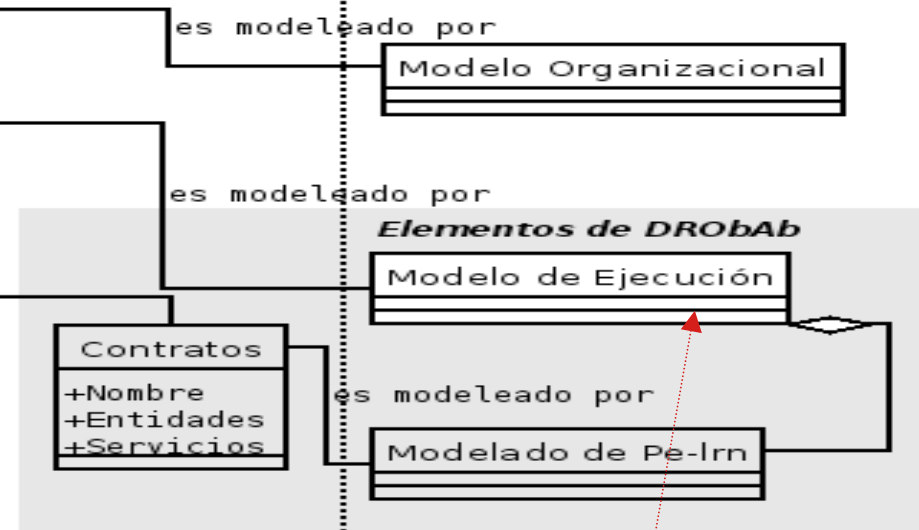


# UWATc+: Modelo de diseño de transacciones e-learning c-a

UWATc+ conceptos y relaciones entre conceptos



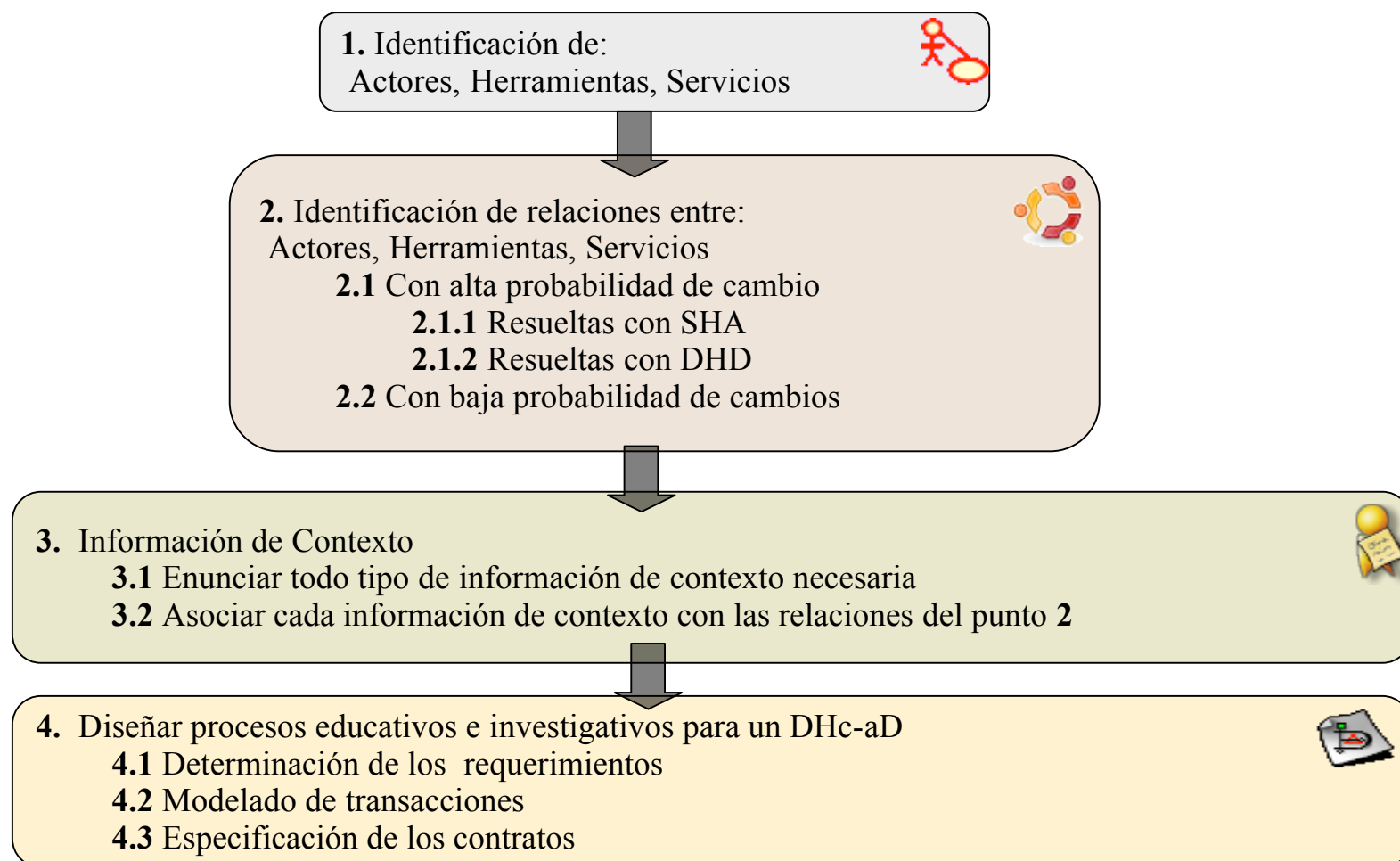
UWATc+ Modelos de Diseños



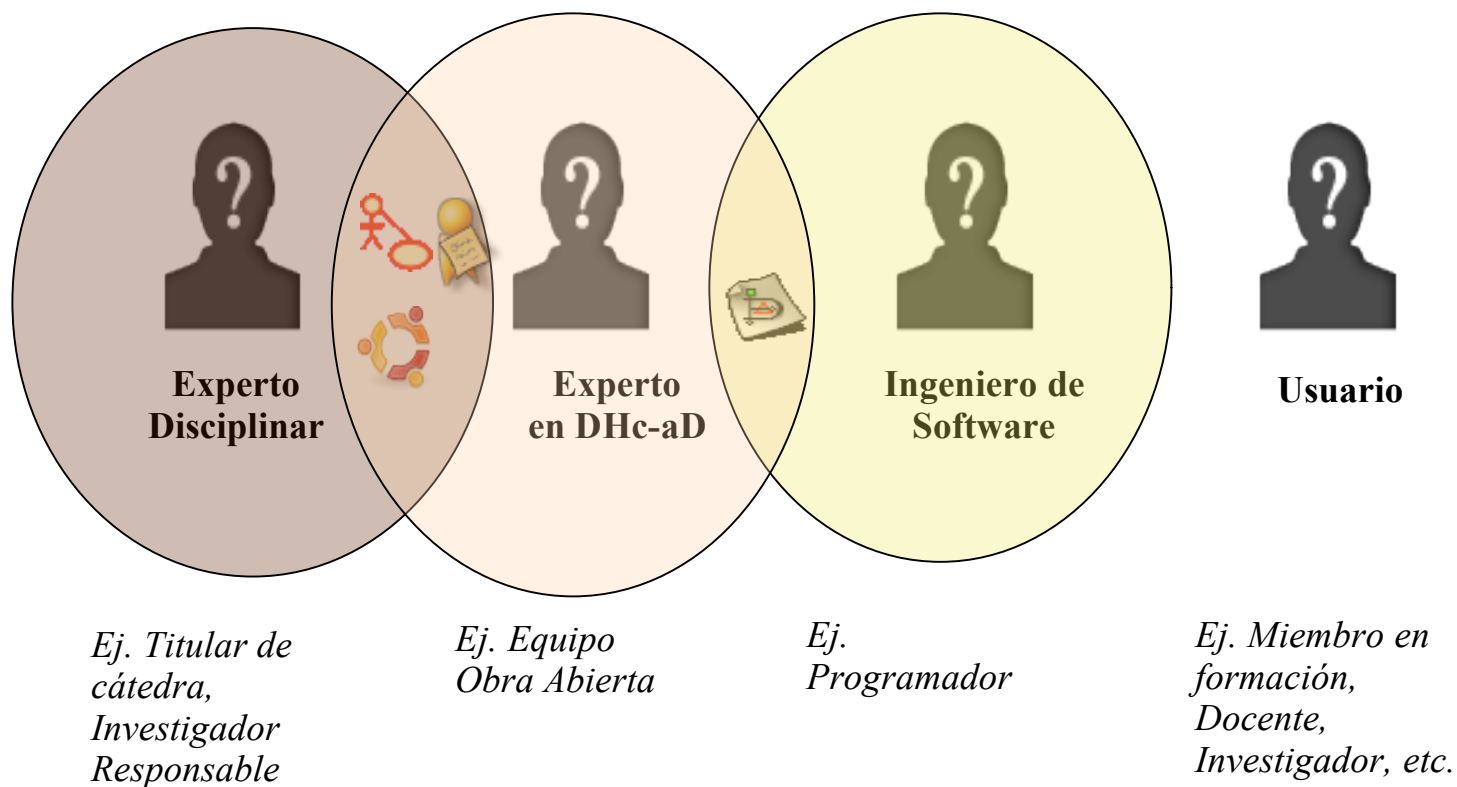
Integrated DEFinition Methods (IDEF-0, 1993).  
<http://www.ideal.com/IDEF0.html>

JCC 2007

# Metodología para la inclusión de los contratos e-a

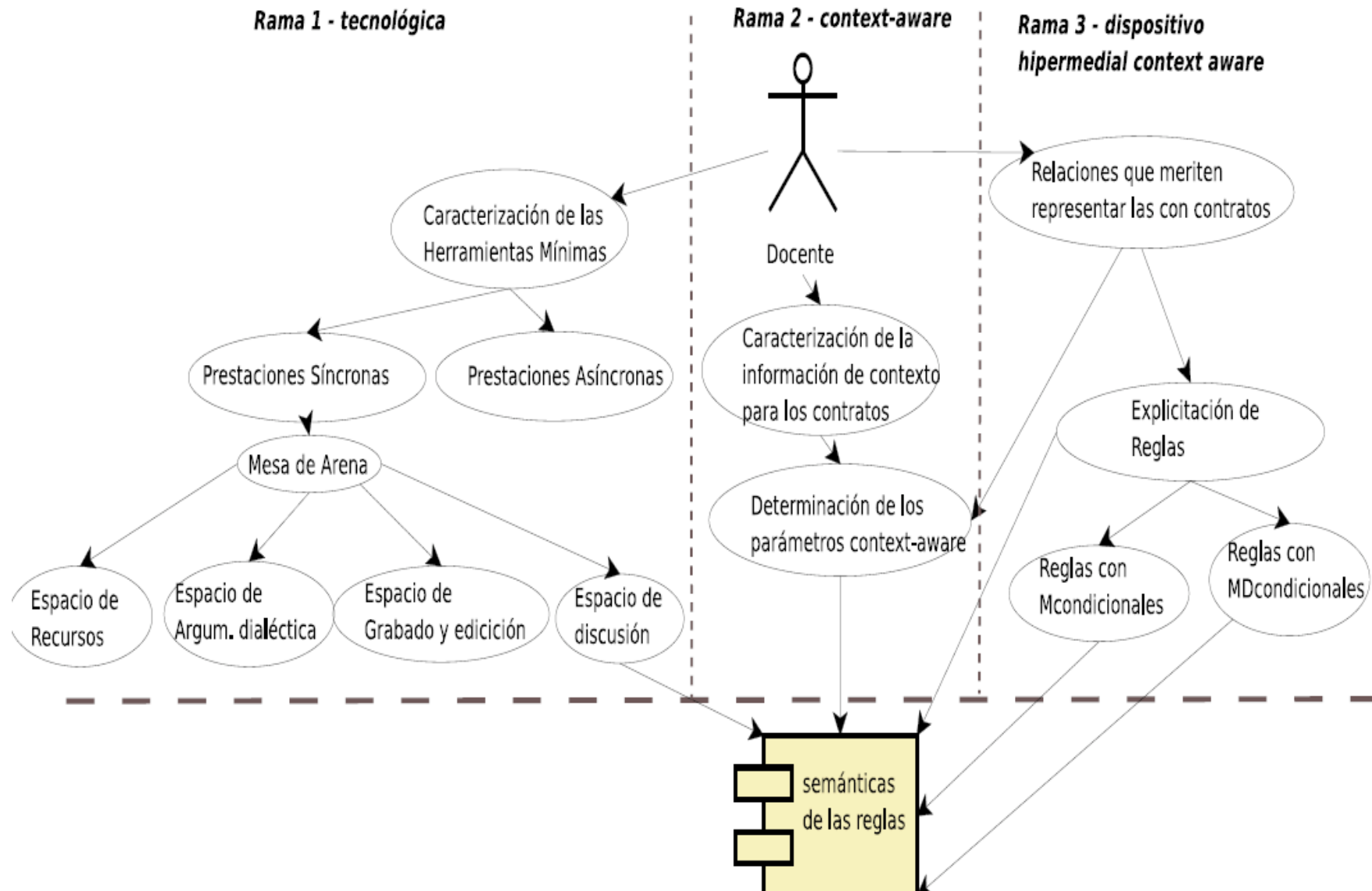


# Actores del ciclo de vida del desarrollo de un DHC-aD



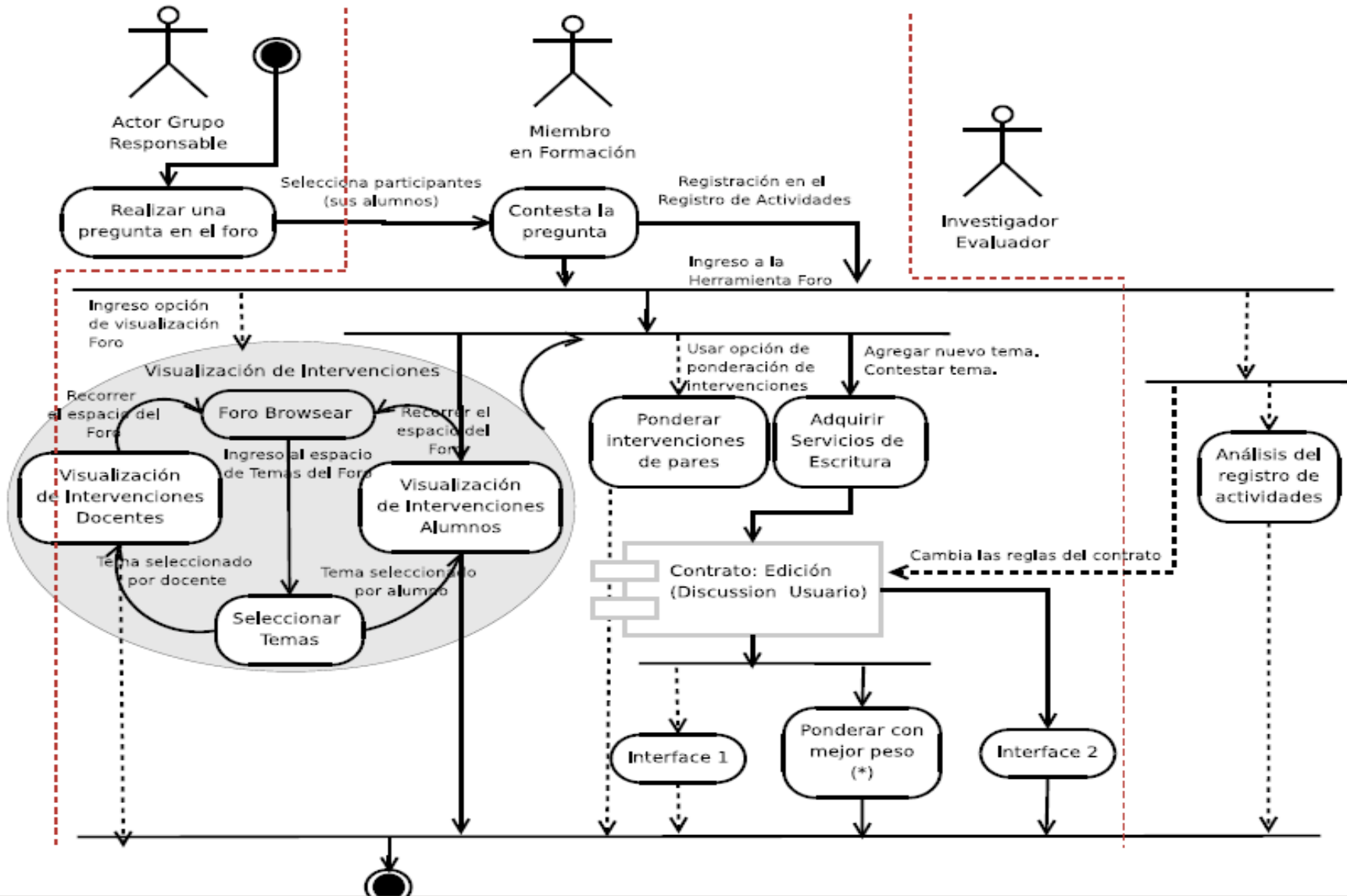


# Determinación de Requerimientos





# Diseño de transacciones e-learning c-a







## Diagrama de un contrato

Contrato: Edición		
1. <i>Participantes:</i>	<b>d:</b> DiscussionAction	<b>u:</b> UserAction
2. <i>Param. c-a:</i>	state, portlet, rundata, context	contextidentifier, identifier
3. <i>Servicios:</i>	initState()	getIdentifier()
4. <i>Pre-Cond:</i>	existe < contexto >	existe < contexto >
5. <i>Pos-Cond:</i>	modifica < contexto >	
6. <i>Reglas de Coordinación:</i>	<b>Si</b> u.contexto='p1;d;r1;c1;' <b>entonces</b> d.showMessage(data,string)	

**Comentario**

1. DiscussionAction y UserAction pertenecen a clases implementadas en JAVA del proyecto Sakai.  
4 y 5. < contexto > refiere a un objeto donde se oculta toda la información de contexto que caracteriza a los usuarios de la plataforma

▼ [e] project		
@ xmlns	http://maven.apache.or	
[e] modelVersion	4.0.0	
▼ [e] parent		
[e] artifactId	jsf-base	
[e] groupId	org.sakaiproject	
[e] version	M2	
[e] relativePath	../pom.xml	

```
<?xml version="1.0"?>
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <artifactId>jsf-base</artifactId>
    <groupId>org.sakaiproject</groupId>
    <version>M2</version>
    <relativePath>../pom.xml</relativePath>
  </parent>
```



# Nuestro espacio Obra Abierta

Los invitamos a nuestro espacio de investigación sobre **D**ispositivos  
**H**ipermediales **c**ontext-**a**ware **D**inámico:

<http://200.80.157.171:8080/portal>



JCC 2007

