

## Linking Norms to Service-Oriented implementations: *From CARREL to OTMA*

Javier Vázquez-Salceda.



Knowledge Engineering and Machine Learning Group  
UNIVERSITAT POLITÈCNICA DE CATALUNYA

## Contents

- Introduction
  - Background research
  - CARREL
  - From Agent-based CARREL to SOA-based OTM
- Provenance handling in the OTM Application
  - Provenance representation
  - Provenance questions
  - Issues to solve
  - Norm enforcement through provenance
- Conclusions



## Introduction



## Background (I)

- Now a days, computing trends move toward **distributed solutions**
  - computer systems are networked into **large distributed systems**;
  - processing power can be introduced in almost any place and device → processing becomes ubiquitous
- The **agent paradigm** is one way to conceptualize and implement distributed (intelligent) systems
  - Agents are *human-oriented* abstractions
  - Each agent can specialize in some (sub)problems and take decisions *locally*
  - Solutions to coordinate the agent society can be borrowed from **human organizations** and **human societies**

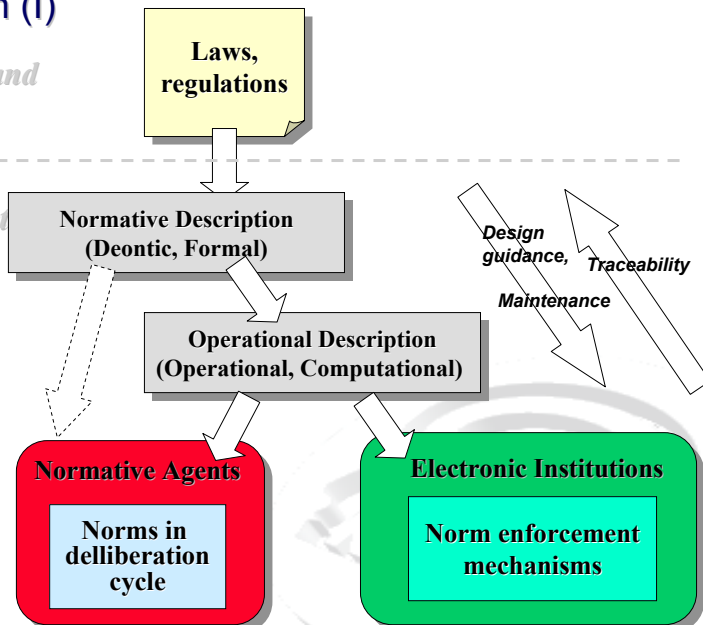
## Background (III)

- **Norms** are a flexible way to specify the boundaries of acceptable behaviour
  - They specify **WHAT** is acceptable and **WHAT** is not, but not **HOW**
  - Agents have autonomy to reach their goals as far as they “move” within the acceptable boundaries.
- Norms **ease agent interaction**:
  - reduce **uncertainty** of other agents’ behaviour
  - reduce **misunderstanding** in interaction
  - allows agents to **foresee the outcome** of an interaction
  - simplify the **decision-making** (reduce the possible actions)
- To ensure acceptable behaviour, a safe environment is needed: **Electronic Institutions**
  - Safe agent interaction environments
  - They include definition of norms and enforcement mechanisms

## Approach (I)

*too abstract and vague*

*more concrete*



## Approach (II)

- Implementation of norms from institutional perspective  $\neq$  Implementing a theorem prover to check protocol compliance
- Implementation of a safe environment (**norm enforcement**)
- 2 options depending on control over agents
  - Defining constraints on unwanted behaviour
  - Defining violations and reacting to these violations
- our assumptions:
  - Norms can be sometimes violated by agents
  - The internal state of agents is neither observable nor controllable
    - actions cannot be imposed on an agent's intentions
    - agents as black boxes
    - only their observable behaviour and actions

7

## Application area: *CARREL 2.0*

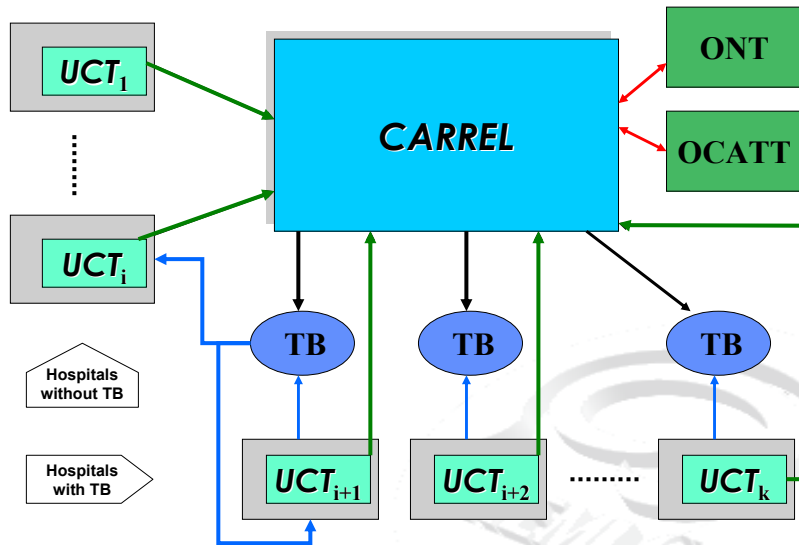
- Aim 1: Decrease the organ and tissue **seek time**.
  - Transplantation success ↑
  - Avoid tissue loss in Tissue Banks
  - Avoid loss of organs suitable for transplant
- Aim 2: Maximize **organ and tissue compatibility**
- **Fair** distribution of organs and tissues
- Follow the actual **norms and laws**

### • **Proposal:**

- To create an **Agent Mediated Institution**

8

## CARREL: Actors



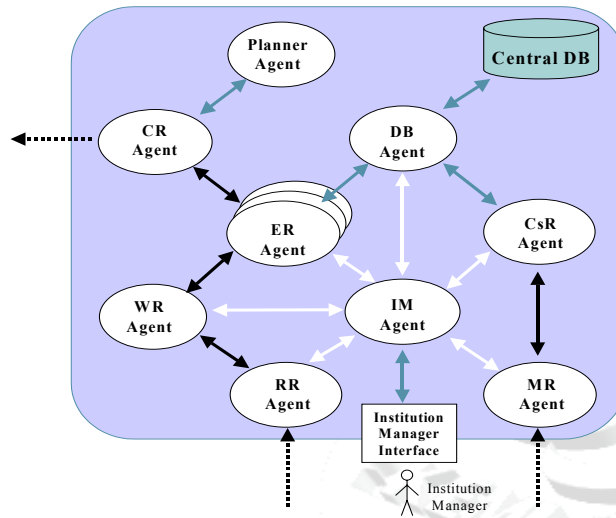
9

## CARREL: tasks

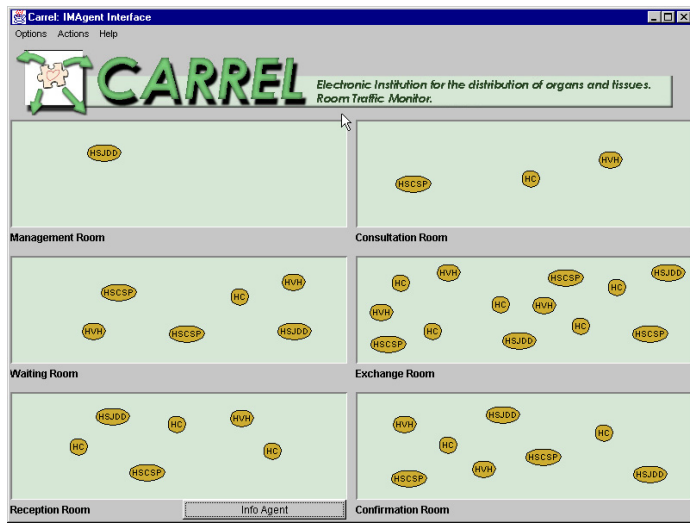
- Ensure the **acceptable behavior** of the agents and the fulfillment of the commitments
- Management of **requests** and assignment
  - Searching for the **best** tissue for a recipient.
  - Searching for the **best** recipient for an organ.
- **Database management**
  - Trace the pieces
  - Security issues
- Planning the **piece delivery** to hospitals

10

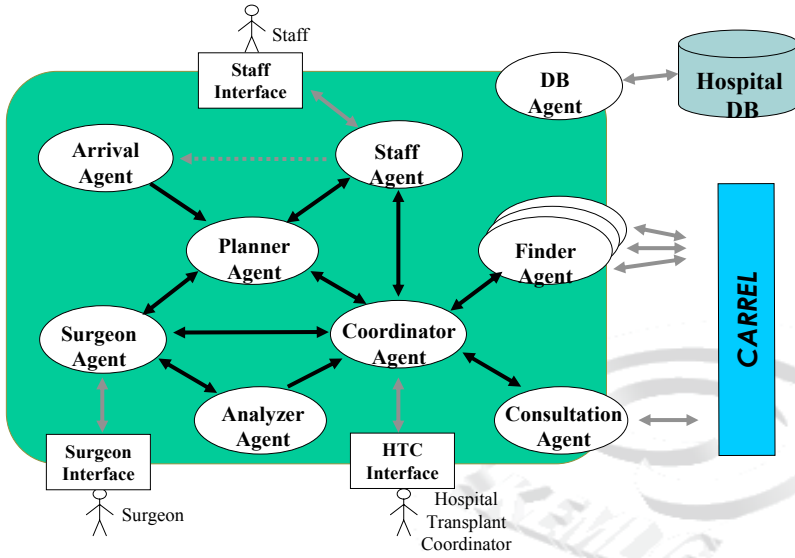
## CARREL: Agent architecture



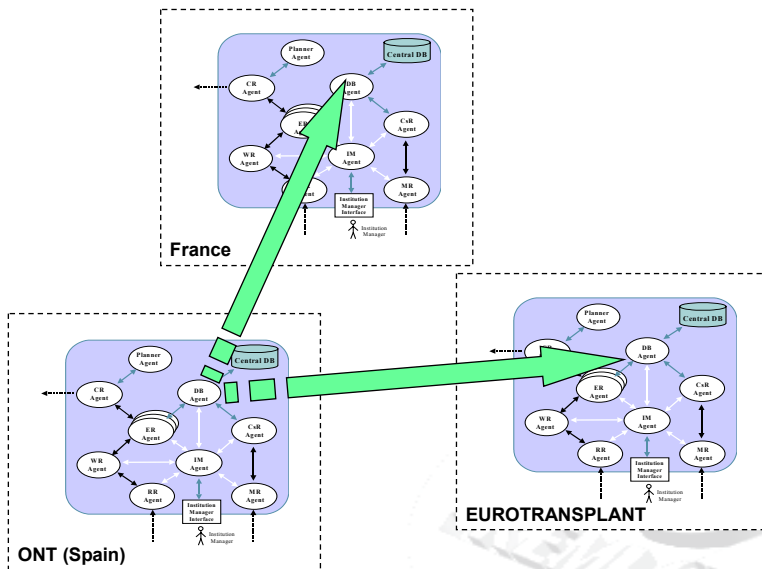
## CARREL: the institution's interface



# CARREL: the UCTx system.



# Distributed search



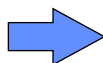
## Problem: agent platforms in health care institutions

- CARREL 1.0 and CARREL 2.0 created interest in the users
  - Even though quite some **medical staff** tends not to like computers, they saw potential in the support the system would offer
- Problem: trying to deploy the prototype into their computer network...
  - System administrators are responsible for the proper functioning of all computers and programs
  - They are reluctant to install anything they do not understand
  - They do not understand agent technology
- **What do we do now?**

15

## Target: Service Oriented Architectures

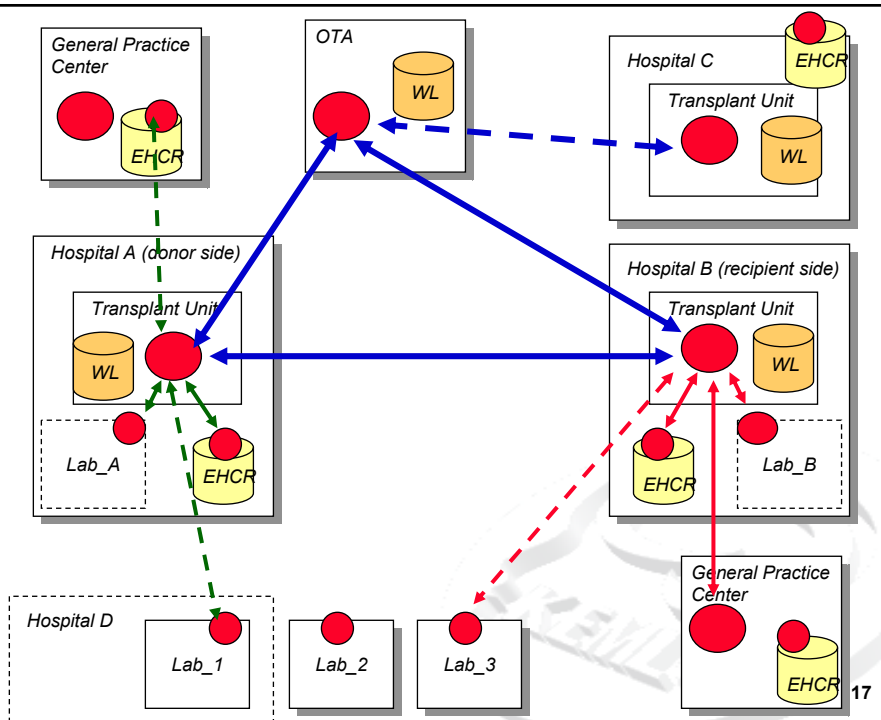
- Current trend in distributed computation: **Webservices, GRID computing**
- Service Oriented Architectures framework
  - Broad definition of **service** as component that takes some inputs and produces some outputs.
  - Services are brought together to solve a given problem typically via a **workflow** definition that specifies their composition.
- Every application is made up of **actors**
- Every change that happens is an action by an actor
- Actors communicate by sending **messages**
- Every action is triggered by a message
- The outputs of (messages sent by) an actor are **caused** by the inputs to (messages received by) the actor



*Direct mapping to multiagent systems*

16





## How can norm compliance be introduced in SOA?

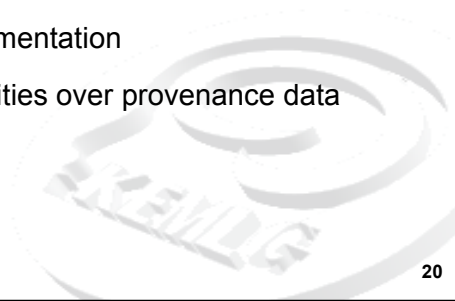
- **SOA governance**
  - refers to policies and software tools that aim to manage service-oriented architecture
  - involves both design-time and run-time aspects
    - Design-time: enterprise architects create a set of rules that define
      - how services should be constructed
      - how services may be deployed (including access rights)
    - Run-time: Governance software
      - helps put the SOA guidelines into action
      - monitors the performance of services
- **SOA provenance**
  - Refers to desired process definition (workflows) and software tools to trace process execution
  - Includes tools to register meaningful events and interactions and to re-create

## Provenance Handling in the OTM Application



### Aims of EU PROVENANCE project

- Define core concepts pertaining to provenance
- Specify functionality required to become “provenance-aware”
- Define open data models and protocols that allow distributed systems to inter-operate
- Standardise data models and protocols
- Provide a reference implementation
- Provide reasoning capabilities over provenance data (higher-level queries)



## Use cases

**Aerospace engineering:** maintain a historical record of design processes, up to 99 years.



**Organ transplant management:** tracking of previous decisions, crucial to maximise the efficiency in matching and recovery rate of patients

21

## Provenance representation

- Provenance architecture from EU PROVENANCE
  - In OTM, each organisational unit is represented by a *service*.
  - Staff connect to the unit's service through a *GUI interface*
  - Services are seen as *actors* exchanging *messages*
  - Our def: "*Provenance of a piece of data is the process that lead to the data*"
  - Provenance of data represented by *p-assertions*:
    - *Interaction p-assertions* (about contents of messages)
    - *Relationship p-assertions* (about relation between an actor's input and output messages)
    - *Actor-state p-assertions* (about actor's internal state in the context of an interaction)
  - P-assertions are stored and organised inside *provenance stores*.

22

## Provenance questions

- Examples of types of queries
  - *Where* did the medical information used in step X came from?
  - *Which* medical actor was the source of information A?
  - *When* a given medical process was carried out?
  - *Who* was responsible for a given medical process?
  - *When* decision Y was taken?
  - *What* was the bases for decision Y?
  - *Which* medical actors were involved in the decision?
  - *Which* medical actor refused to provide medical data for a decision?

23

## Issues to solve

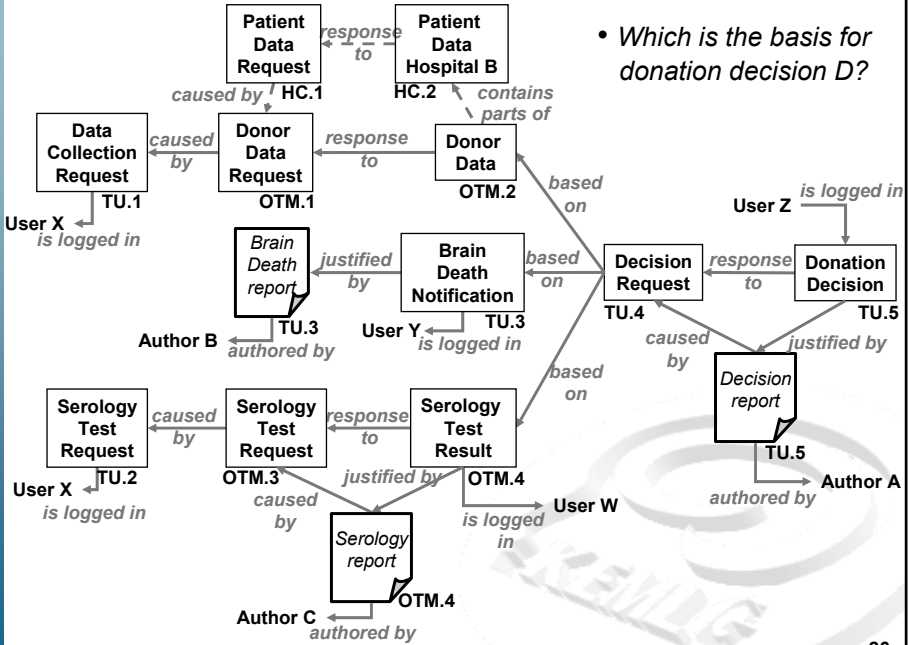
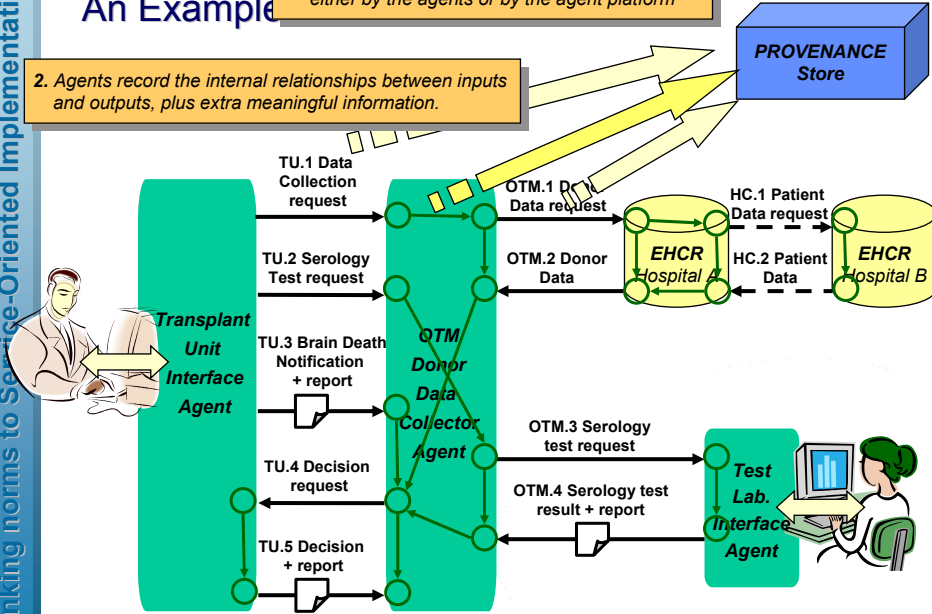
- The provenance of most of data is *not a computational service*, but decisions and actions carried out by *real* people in the *real* world.
  - Make sure the *electronic* system gets *enough information* from the *real* processes but trying to *minimise interference* to medical actors.
- **Past treatments** of a given patient in other institutions **may be relevant** to, e.g., current decision in current institution
  - p-assertions about the processes underwent in previous treatments should be **connected somehow** to **current p-assertions**
- **Security and privacy** issues

24

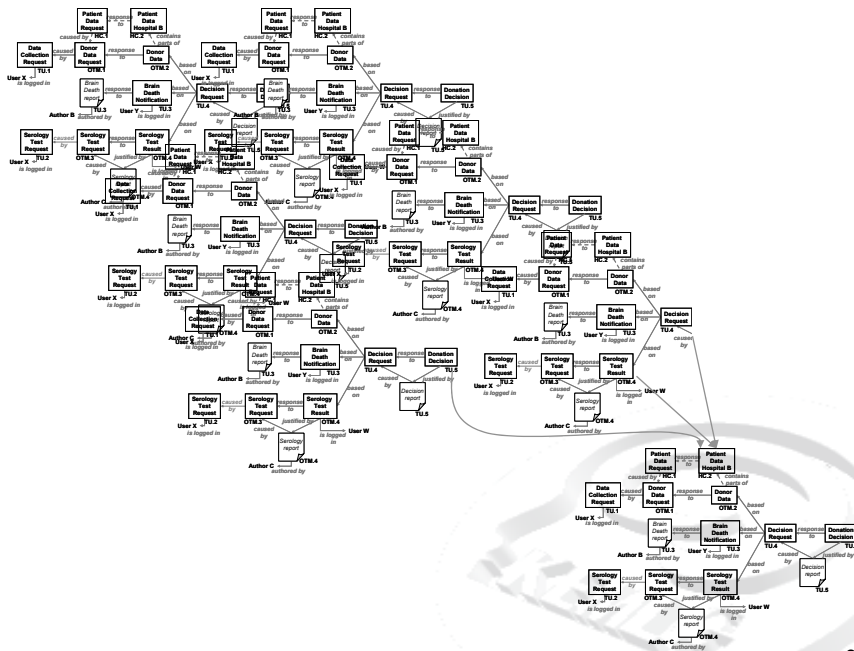
## An Example

1. Agent messages are recorded as interactions, either by the agents or by the agent platform

2. Agents record the internal relationships between inputs and outputs, plus extra meaningful information.



• Which is the basis for donation decision D?



## Norm enforcement through provenance (I)

- Idea: use the p-assertions in the provenance store to track distributed execution
  - Interaction p-assertions to trace actions
    - utterances between agents

*Asserted(ddca2, ddca2 uttered(serology\_test\_request(ddca2,t11,Donor\_data))*

*Asserted(t11, ddca2 uttered(serology\_test\_request(ddca2,t11,Donor\_data))*

- Calls to external resources

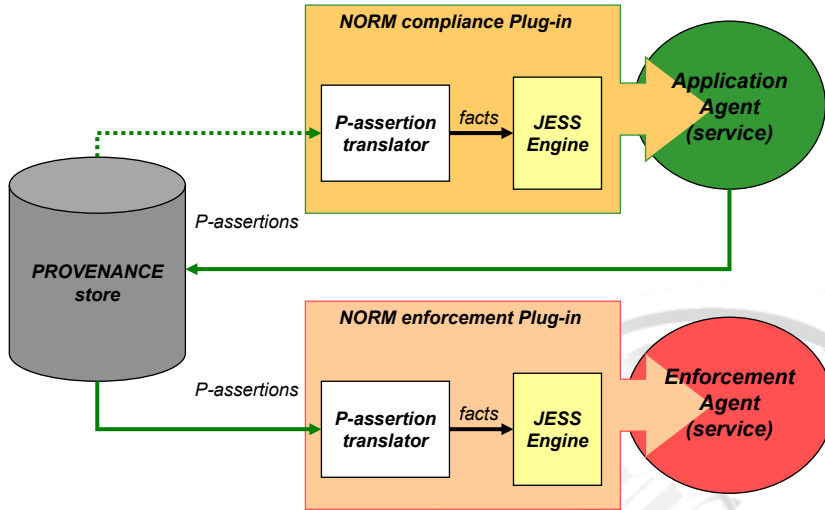
*Asserted(echr1, echr1 DO DBaccess(echr1,db1,query\_data)*

- Actor-state p-assertions to trace events and statements

*Asserted(ddca2, based\_on(decision3, {serology\_test\_result27, DHLA\_test 14})*

*Asserted(ddca2, decision\_result(decision3, donate\_all\_organ))*

## Norm enforcement through provenance (II)



## Conclusions and ongoing work

## Conclusions and future work

- The agent paradigm is still a good way to model distributed applications
- Sometimes agent platforms are not the best implementation solution
  - Some users still do not trust the technology
- Service Oriented Architectures are a good alternative
- Our experience: OTM application for **Distributed allocation of human organs** for transplantation purposes
- How to adapt eInstitution mechanisms to SOA?
  - SOA governance
  - SOA provenance
- Currently developing eInstitution enforcement mechanisms over standard SOA mechanisms.

31



<http://www.lsi.upc.es/~jvazquez>

32