Developing an Intelligent Parking Management Application based on Multi-Agent Systems and SemanticWeb Technologies

Andrés Muñoz
amunoz@um.es

Juan A. Botía
juanbot@um.es

University of Murcia
SPAIN

Motivation

- Exploring the combination of Semantic Web & Argumentation technologies in MAS to:
  - Represent and reason with knolwedge (KR&R)
  - Solve conflicts of knowledge

- Applying this approach to a real scenario through an intelligent parking management application:
  - SEISCIENTOS project

http://www.grc.upv.es/600/

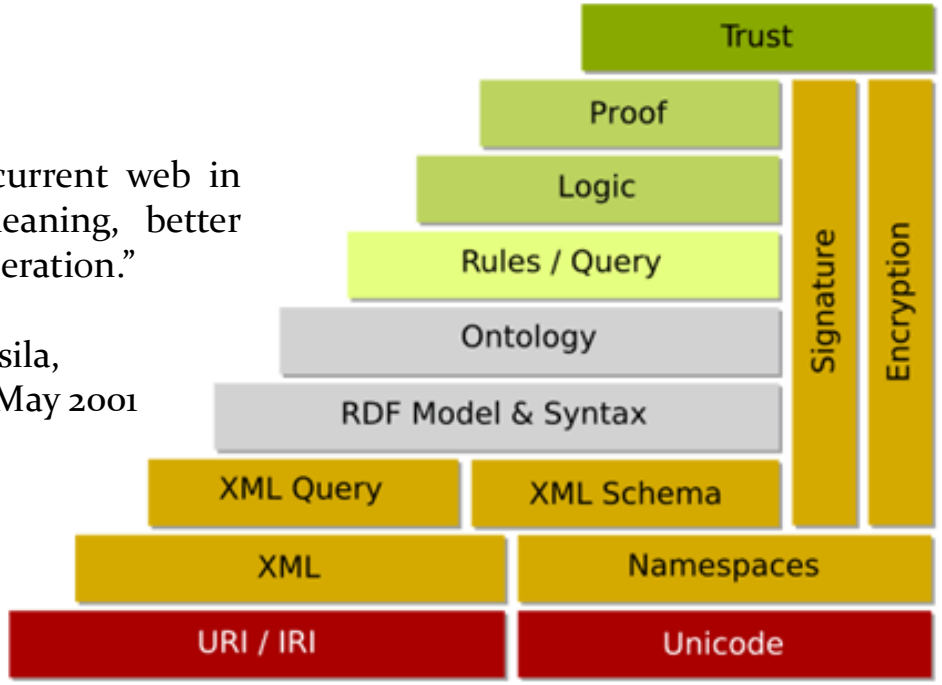
Outline

- Semantic Web Background
- An Architecture Based on Semantic Web Technologies to Manage Knowledge in MAS
  - Conflicts of Knowledge & Argumentation
  - Solving Conflicts of Knowledge through ASBO
- An Intelligent Parking Management Application
- Conclusion and Future Work

Semantic Web Background

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

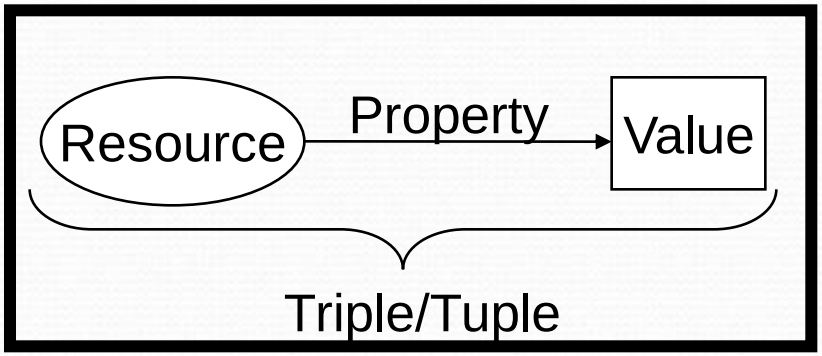
Tim Berners-Lee, James Hendler, Ora Lassila,
The Semantic Web, Scientific American, May 2001

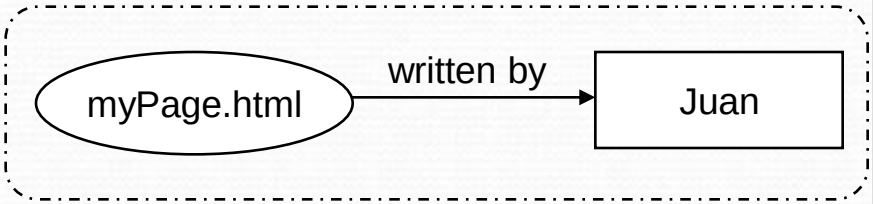


- Basic Idea: Add metadata to World Wide Web documents in order to enable computers to process information.

Semantic Web Background

RDF





Ontology Layer (OWL)

- *Domain Model*:
  - Concepts, relationships between them, axioms and individuals.
  - *Formal*, so it can be processed by computers.
  - Easily *shareable* and *reusable*.
  - *Open*, it represesents an incomplete and extensible view on the domain.

Rules Layer (SWRL/RIF)

$$\text{father}(?x,?y) \wedge \text{brother}(?y,?z) \rightarrow \text{uncle}(?x,?z)$$

Semantic Web Background

Ontology Language

OWL

- ✓ Based on Description Logic (First-Order Logic subset)
- ✓ RDF/XML Syntax
- ✓ Ontology models divided into TBox/ABox

Reasoning Capabilities

- Discover new information about concepts and individuals

  Man ⊑ Person

  ⇒ Person(Andres)

  Man(Andres)

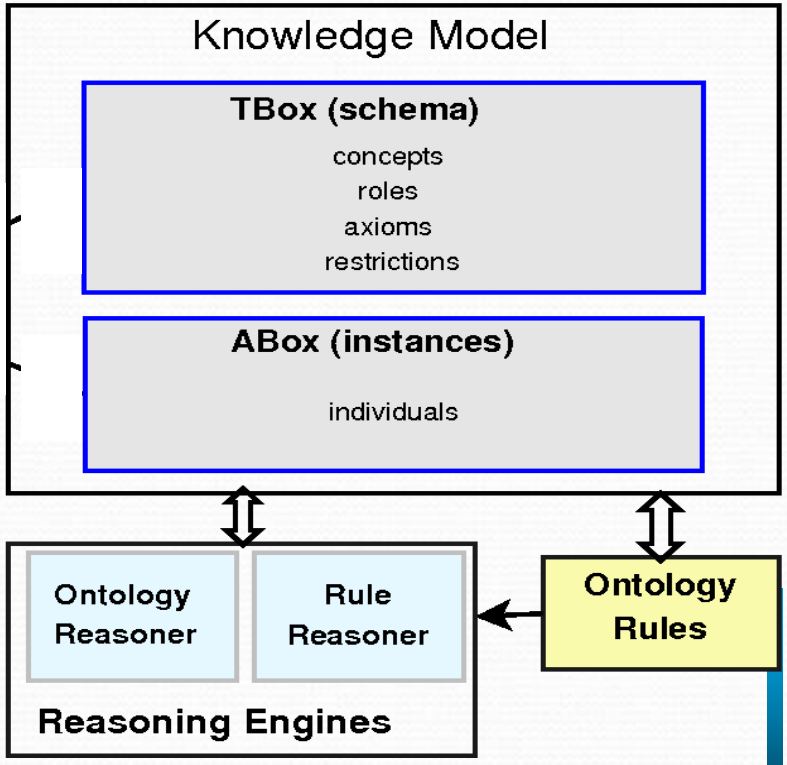
- Check model consistency:

  Man ⊓ Woman ≡⊥

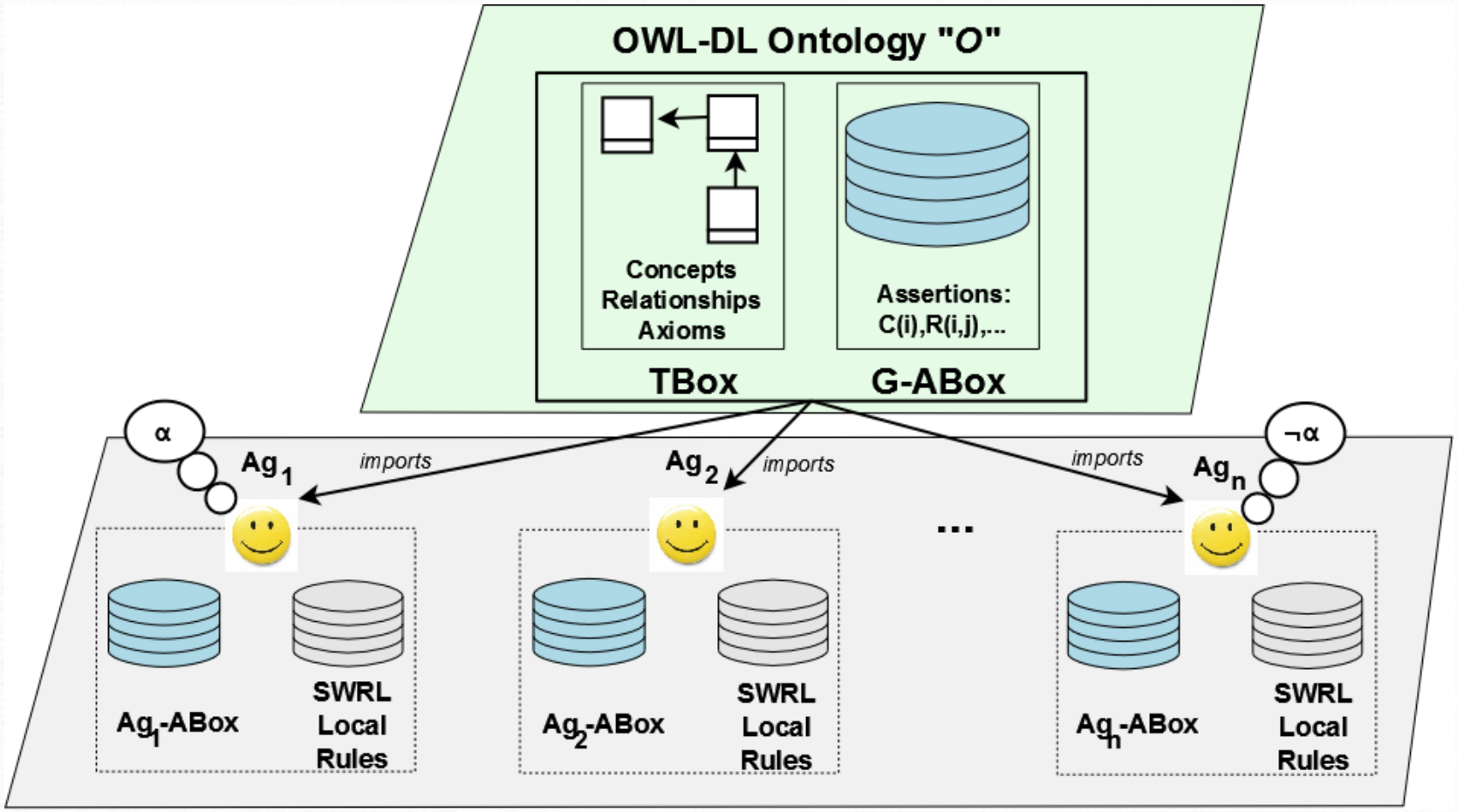
  ⇒ # Inconsistency!!

  {Man(Andres), Woman(Andres)}

- Rule-based reasoning



An Architecture Based on Semantic Web Technologies to Manage Knowledge in MAS

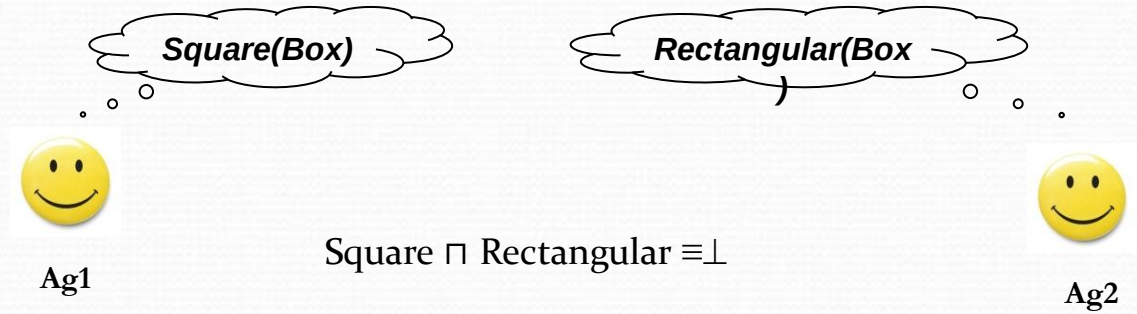


Conflicts of Knowledge & Argumentation

Conflicts of Knowledge: Two types

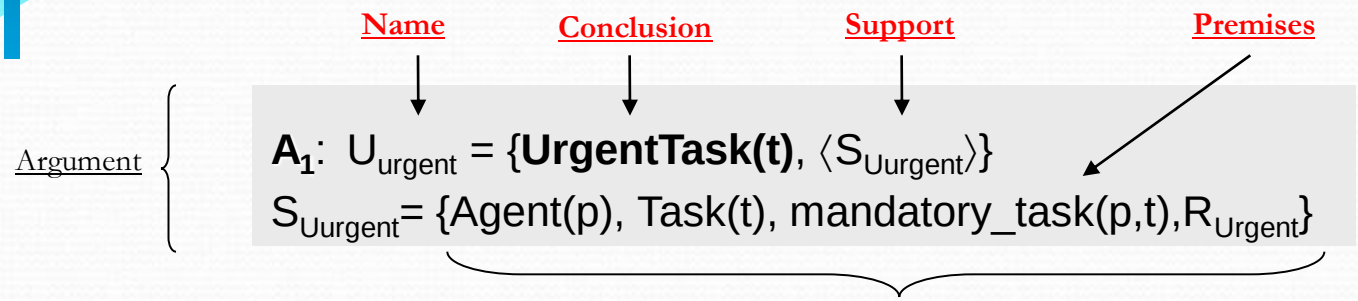
Contradictions: Appear independently of the domain modelled in the system → A positive and negative assertion on the same information.

Violation of restrictions through differences: Tightly related to a specific domain and have no effect out of it.



Square ⊓ Rectangular ≡⊥

Conflicts of Knowledge & Argumentation



R_{Urgent} = *(Agent(?x), Task(?t), mandatory_task(?x,?t))* → *UrgentTask(?t)*

- **Conflicts**: Attacks among arguments U = {α, ⟨S_u⟩} V = {β, ⟨S_v⟩}
  - Rebutting: An argument U rebuts an argument V iff (α, β) are inconsistent.

A_2: $U_{trivial}$ = {¬**UrgentTask(t)**, ⟨$S_{Utrivial}$⟩}
$S_{Utrivial}$ = {Agent(p), Task(t), recommended_task(p,t),$R_{trivial}$}

$R_{Trivial}$ = *(Agent(?x), Task(?t), recommended_task(?x,?t))* → *¬UrgentTask(?t)*

  - Undercutting: An argument A undercuts an argument B iff (α,S_V) are inconsistent.

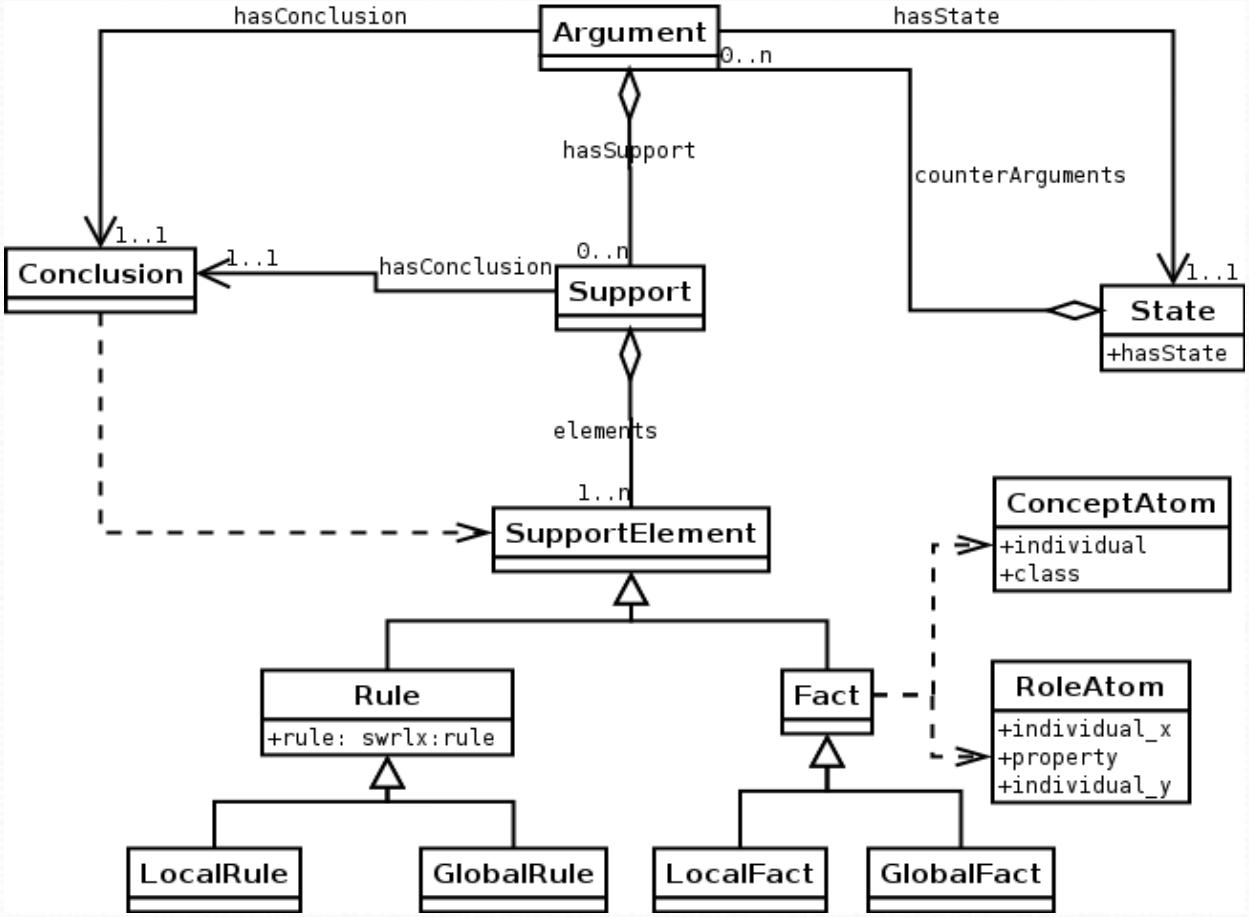
Ag_2: $U_{\neg mandatory}$ = {¬**mandatory_task(p,t)**, ⟨$S_{U\neg mandatory}$⟩}
$S_{U\neg mandatory}$ = …

Conflicts of Knowledge & Argumentation

- **Defeat between arguments**
  - Let U1,U2 be two arguments. U1 ***defeats*** U2 iff:
    - U1 undercuts U2; or
    - U1 rebuts U2 and U1 is *preferable* to U2.
  - U1 ***strictly defeats*** U2 iff U1 defeats U2 and U2 does not defeat U1.
- **Acceptability status**
  - An argument can be classified in one of the *acceptable, non-acceptable (defeated),* or *unknown* state.
  - To set the status of any argument, it is needed a process that takes into account not only conflicting arguments, but all the relevant ones.
  - The status is established by means of a ***persuasion dialogue***

Solving Conflicts of Knowledge through ASBO

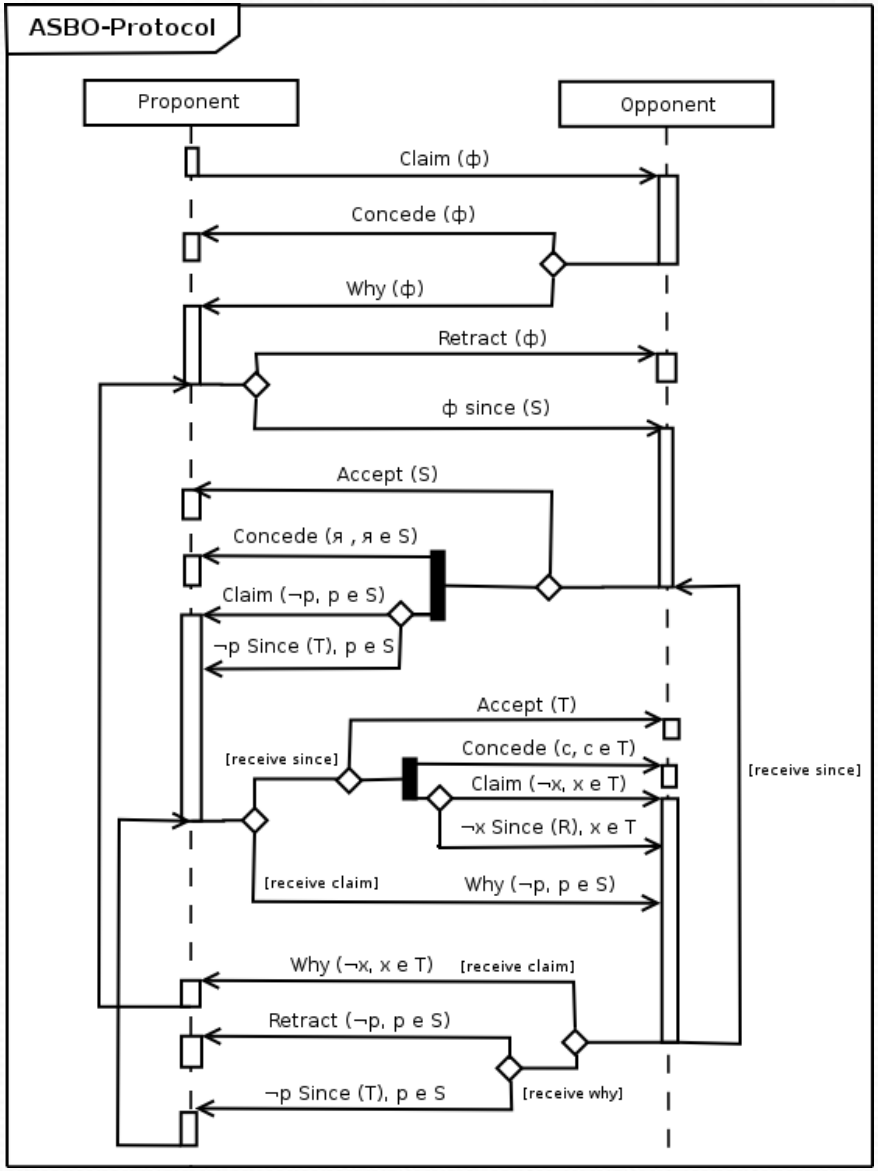
ASBO: Argumentation System Based on Ontologies



Argument Representation in OWL-DL

A. Muñoz and J. A. Botía. *ASBO: Argumentation System Based on Ontologies*. Cooperative Information Agents XII, volume 5180 of LNAI, pages 191–205. Springer, 2008.

Solving Conflicts of Knowledge through ASBO

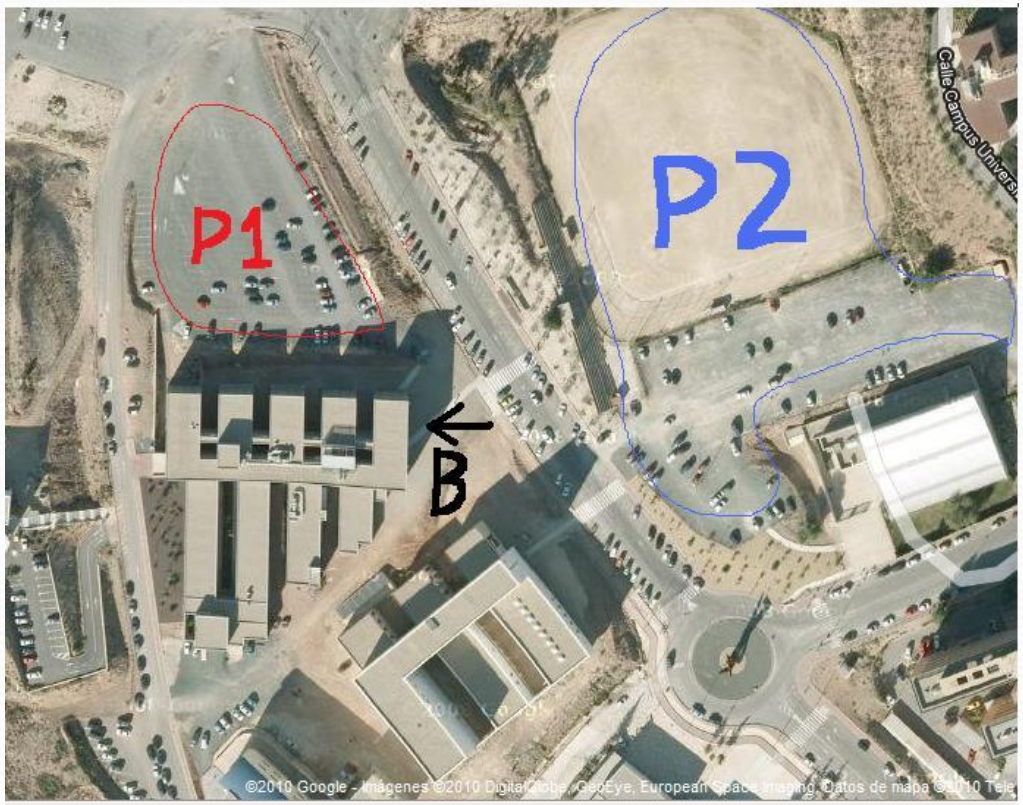


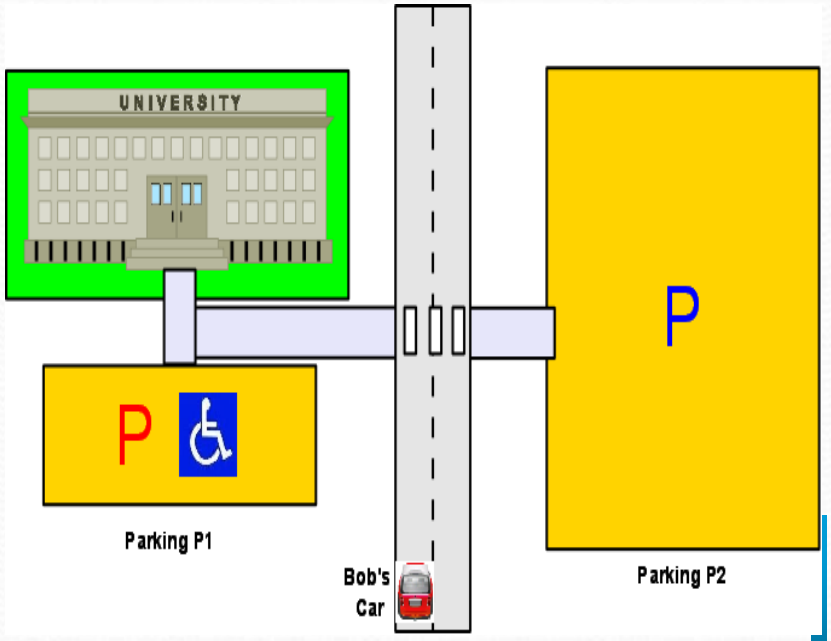
Argumentation Dialog

A. Muñoz and J. A. Botía. *A Formal Model of Persuasion Dialogs for Interactions among Argumentative Software Agents*. Journal of Physical Agents, 3(3), 2009.

An Intelligent Parking Management Application

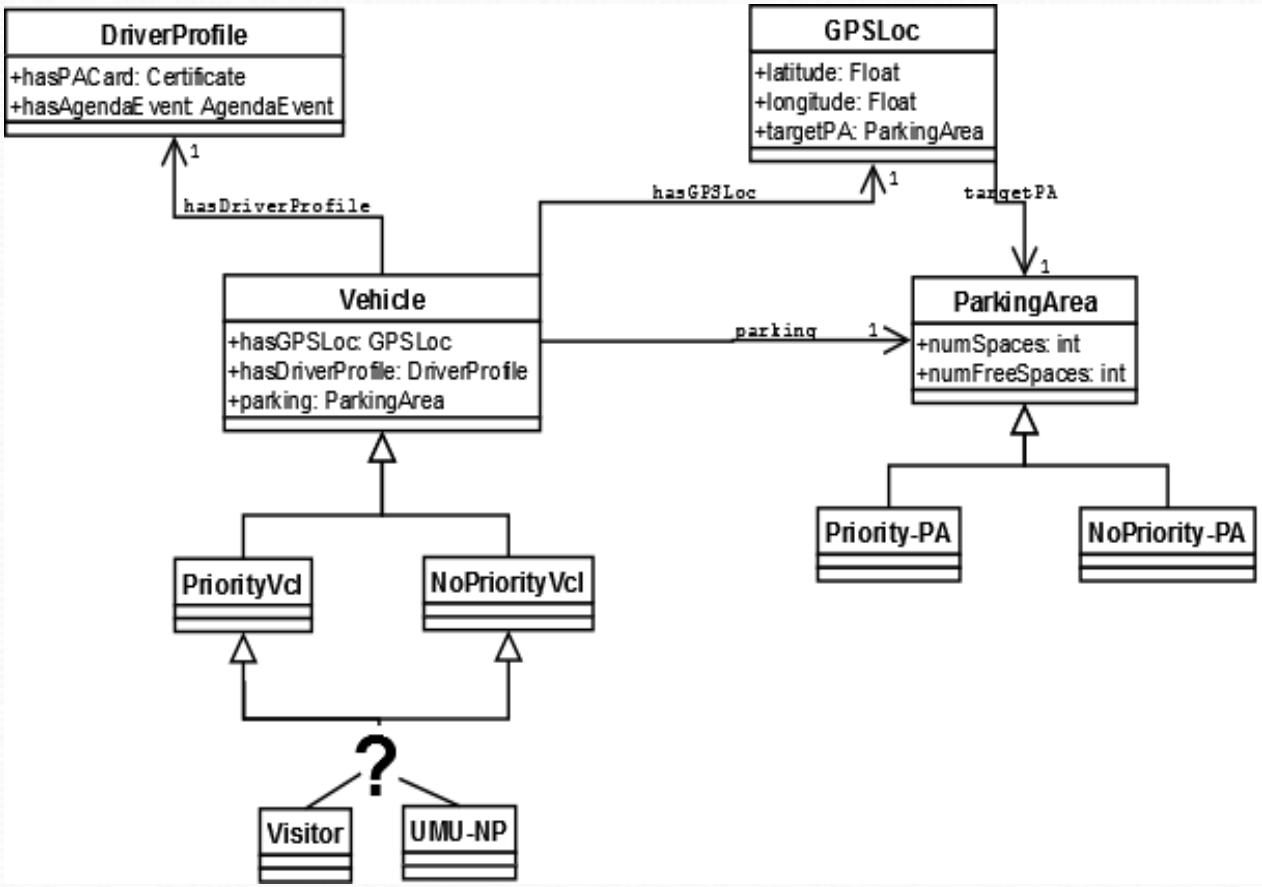
- Scenario developed in School of Computer Science at UMU campus
- Prototype: car with RFID tag and RFID reader in parking barrier



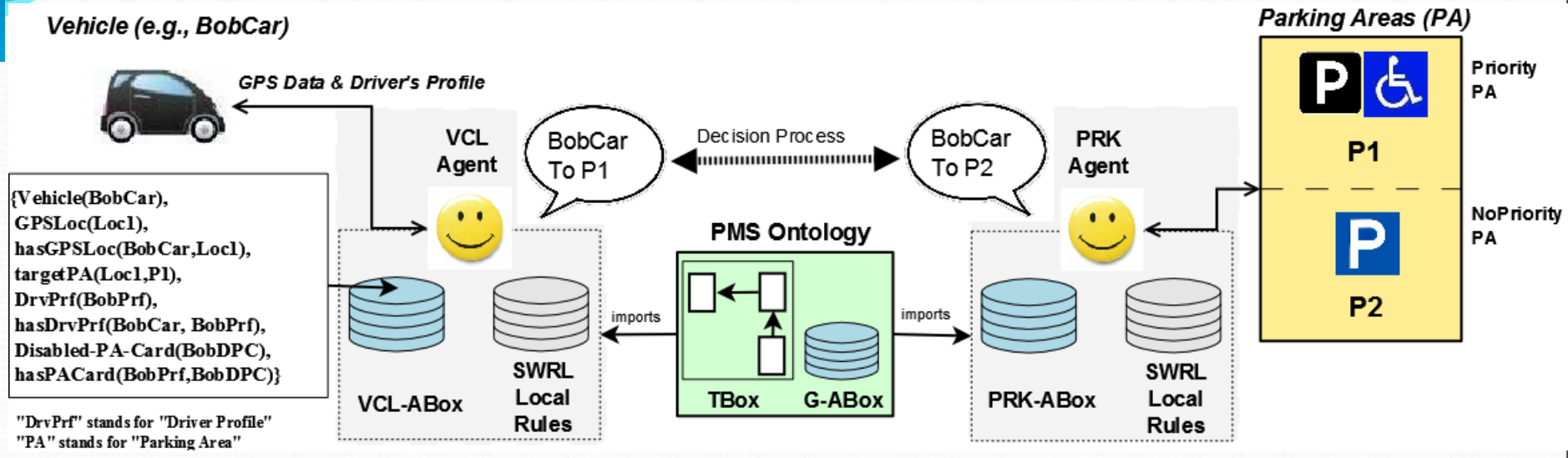


An Intelligent Parking Management Application

- Domain Model represented as an OWL-DL ontology → **PMS Ontology**
- Represented here as a UML diagram.
- Visitor and UMU-NP vehicles classification as PriorityVcl or NoPriorityVcl depends on agents policies



An Intelligent Parking Management Application



- **G-ABox** = {PA(P1), PA(P2), latitude(P1, 18.36º), ...}
- **VCLAgent:**
  - VCL-ABox shown in figure.
  - Local Rules:

$$\mathbf{R}_{\text{\_GPSParking}}: \text{Vehicle}(?v) \wedge \text{GPSLoc}(?g) \wedge \text{PA}(?p) \wedge \text{hasGPSLoc}(?v, ?g) \wedge \text{targetPA}(?g, ?p) \Rightarrow \text{parking}(?v, ?p)$$

$$\mathbf{R}_{\text{\_Disabled}}: \text{Visitor}(?v) \wedge \text{DrvPrf}(?d) \wedge \text{Disabled-PA-Card}(?c) \wedge \text{hasDrvPrf}(?v, ?d) \wedge \text{hasPACard}(?d, ?c) \Rightarrow \text{PriorityVcl}(?v)$$

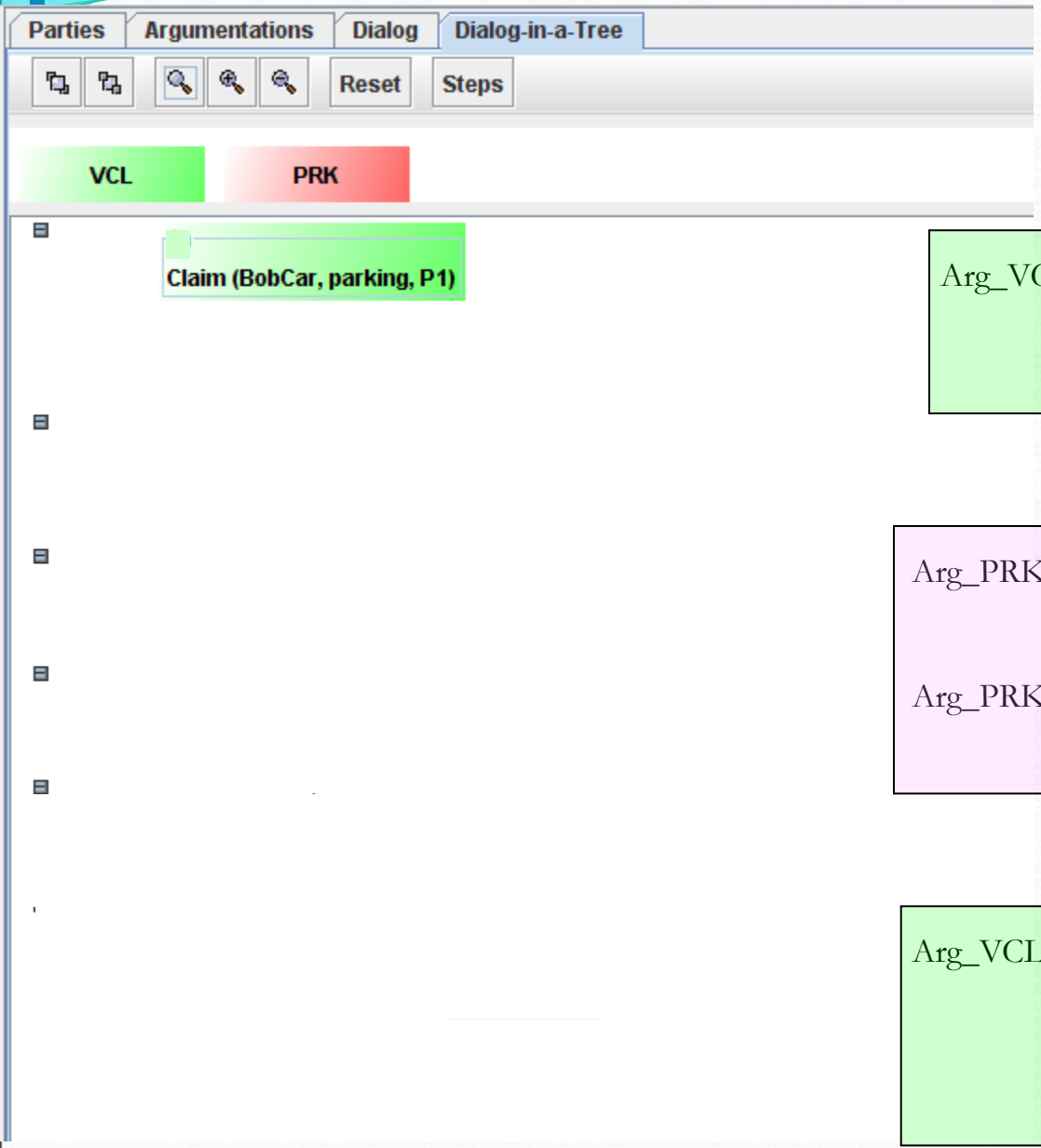
- **PRKAgent:**
  - PRK-ABox = {Priority-PA(P1), NoPriority-PA(P2), numFreeSpaces(P1, 10), ...}
  - LocalRules:

$$\mathbf{R}_{\text{\_ Priority-PA}}: \text{PriorityVcl}(?v) \wedge \text{Priority-PA}(?p) \Rightarrow \text{parking}(?v, ?p)$$

$$\mathbf{R}_{\text{\_NoPriority-PA}}: \text{NoPriorityVcl}(?v) \wedge \text{NoPriority-PA}(?p) \Rightarrow \text{parking}(?v, ?p)$$

$$\mathbf{R}_{\text{\_Visitor}}: \text{Visitor}(?v) \Rightarrow \text{NoPriorityV lc}(?v)$$

An Intelligent Parking Management Application



Arg_VCL1=[**parking(BobCar, P1)**, {Vehicle(BobCar), GPSLoc(Loc1), PA(P1), hasGPSLoc(BobCar, Loc1), targetPA(Loc1, P1), $R_{\_GPSParking}$}]

Arg_PRK1=[**parking(BobCar, P2)**, {NoPriorityVcl(BobCar), NoPriorityPA(P2), $R_{\_NoPriority\text{-}PA}$}]

Arg_PRK2=[**NoPriorityVcl(BobCar)**, {Visitor(BobCar), $R_{\_Visitor}$}]

Arg_VCL2=[**PriorityVcl(BobCar)**, {Visitor(BobCar), DrvPrf(BobPrf), Disabled-PA-Card(BobDPC), hasDrvPrf(BobCar, BobPrf), hasPACard(BobPrf,BobDPC), $R_{\_Disabled}$}]

Conclusion and Future Work

- MAS and SemanticWeb technologies can be combined giving as a result an architecture to automatically manage knowledge in distributed environments.
- Appearance of conflicts is an inherent problem of such environments
  - The architecture is extended with an argumentation system called ASBO which enables agents to rationally deal with conflicts.
- In this work we exploit the integration of all these technologies to develop an intelligent parking management application.
- Future work is directed to develop new situations in the PMS, evaluate the performance of the proposed architecture in this application and extend its usage in other applications.

- ASBO Implementation: ORE-AS tool

http://sourceforge.net/projects/ore-as/

Developing an Intelligent Parking Management Application based on Multi-Agent Systems and SemanticWeb Technologies

Thank you for your attention!

Any Questions?