

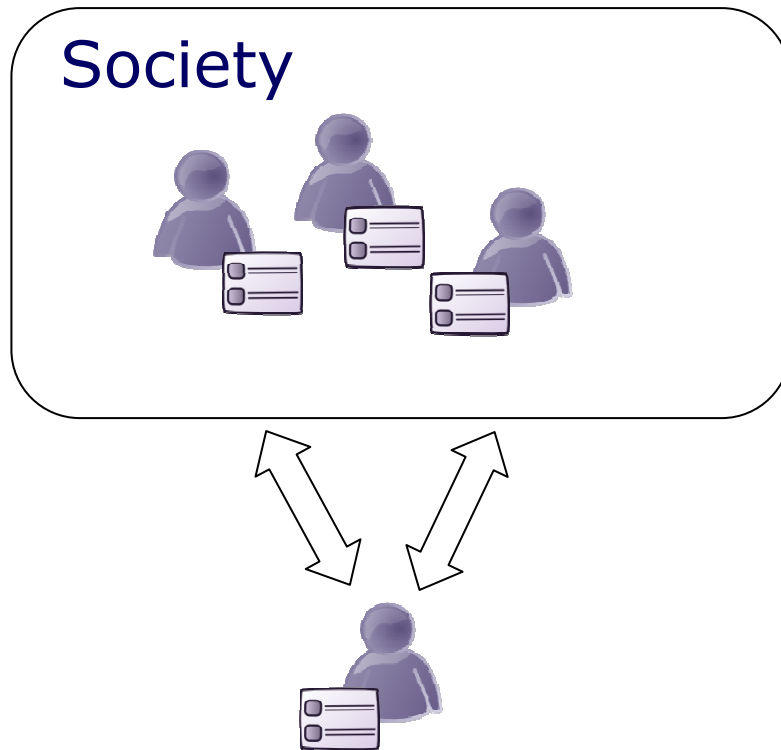
Interaction Protocols and Capabilities: a preliminary report

M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella

Dipartimento di Informatica – Università degli Studi di
Torino

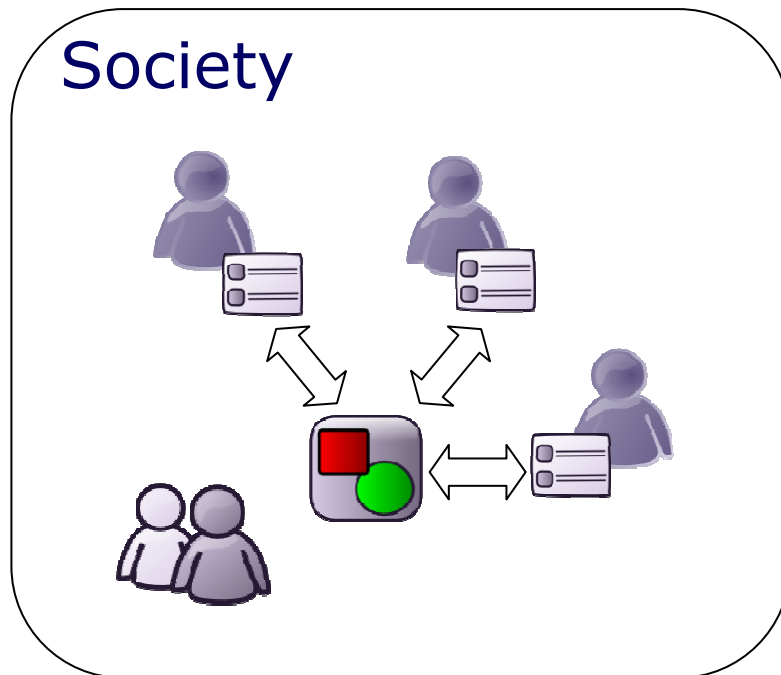
Corso Svizzera 185, 10149 Torino (Italy)

Interoperability



- Heterogenous and independent entities want to execute a shared task
- Interoperability is the capability of an entity of interacting with others
- Each entity must verify interoperability with other participants

Interoperability



- We can introduce a description of the overall behaviour



Choreography

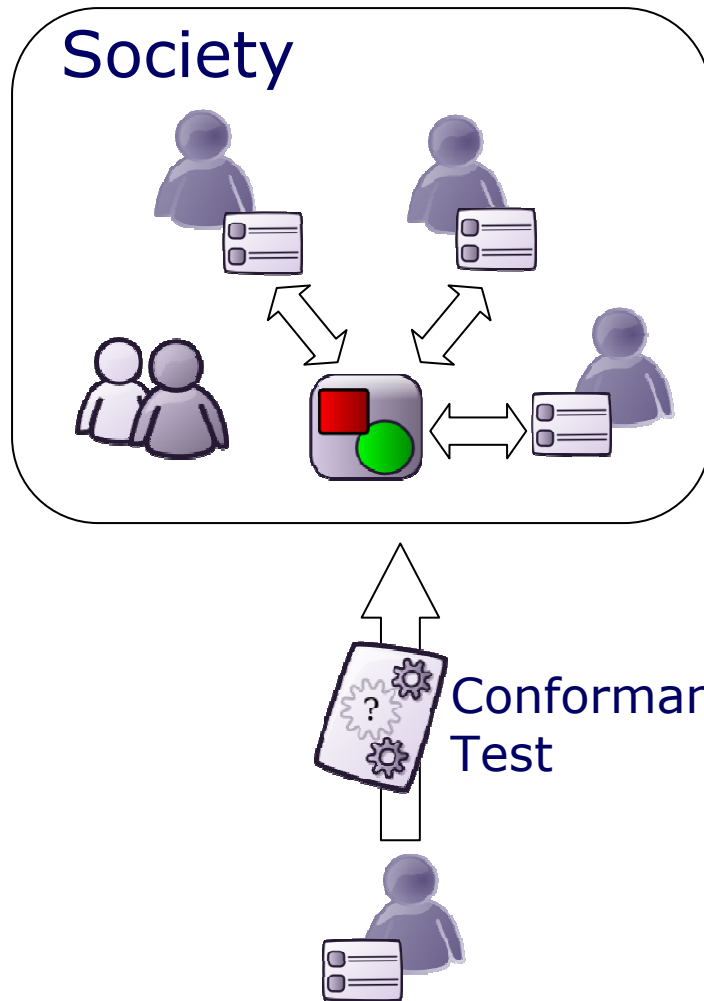
(Interaction protocol):
global point of view by
means of

- roles
- messages exchanged



Policy: local point of view
of a single entity
(orchestration)

Checking interoperability



- An entity that conforms to a protocol produces a legal (complete) conversation
- Conformance test entails a priori interoperability
- See Baldoni et al.
 - Agents: [Clima V, Clima VI]
 - Web Services: [WS-FM 05]

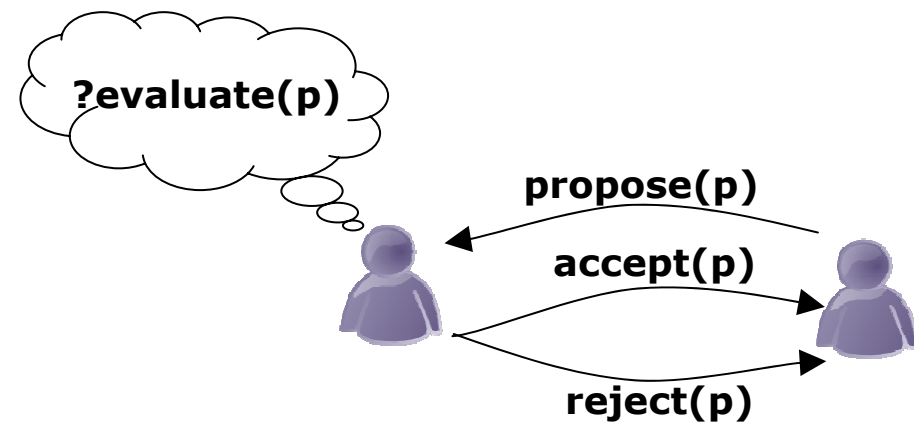
Interoperability

What happens if the entity has no valid policy (for example, the conformance test fails) ?

- (1) It may ask to other entities a correct interaction policy for the role that it wants to play
 - CooBDI, CooWS [Bozzo et al., ICWI 2005]
 - Knowledge exchange in DALI language [Costantini, Tocchio, WOA 05]
- (2) It may synthesize a conformant policy from a high-level description of the interaction

Interoperability

- Interaction protocols only concern roles and communicative behaviour
- But an entity that wants to play a specific role must also execute actions that do not only concern communication
 - e.g.: producing a proposal, processing some data, checking if a product is in the store

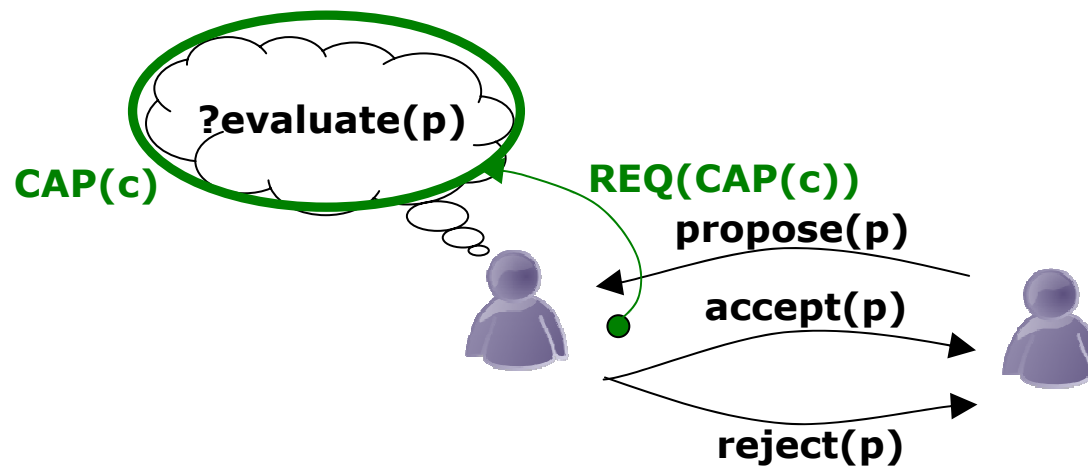


Capability requirements

- We can enrich Interaction Protocols with an high-level description of the actions that each entity must be able to execute if it wants to play a role
- The entity must own an implementation of these actions in order to synthesize in a semi-automatic way an executable policy
- We call these skills **capability requirements**

Capability requirements

- The term “capability” has been used by Padgham in the BDI framework
- We propose the extension of Interaction Protocols/choreographies with the notion of “requirement for a capability”

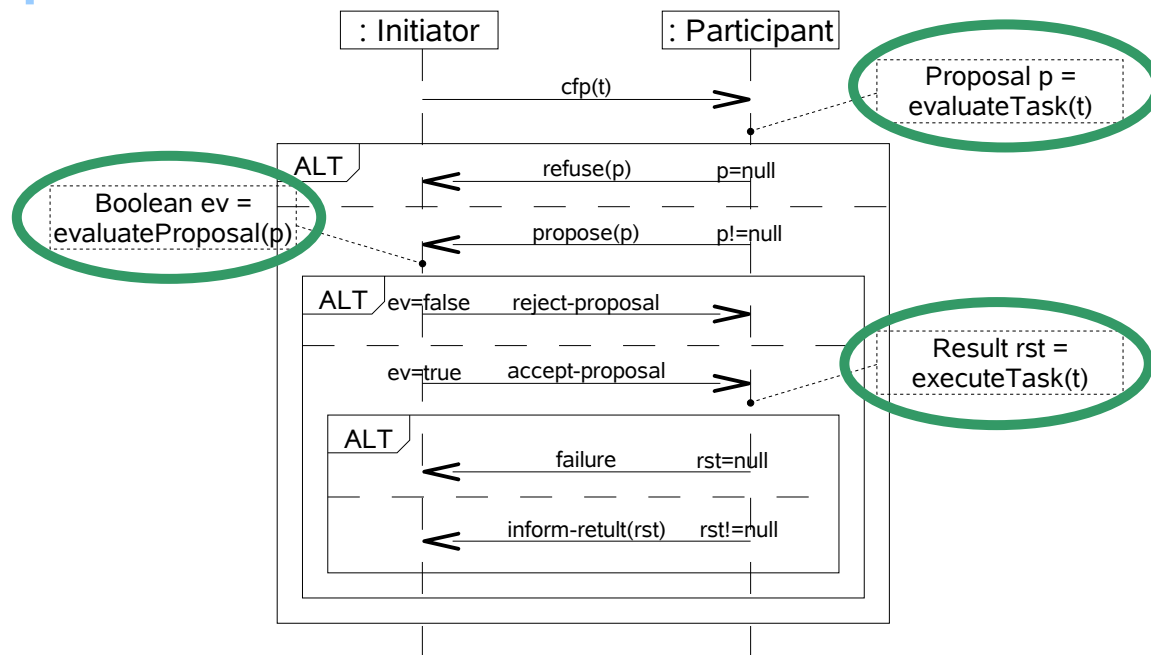


Capability requirements

- In literature we can find a similar concept:
 - Jade Platform [Bellifemine, Poggi, Rimassa]
 - PowerJava [Baltoni, Boella, Van der Torre - ProMAS 2005, SAC 2006]

Interaction Protocols and Capabilities: an example

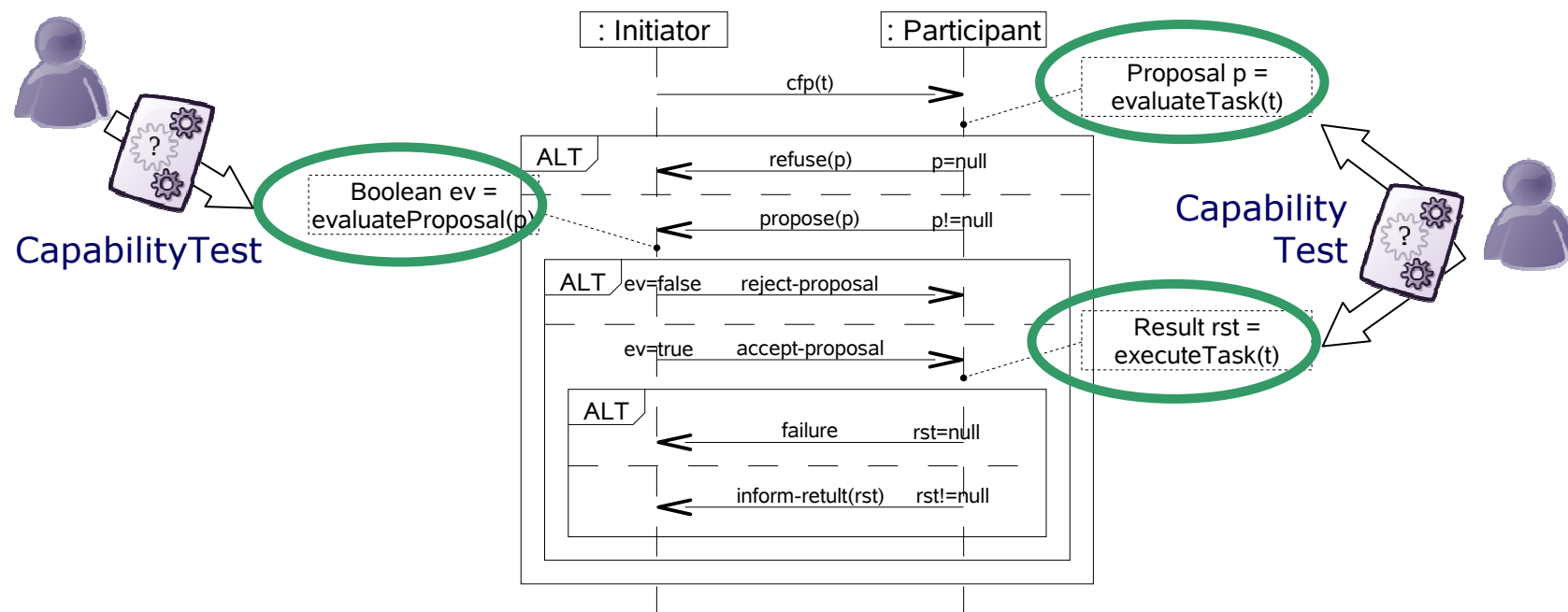
Fipa ContractNet Protocol



- The initiator must be able to **evaluate a proposal**
- A participant must be able to **evaluate a task** and **execute a task**

Capability test

How an entity can establish if it is be able to synthesize a policy for a specific role?



Checking capabilities

Different matching techniques can be used in the capability test

- Signature matching
 - simple
 - not-flexible
 - used also in PowerJava
- Semantic matchmaking
 - developed for semantic Web Services discovery
 - based on ontologies of concepts
 - support matching between different names and numbers/types of input/output parameters

Checking capabilities

Semantic matchmaking approaches:

- based on DAML-S proposed by Paolucci et al.
 - ontological reasoning is applied to input and output parameters
 - search is not goal-driven

Checking capabilities

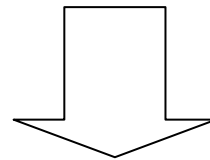
Semantic matchmaking approaches:

- Web Services Modelling Ontology (WSMO)
 - services are described by preconditions, assumptions, effects and postconditions (“capability” construct)
 - users can look for a service by specifying a goal described by means of the desired preconditions

Checking capabilities

WSMO can be used in our approach:

- a capability requirement inside an Interaction Protocol can be represented as a WSMO goal
- actions owned by an entity can be described by means of a WSMO capability construct



We can apply existing matching techniques in
capability test

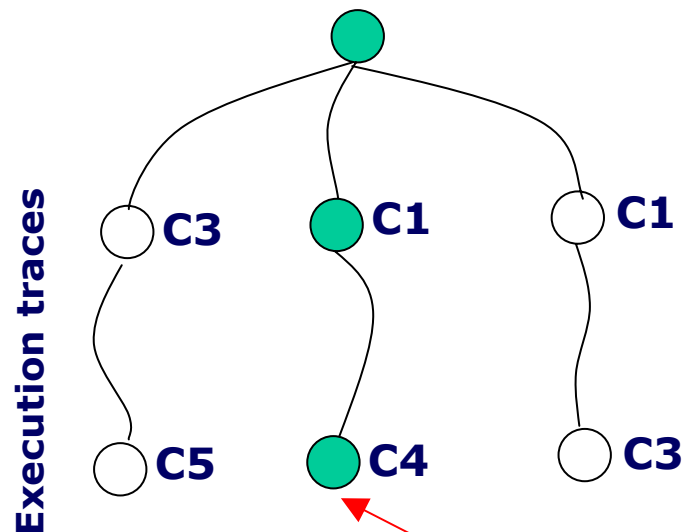
Reasoning on capabilities

Let us consider the interaction from the perspective of a given role, the role that the entity wants to play

- Must the entity synthesize a policy that implements **all** paths foreseen by it?
- Must the entity have **all** the capabilities required for the role?

Reasoning on capabilities

It is possible to perform some forms of customization



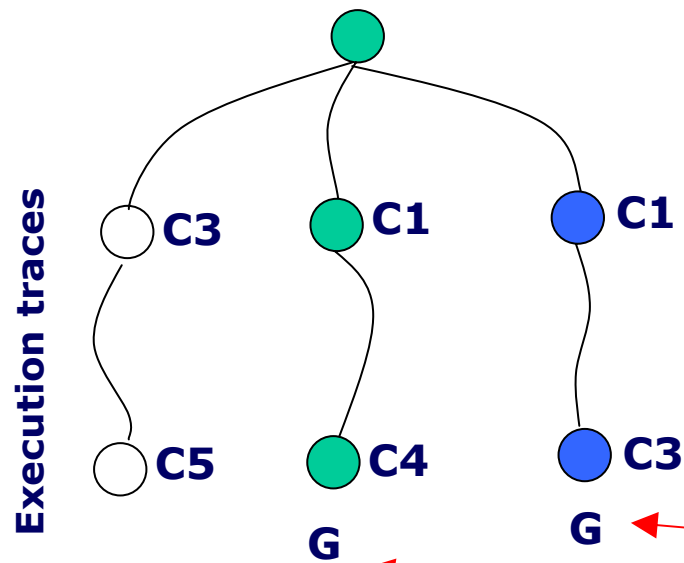
- An entity can find paths that contains only the capabilities owned by it.

Entity has only capabilities C1, C4 and C5: OK!

Reasoning on capabilities

It is possible to perform some forms of customization

- An entity can apply reasoning (e.g. Procedural planning) to choose paths that allow to reach a given goal and that contain only the owned capabilities



Entity wants to reach goal G and has only capabilities C1, C4 and C5: OK!

Capability requirements in WS-CDL

- Extension of the WS-CDL language that includes capability requirements representation
- Capabilities represent operations performed by an entity which are non-observable by other entities, like SilentAction elements in WS-CDL
- Capability requirements are expressed by means of input and output parameters with ontological relations

Capability requirements in WS-CDL

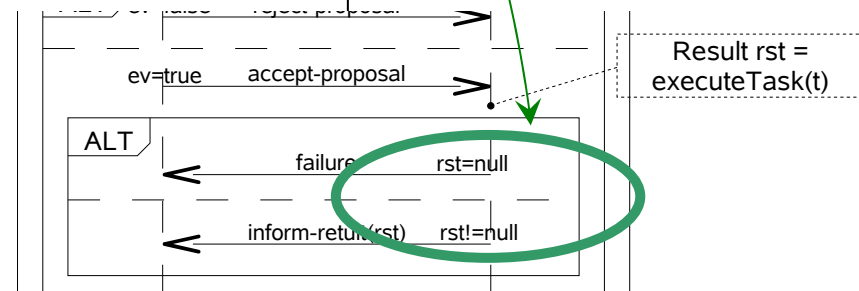
- New tag "capability" inside "silentAction" tag
- Each parameter refer to a variable defined in the choreography document

```
<silentAction roleType="Participant">
  <capability name="evaluateTask">
    <input>
      <parameter
variable="cdl:getVariable('tns:t','','')"/>
    </input>
    <output>
      <parameter
variable="cdl:getVariable('tns:p','','')"/>
    </output>
  </capability>
</silentAction>
```

Capability requirements in WS-CDL

- Input and output parameters can be used in the whole documents in standard ways (Interaction, Workunit, etc)

```
<choice>
  <workunit name="informResultWorkUnit"
    guard="cdl:getVariable('tns:rst', '', '',
'tns:Participant') != 'null' ">
    <interaction name="informResultInteraction">
      ...
    </interaction>
  </workunit>
  <interaction name="failureExecuteInteraction">
    ...
  </interaction>
</choice>
```



Conclusions

This approach

- extends the specification of Interaction Protocols by means of “requirements of capabilities”
- allows an entity to improve its interoperability by synthesizing a new policy in a semi-automatic way
- permits entities to exploit reasoning techniques for customizing the policy synthesis

Future works

- More thorough formalization of the proposal
- Integration of the WSMO approach in
 - checking capabilities
 - reasoning techniques
- Design and implementation of a system in order to check the feasibility of the proposed approach



Thank you!