eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Semantic Web Technology for Agent Communication Protocols

Idoia Berges

J. Bermúdez, A. Goñi and A. Illarramendi

Grupo BDI. UPV-EHU.

5th European Semantic Web Conference.

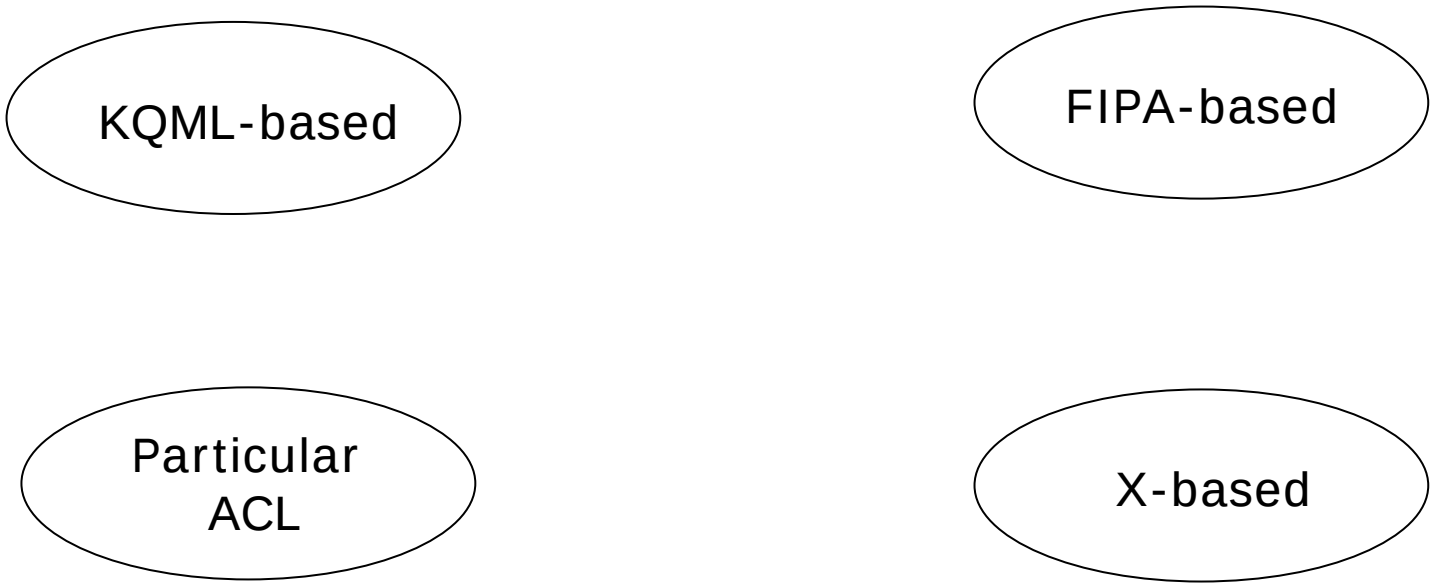
Tenerife. Spain.

Outline

- Motivation and goal
- Ontology for communication acts
- Protocol descriptions
- Relationships between protocols
- Conclusions

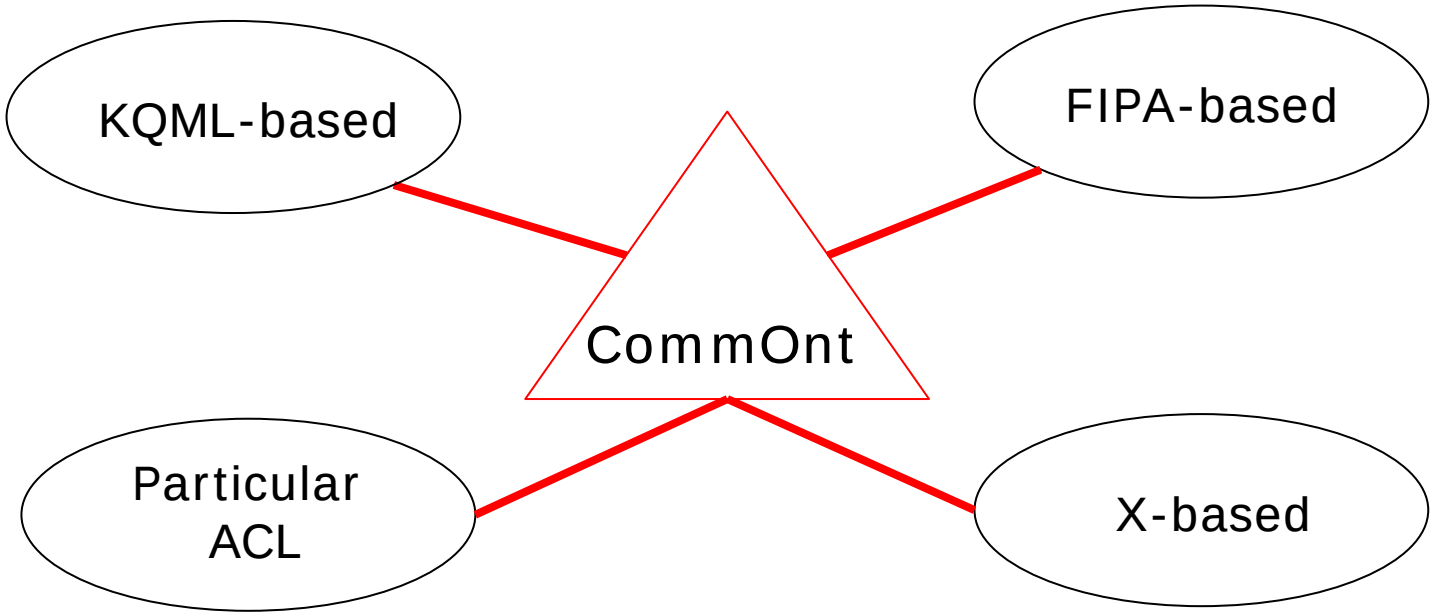
Motivation

Right Now!



Motivation

Where we want to arrive at!



Exchangeability of communication acts

Motivation

- Exchangeability of communication acts is not enough
- Sharing of communication protocols is needed

Proposal: Goal

- *Referential* representations for
  - communication acts
  - communication protocols

  using Semantic Web technology.

Proposal: Contributions

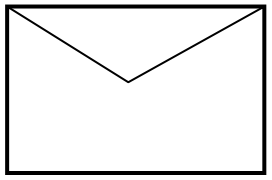
- To favour a flexible agent interoperation.
- To facilitate customization of communication protocols.
- A basis for reasoning about protocol relationships.
- Take account of semantics in protocol representations.

Outline

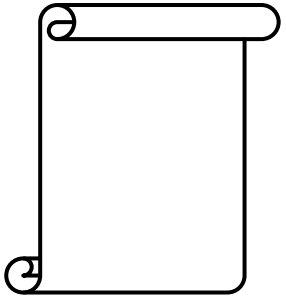
- Motivation and goal
- Ontology for communication acts
- Protocol descriptions
- Relationships between protocols
- Conclusions

CommOnt design criteria(1)

- Speech Acts Theory
  - A communication act is basically composed of an envelope and a content



Intention + communication information



Object of the intention

CommOnt design criteria(2)

- Social commitments approach
  - Objective and verifiable semantics
  - Formalization:
    - Commitment
      - C(x, y, p)
    - Conditional commitment
      - CC(x,y,c,p)
      - CC(x,y,c,p) ∧ c → C(x,y,p)

CommOnt design criteria(3)

- Axiomatization of communication acts with Event Calculus:
  - First-order theory for reasoning about actions
  - Events (actions) initiate and terminate fluents
  - Fluents are propositions whose value is subject to change over time

Event Calculus predicates

1. *Initiates*(a, f, t) means that f holds after event a at time t.
2. *Terminates*(a, f, t) means that f does not hold after event a at time t.
3. *Initially*$_P(f)$ means that f holds from time 0.
4. *Initially*$_N(f)$ means that f does not hold from time 0.
5. *Happens*(a, t_1, t_2) means that event a starts at time t_1 and ends at t_2.
6. *HoldsAt*(f, t) means that f holds at time t.
7. *Clipped*(t_1, f, t_2) means that f is terminated between t_1 and t_2.
8. *Declipped*(t_1, f, t_2) means that f is initiated between t_1 and t_2.

From Pınar Yolum and Munindar P. Singh

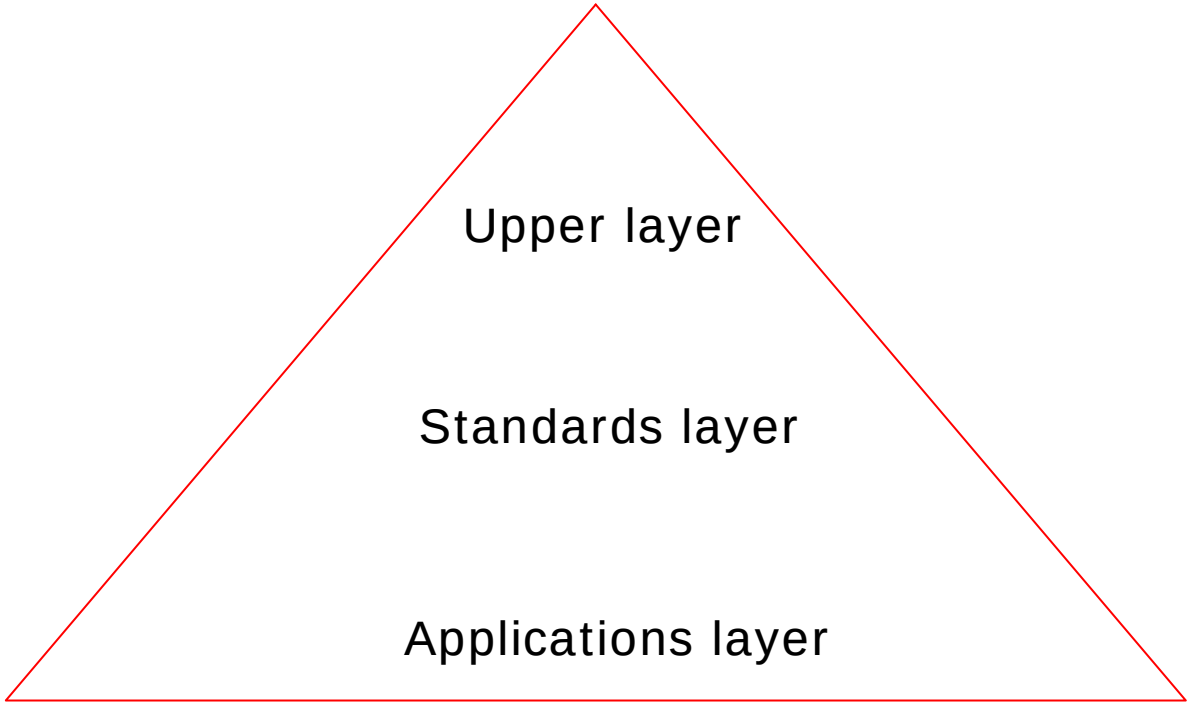
Rules

- To capture the dynamics of commitments

$$HoldsAt(\mathsf{C}(x, y, p), t) \wedge Happens(e(x), t) \wedge Initiates(e(x), p, t) \rightarrow$$
$$Terminates(e(x),\mathsf{C}(x, y, p), t).$$

CommOnt design criteria(4)

- Materialization of CommOnt: OWL ontology



CommOnt upper layer

$$\texttt{CommunicationAct} \sqsubseteq \forall \texttt{hasSender.Actor} \sqcap =1\,\texttt{.hasSender} \sqcap \forall \texttt{hasReceiver.Actor} \sqcap \forall \texttt{hasContent.Content}$$

Main subclasses:
Assertive, Directive, Commissive, Expressive and Declarative

Other subclasses:

$$\texttt{Inquiry} \sqsubseteq \texttt{Directive}$$
$$\texttt{Request} \sqsubseteq \texttt{Directive}$$
$$\texttt{Responsive} \sqsubseteq \texttt{Assertive}$$

CommOnt upper layer

$$\texttt{Assertive} \equiv \texttt{CommunicationAct} \sqcap \exists \texttt{hasContent.Proposition} \sqcap \exists \texttt{hasCommit.Commitment}$$

$$\mathit{Initiates}(\mathit{Assertive}(s, r, P), \mathsf{C}(s, r, P), t))$$

$$\texttt{Directive} \equiv \texttt{CommunicationAct} \sqcap \exists \texttt{hasContent.Action} \sqcap \exists \texttt{hasCommit.ConditionalCommitment}$$

$$\mathit{Initiates}(\mathit{Directive}(s, r, P), \mathsf{CC}(r, s, \mathit{accept}(r, s, P), P), t))$$

CommOnt standards layer

$$\begin{aligned} \texttt{FIPA-Inform} &\sqsubseteq \texttt{Assertive} \\ \texttt{FIPA-Confirm} &\sqsubseteq \texttt{Assertive} \\ \texttt{FIPA-Disconfirm} &\sqsubseteq \texttt{Assertive} \\ \texttt{FIPA-Request} &\sqsubseteq \texttt{Directive} \end{aligned}$$

$$\begin{aligned} \texttt{KQML-Tell} &\sqsubseteq \texttt{Assertive} \\ \texttt{KQML-Ask-If} &\sqsubseteq \texttt{Inquiry} \sqcap \\ &\forall \texttt{hasContent}.(\forall \texttt{hasQuery}.\texttt{Proposition}) \end{aligned}$$

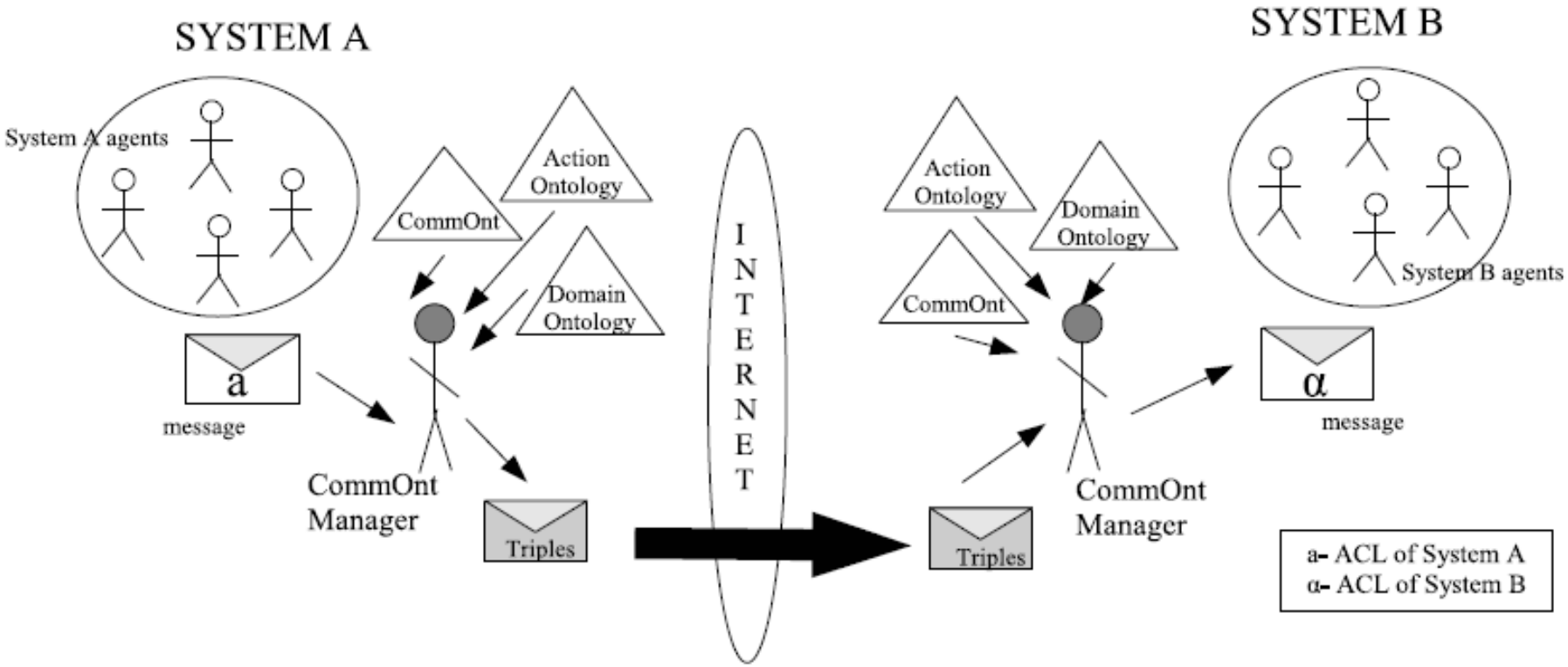
- Very important for the interoperability goal:

$$\begin{aligned} \texttt{FIPA-Inform} &\equiv \texttt{KQML-Tell} \\ \texttt{KQML-Achieve} &\sqsubseteq \texttt{FIPA-Request} \\ \texttt{KQML-Ask-If} &\equiv \texttt{FIPA-Query-If} \\ \texttt{KQML-Achieve} &\equiv \texttt{FIPA-Request} \sqcap \exists \texttt{hasContent}.\texttt{Achieve} \end{aligned}$$

CommOnt applications layer

$$
\begin{aligned}
\texttt{MedicineModify} &\equiv \texttt{Request} \sqcap {=}1.\texttt{hasContent} \sqcap \\
&\quad \forall \texttt{hasContent}.(\texttt{Overwrite} \sqcap \exists \texttt{hasSubject}.\texttt{Medicine}) \\
\texttt{LocationQuery} &\equiv \texttt{Inquiry} \sqcap {=}1.\texttt{hasContent} \sqcap \\
&\quad \forall \texttt{hasContent}.(\\
&\quad \forall \texttt{hasQuery}.(\texttt{RefExpression} \sqcap \exists \texttt{hasSubject}.\texttt{Location})) \\
\texttt{VitalSignInform} &\equiv \texttt{Responsive} \sqcap {=}1.\texttt{hasContent} \sqcap \\
&\quad \forall \texttt{hasContent}.(\texttt{Proposition} \sqcap \exists \texttt{hasSubject}.\texttt{VitalSignData})
\end{aligned}
$$

Communication process

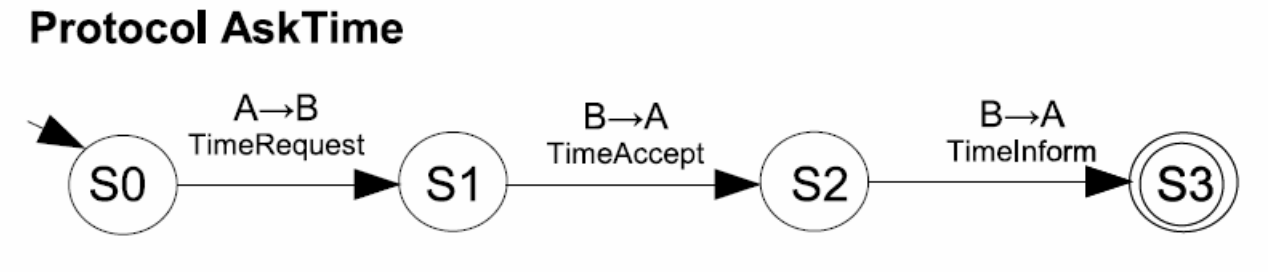


Outline

- Motivation and goal
- Ontology for communication acts
- Protocol descriptions
- Relationships between protocols
- Conclusions

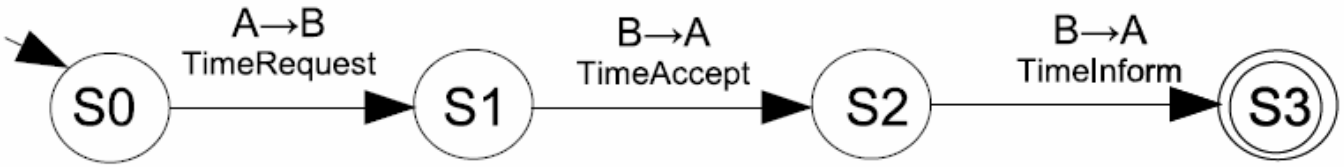
Protocol descriptions

- Model: State Transition System
  - Transitions labeled with classes of Communication Acts
  - States associated to sets of fluents



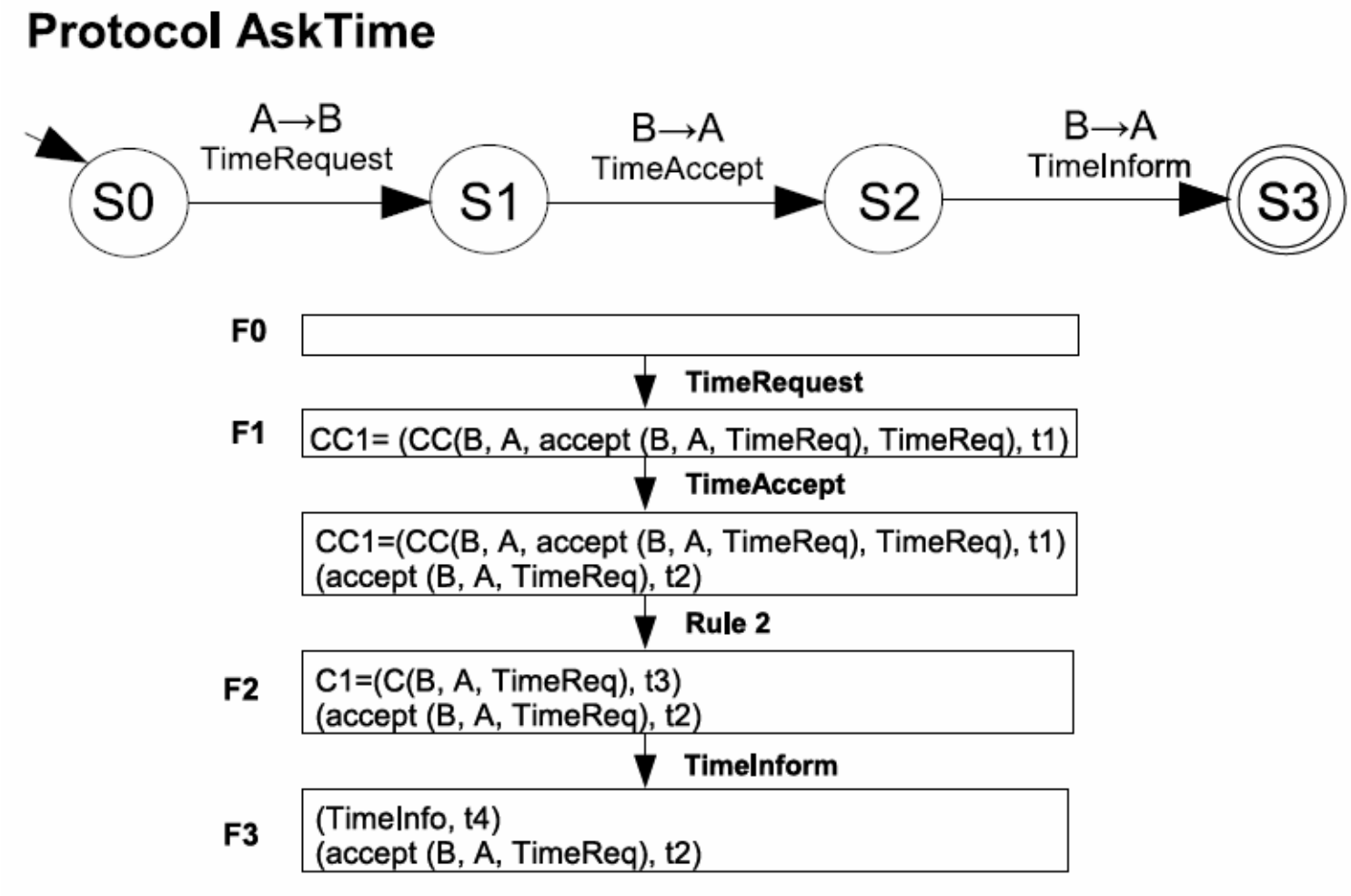
Protocol description

Protocol AskTime



$$
\begin{aligned}
\texttt{Asktime} &\equiv \texttt{Protocol} \sqcap \exists \texttt{hasInitialState.S0} \\
\texttt{S0} &\equiv \texttt{State} \sqcap \exists \texttt{hasTransition.T01} \sqcap \exists \texttt{hasFluent.F0} \\
\texttt{S1} &\equiv \texttt{State} \sqcap \exists \texttt{hasTransition.T12} \sqcap \exists \texttt{hasFluent.F1} \\
\texttt{S2} &\equiv \texttt{State} \sqcap \exists \texttt{hasTransition.T23} \sqcap \exists \texttt{hasFluent.F2} \\
\texttt{S3} &\equiv \texttt{FinalState} \sqcap \exists \texttt{hasFluent.F3} \\
\texttt{T01} &\equiv \texttt{Transition} \sqcap \exists \texttt{hasCommAct.TimeRequest} \sqcap \exists \texttt{hasNextState.S1} \\
\texttt{T12} &\equiv \texttt{Transition} \sqcap \exists \texttt{hasCommAct.TimeAccept} \sqcap \exists \texttt{hasNextState.S2} \\
\texttt{T23} &\equiv \texttt{Transition} \sqcap \exists \texttt{hasCommAct.TimeInform} \sqcap \exists \texttt{hasNextState.S3} \\
\texttt{TimeRequest} &\equiv \texttt{Request} \sqcap =1\ \texttt{hasContent.TimeReq} \\
\texttt{TimeAccept} &\equiv \texttt{Accept} \sqcap =1\ \texttt{hasContent.TimeReq} \\
\texttt{TimeInform} &\equiv \texttt{Responsive} \sqcap =1\ \texttt{hasContent.TimeInfo} \sqcap =1\ \texttt{inReplyTo.TimeRequest}
\end{aligned}
$$

Simulation of a protocol run

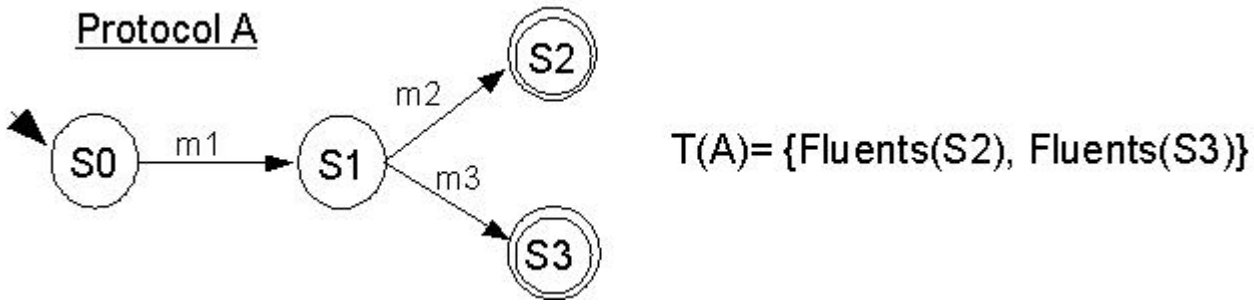


Outline

- Motivation and goal
- Ontology for communication acts
- Protocol descriptions
- Relationships between protocols
- Conclusions

Protocol relationships (1)

- T(A) is the set of traces of A

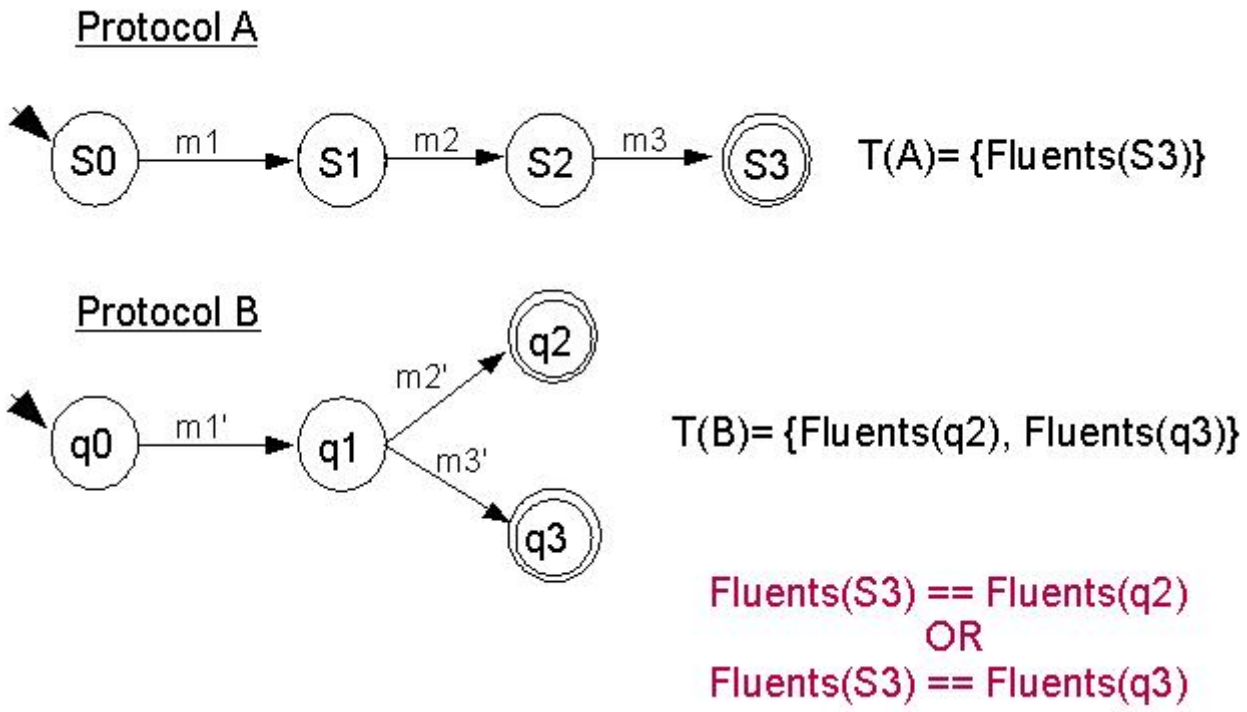


- Protocol A is ***equivalent*** to protocol B if

  "T(A) = T(B)"

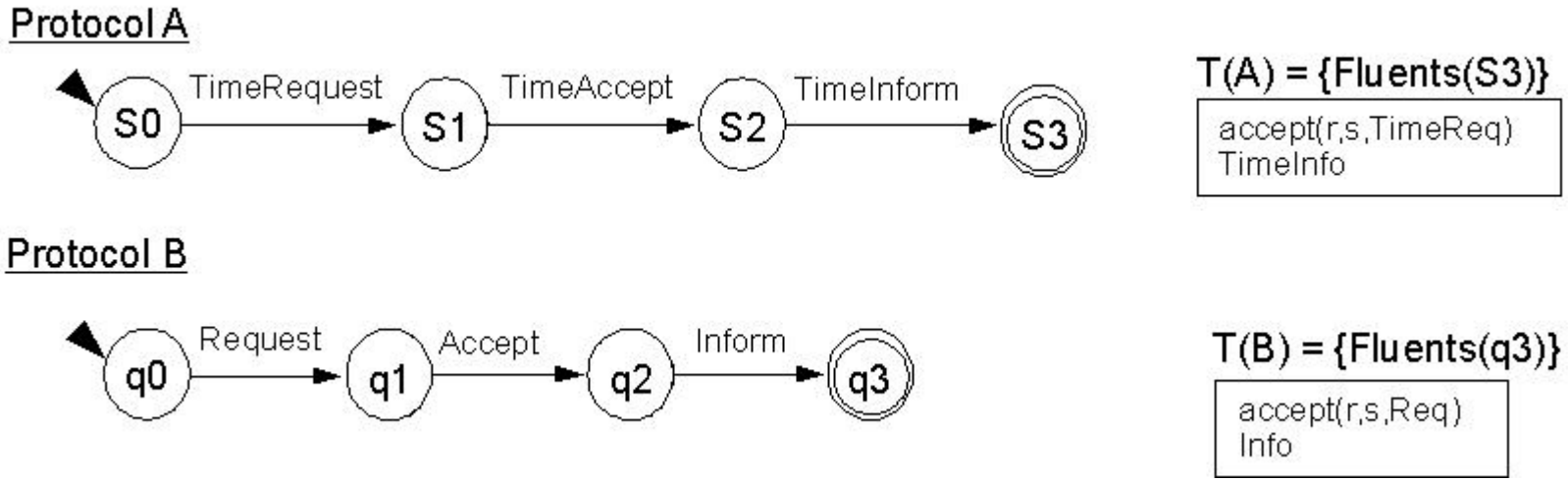
Protocol relationships (2)

- Protocol A is a ***restriction*** of protocol B if "T(A) ⊂ T(B)"



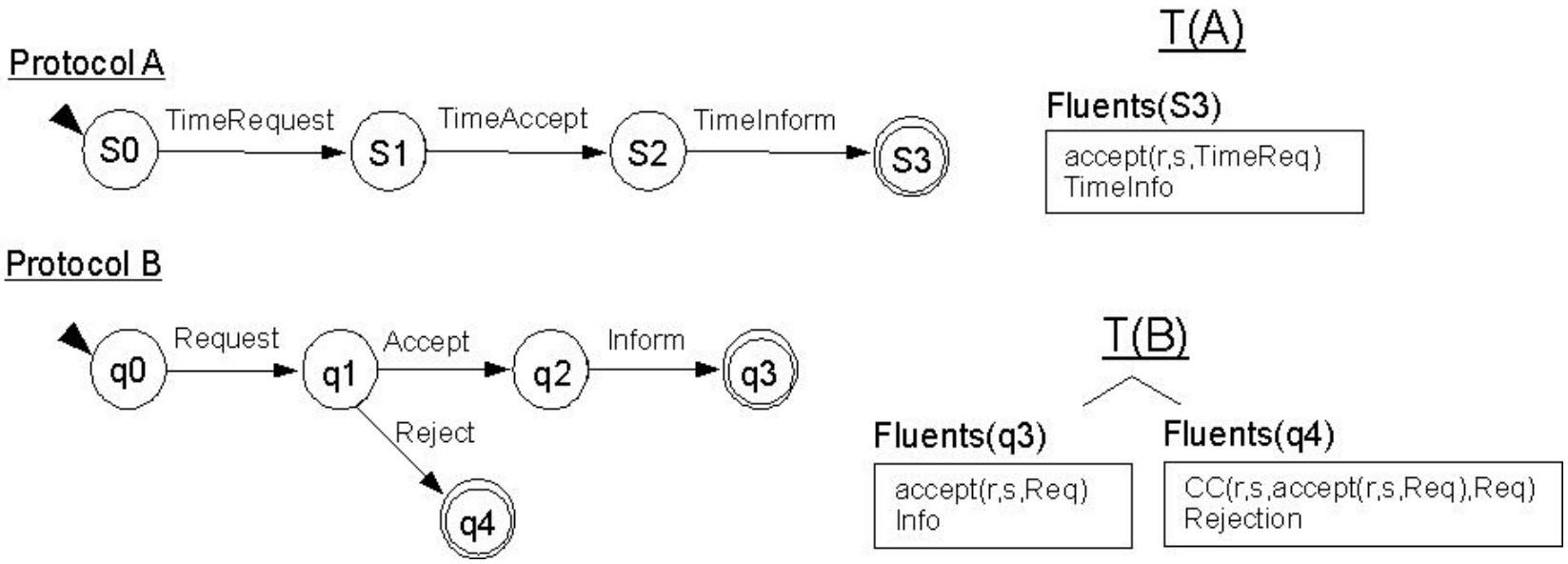
Protocol relationships (3)

- Protocol A is ***specialized-equivalent*** to protocol B if “T(A) =« T(B)”



Protocol relationships (4)

- Protocol A is ***specialized-restriction*** to protocol B if "T(A) ⊂« T(B)"



Outline

- Motivation and goal
- Ontology for communication acts
- Protocol descriptions
- Relationships between protocols
- Conclusions

Conclusions

- Our proposal facilitates:
  - Using CommOnt ontology:
    - Management of semantic aspects when dealing with agent communication protocols
    - Customization of standard communication protocols
  - Support for discovering different kinds of relationships between protocols
  - Use of standard Semantic Web tools

Thanks for your attention!