

An architectural perspective on multiagent societies

Juan Manuel Serrano, Sergio Saugar

Department of Computing

University Rey Juan Carlos, Madrid

**11th International Workshop on
AGENT ORIENTED SOFTWARE ENGINEERING**

Toronto, 10 May 2010

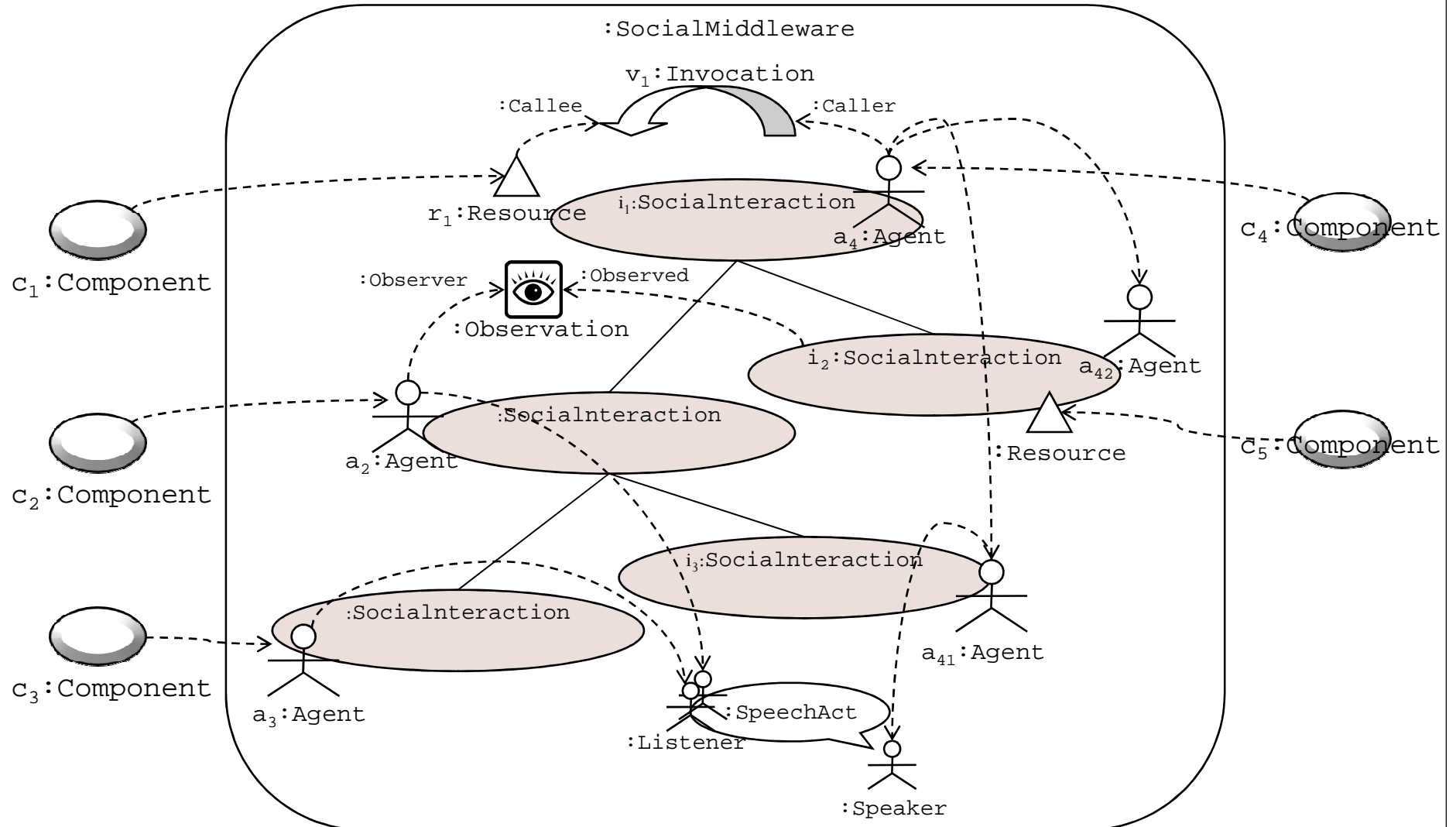
Component & connector perspective

- *Component & connector (C&C) viewtype*: runtime perspective on the structure of software systems
 - Connectors embody runtime interactions between components
- *C&C styles*
 - Closely related to computational models and programming paradigms
 - Publish & subscribe, object-oriented architectures, pipe & filter, shared memory, etc.
 - Characteristic connector types: event channels, invocations, pipes, etc.
- *Middleware as composite connectors*
 - E.g. atomic interaction mechanisms in CORBA-based middleware: synchronous, deferred synchronous, one-way, ...

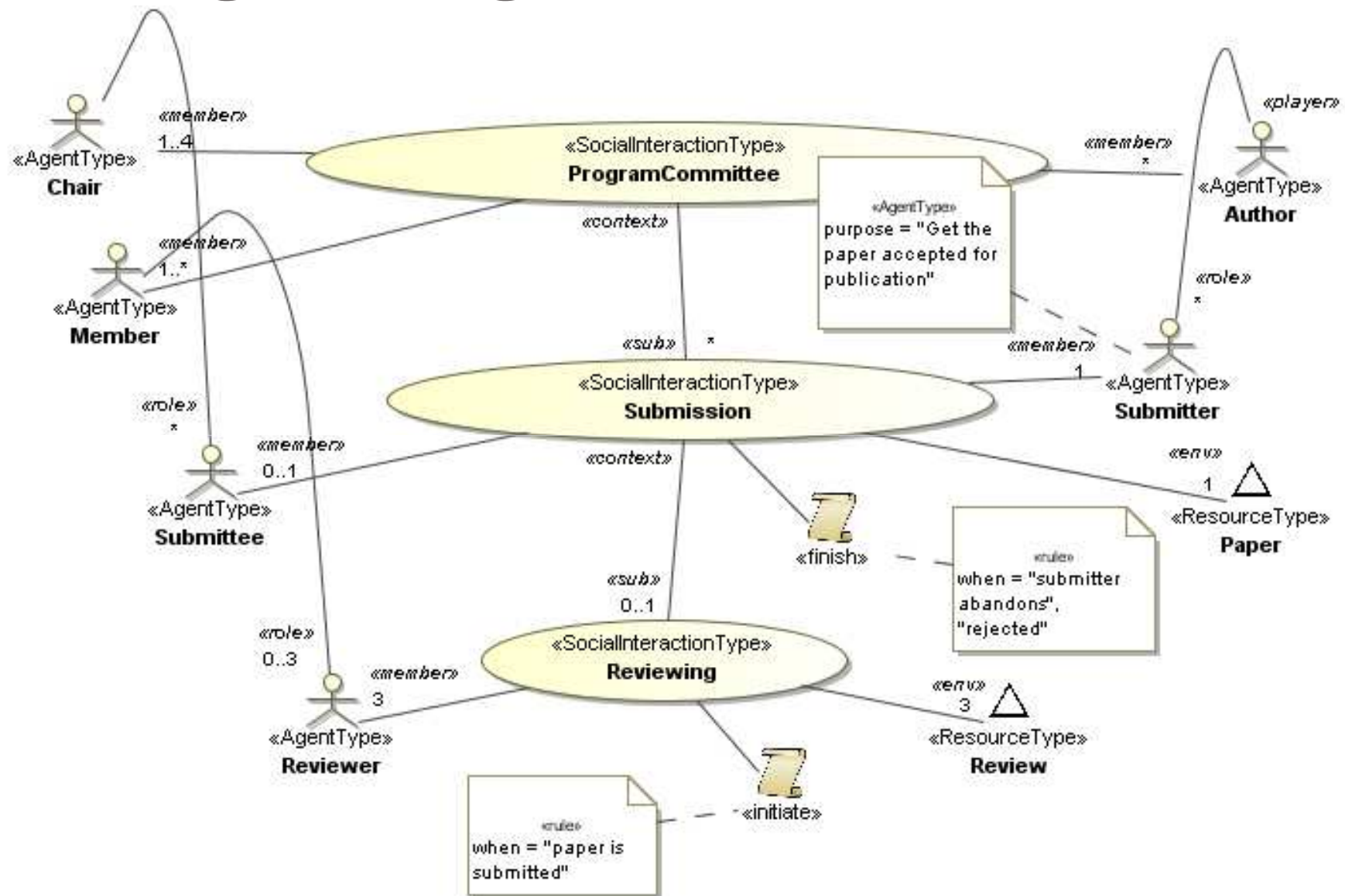
Goals

- How can *multiagent societies* be characterised as an *architectural style* of the C&C viewpoint?
 - Which are the kinds of connectors supported by *social environments / middleware*?
 - How to approach the *design* of the social environment from this perspective?
- Which benefits does the C&C perspective bring to multiagent system research *and viceversa*?

Social middleware as a composite connector



Programming the social environment



Advantages of the C&C perspective

- *Alignment* of multiagent societies with common software engineering practice
 - Speech acts, observations, invocations, social interactions are *high-level* software connectors
 - *Role-based* notion of agenthood
- Connectors as a *design metaphor*
 - *Synchronicity* of speech acts, *pending interactions*, ...
 - *Generic* social interaction mechanism which subsumes the plethora of communicative and organisational abstractions
- New insights into software architecture research
 - Software connectors embed functional requirements too!

Related work

- Weyns on *patterns for situated multiagent systems*
 - Emphasis on *patterns*, formalised through conventional (i.e. low level) component & connectors
- Singh et al. on *commitment-based architectures*
 - Protocols as the only interconnection mechanism, biased towards commitment-based interactions
- Agents & Artifact metamodel
 - *Coordination artifacts* work at the level of general software connectors

Current & future work

- Work developed within the context of *Speech*, a language for *programming* social interactions
 - Runtime semantics [AAMAS'07; CLIMA'08]
 - Type system [LADS'09]
 - Normative incompleteness [COIN'09]
 - Commitments [ECAI'10]
- Two possible extensions
 - Formal model of social connectors, using action languages
 - Multiagent societies as a refinement of *Publish & Subscribe* architectures

Conclusion

- Multiagent societies can be architecturally explained through *high-level* software connectors
- The design of the social environment for a particular application amounts to *social connector programming*
- Connectors provide a fruitful *metaphor* for the design of new features of societal abstractions
- Multiagent societies provide new insights into the nature of software connectors, namely their *programmability*
- SPEECH research on www.speechlang.org