A Graph-based Operational Semantics for a Machine Model with Actor-based Concurrency

Tim Molderez, Hans Schippers & Dirk Janssens

Universiteit Antwerpen

Dissection of the title

- A Graph-based Operational Semantics for a Machine Model with Actor-based Concurrency

- **Machine model**: Describes how a language's compilation target works
  (e.g. the semantics of Java's bytecode)
- **Graph-based operational semantics**: The model is described in graph rewrite rules, which can be executed (using a tool such as AGG)
- **Actor-based concurrency**: The model supports concurrency using the actor model

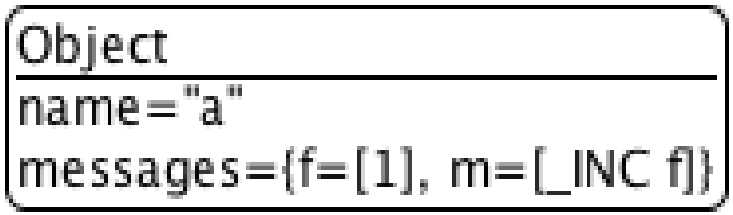
The delMDSOC machine model

- Our machine model **delMDSOC**:
  Delegation-based Multi-dimensional Separation of Concerns

- Languages with MDSOC: Aim to break the "tyranny of dominant decomposition"
  - Aspect-oriented languages
  - Context-oriented languages
  - Subject-oriented languages
  - Feature-oriented languages
  - Role-based languages
  - ...

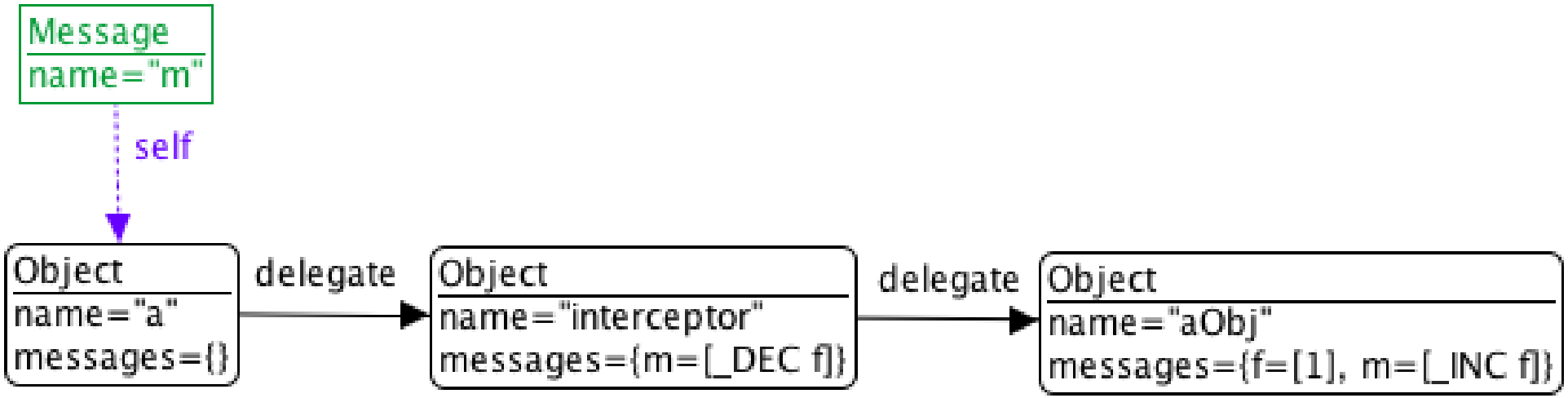
Prototypes and messages

- A prototype-based OO environment with message passing
  - There only are objects that communicate by sending messages to each other

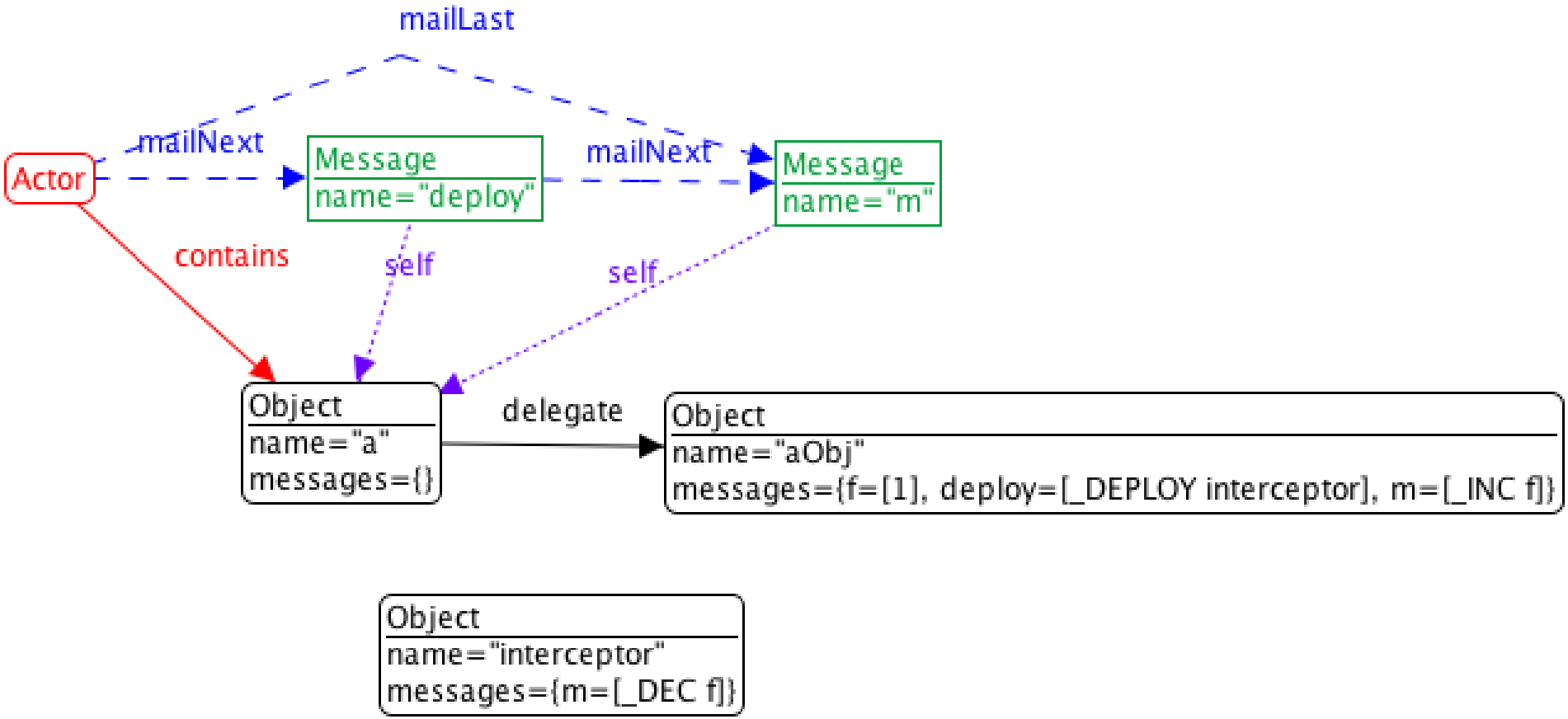


Message delegation

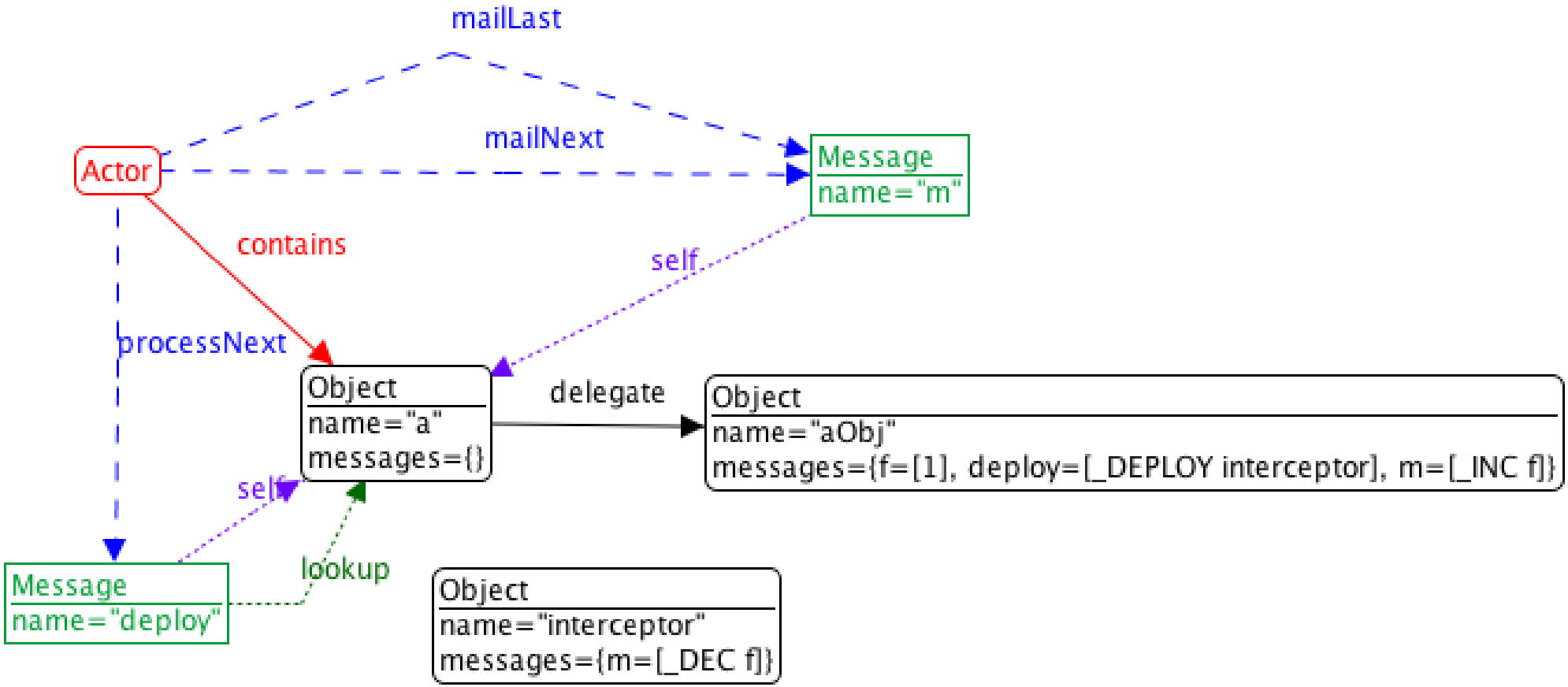
- Delegation-based:
  - An object can be linked with another object
  - An object can delegate a message to the one it's linked with, if the message is not understood
  - Allows for modularization of crosscutting concerns



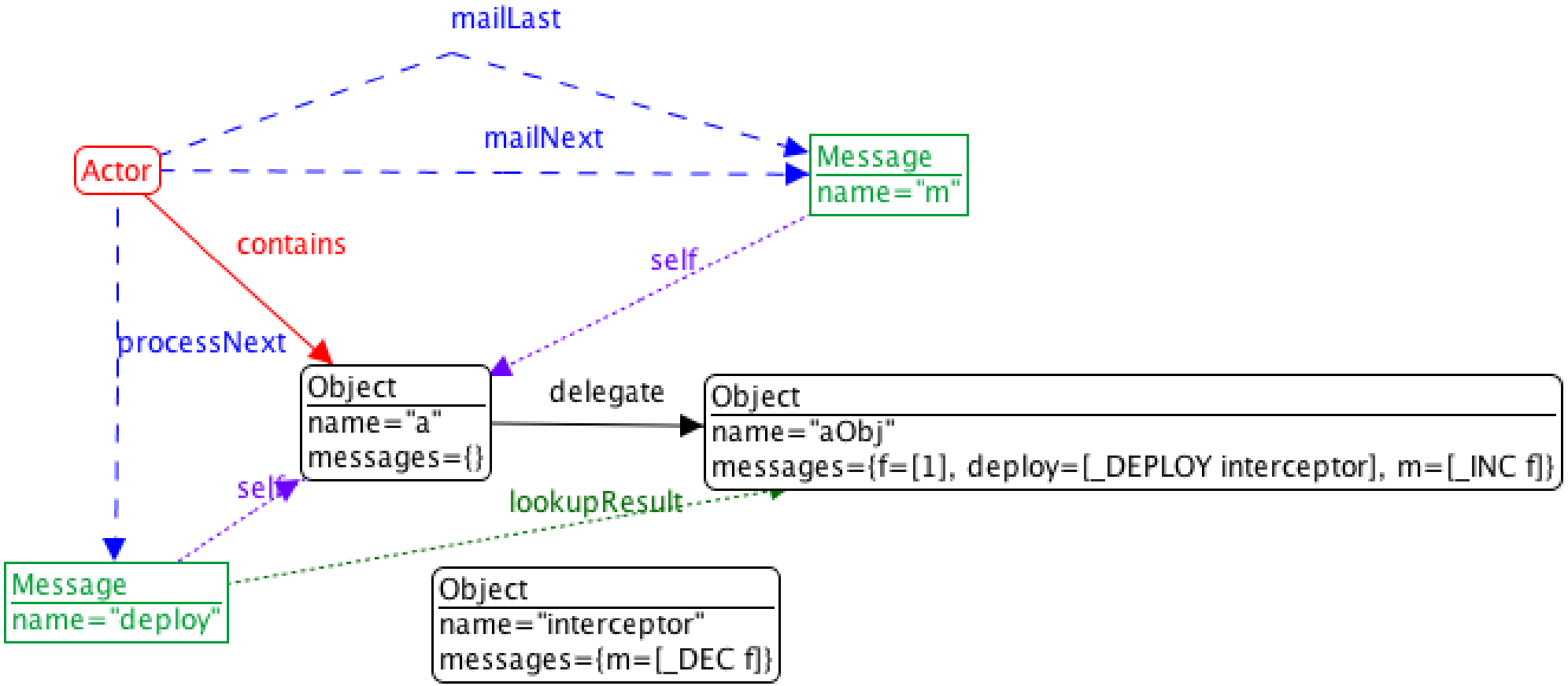
Actor-based concurrency



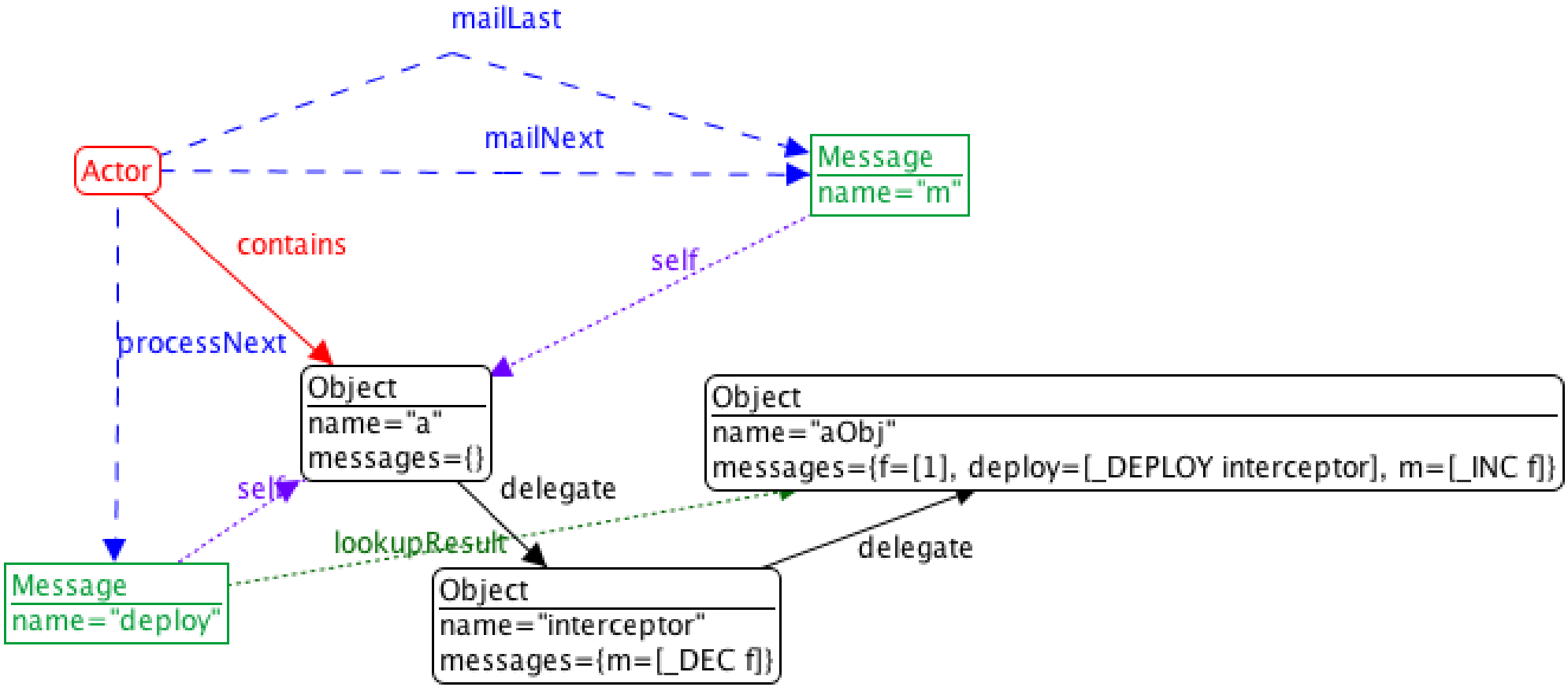
Actor-based concurrency



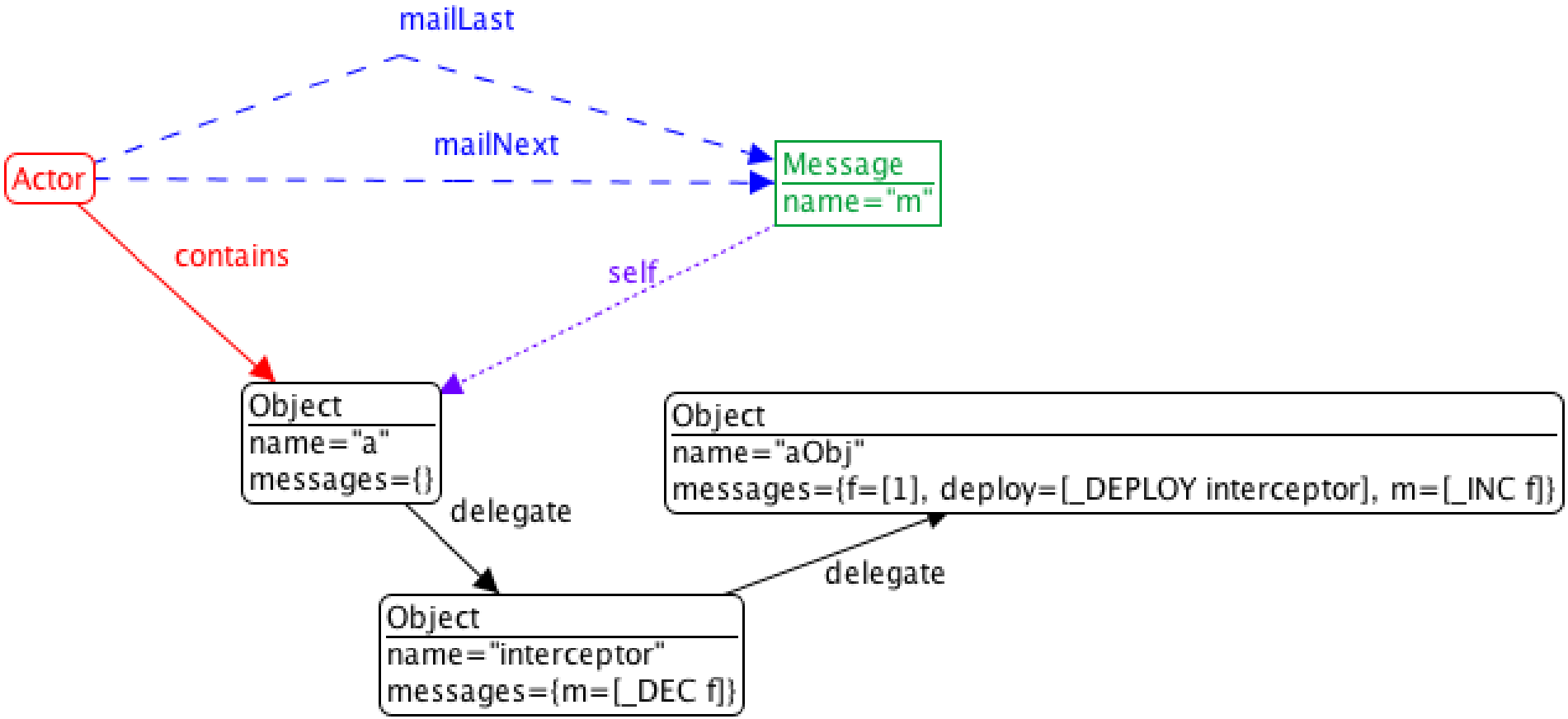
Actor-based concurrency



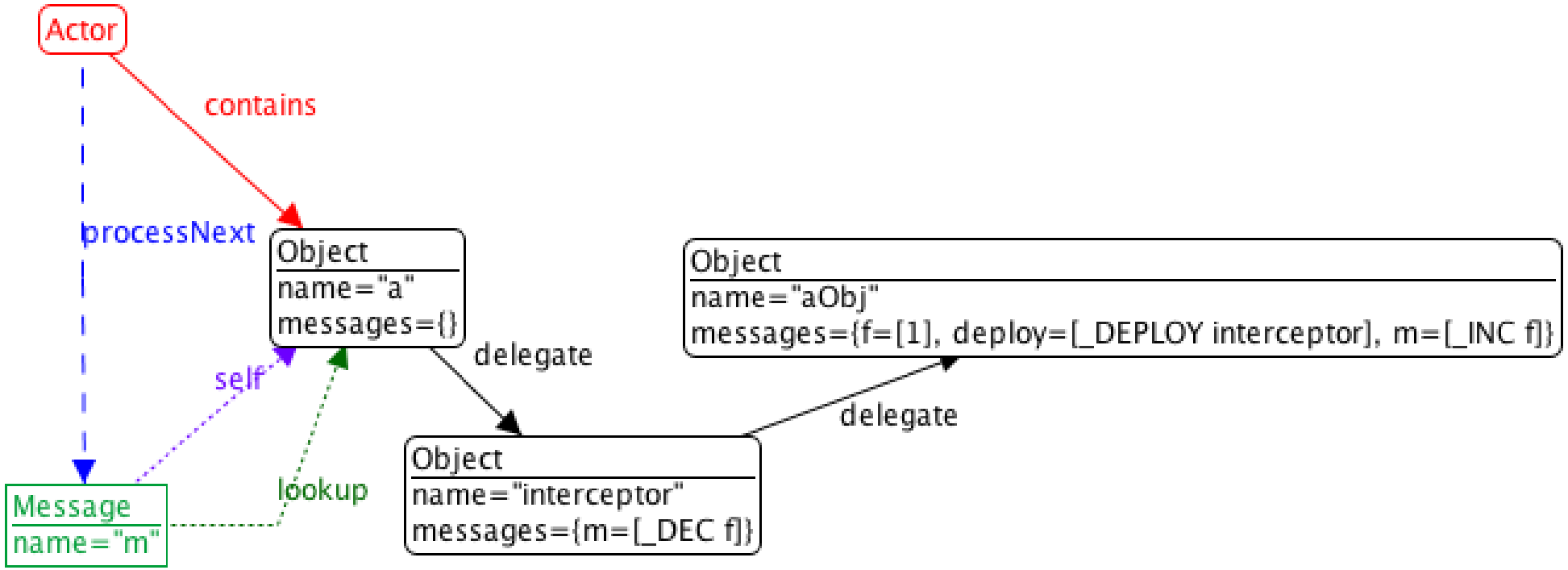
Actor-based concurrency



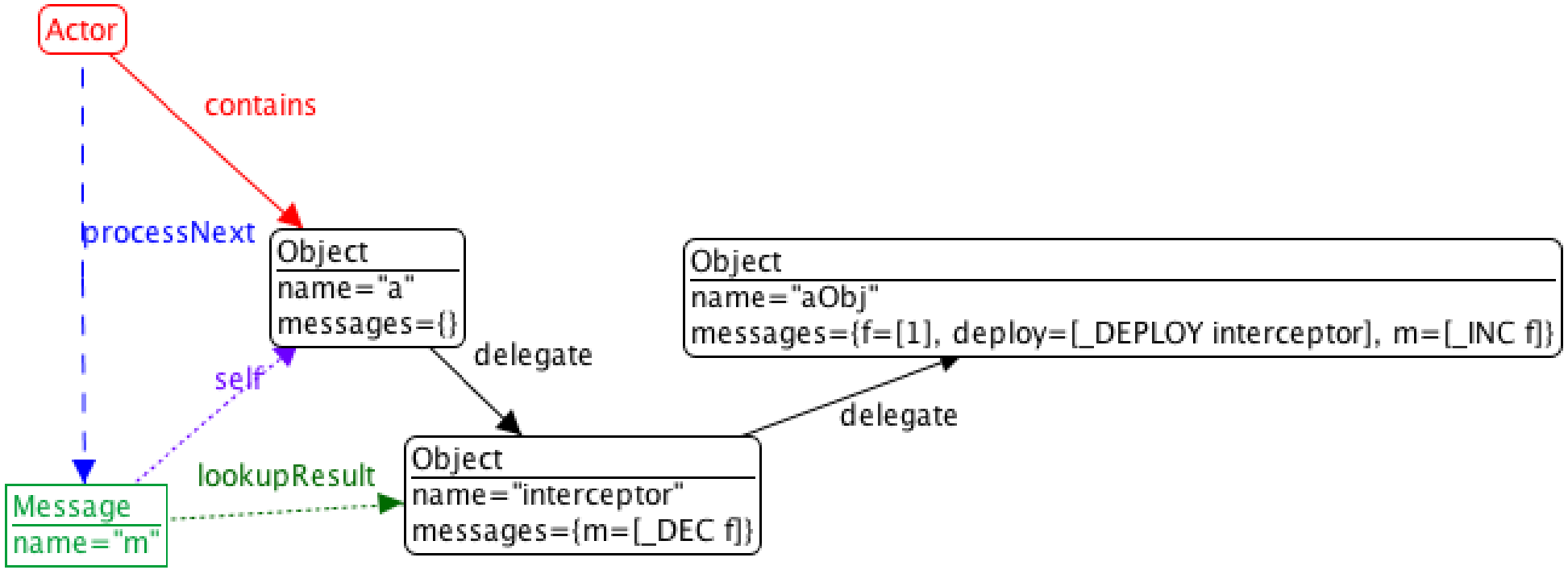
Actor-based concurrency



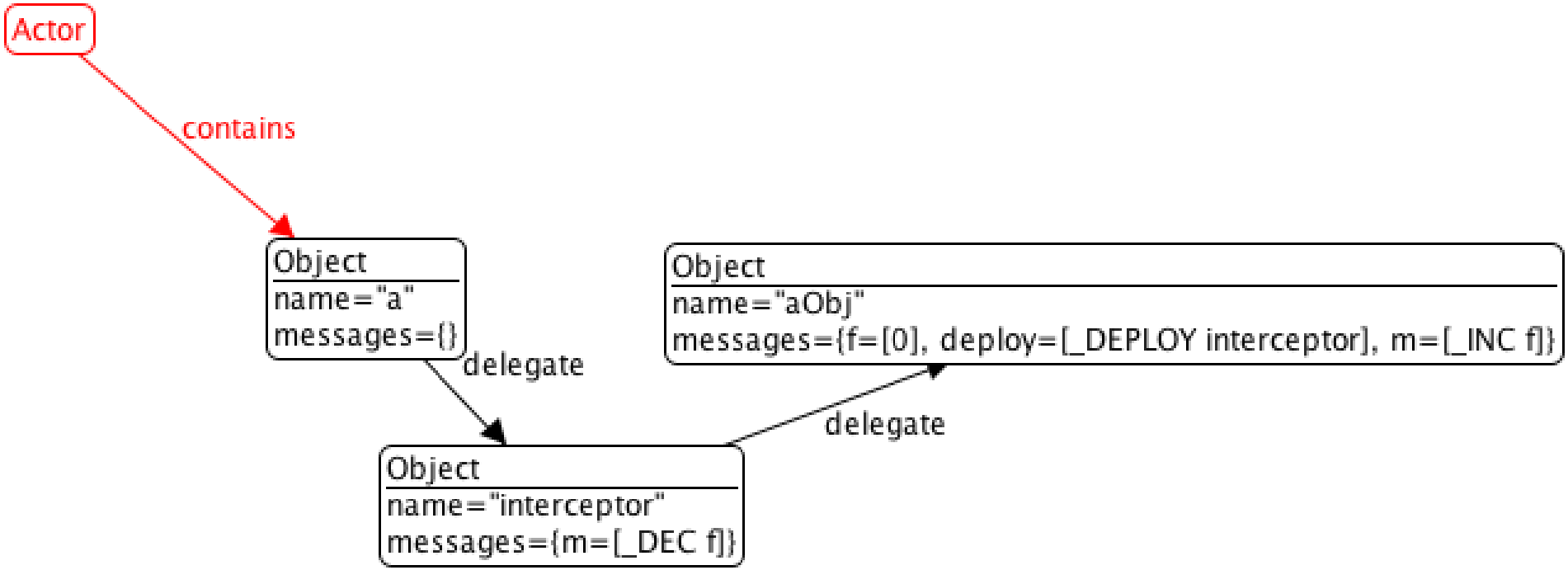
Actor-based concurrency



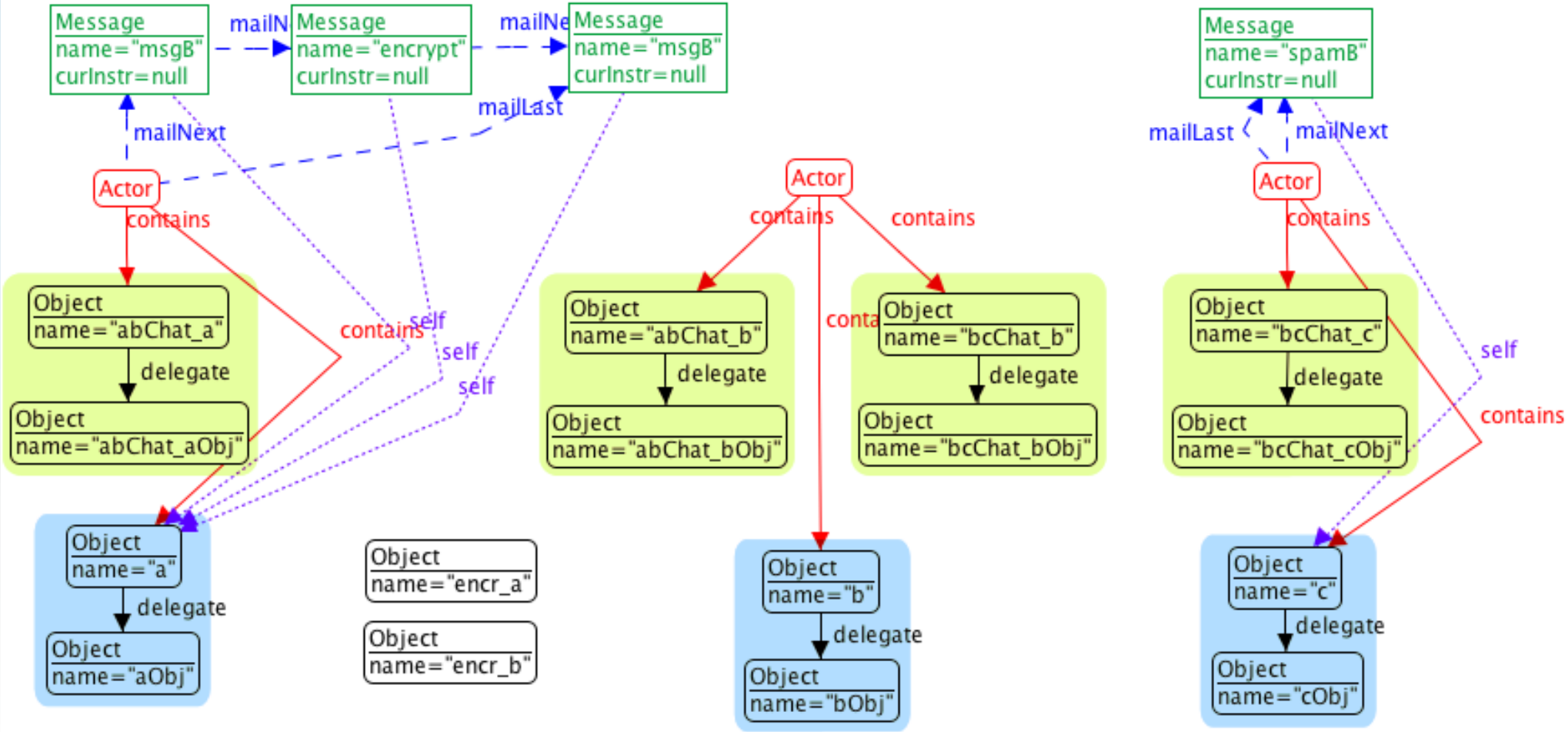
Actor-based concurrency



Actor-based concurrency



Chat application example



Conclusion and Future Work

- delMDSOC: A machine model for MDSOC languages
  - Graph-based operational semantics with the AGG tool
  - Actor-based concurrency
- Short/mid-term future work:
  - Delegation as a function
  - Synchronous communication between actors or support for futures
  - Migrating objects between actors
  - Replace global namespace with something better
  - Garbage collection

Conclusion and Future Work

- Long-term future directions
  - Use the model for verification purposes
  - Implementation of delMDSOC VM using Maxine, Erlang, …
  - Implement several high-level MDSOC languages on top of the model
    - Compare/combine/evolve different MDSOC paradigms
    - Transform applications from one language to another

Questions?

Synopsis

- delMDSOC machine model
- Graph-based operational semantics
- Actor-based concurrency