

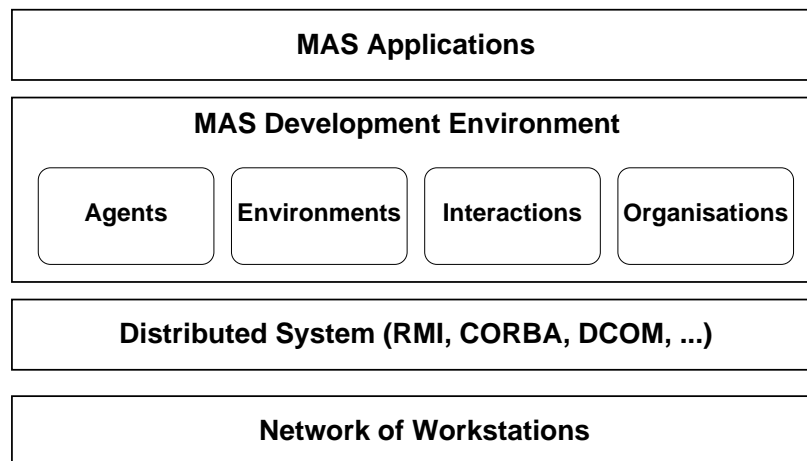
SACI: Uma Ferramenta para Implementação e Monitoração da Comunicação entre Agentes

Jomi Fred Hübner e Jaime Simão Sichman
Universidade de São Paulo
Laboratório de Técnicas Inteligentes, EPUSP

<http://www.lti.pcs.usp.br/saci>

Contexto

Desenvolvimento de Sistemas Multi-Agentes



[[Demazeau, 1995](#)]

O que uma ferramenta para interação deveria ter:

- Envio e recebimento de mensagens utilizando uma linguagem de comunicação
- Uso de protocolos de comunicação especificados pelo usuário
- Acompanhamento do funcionamento do sistema
- Facilidades: páginas amarelas, execução remota, interface, ...

Motivação

Uma ferramenta com as seguintes características

- interoperabilidade (linguagens, SO, ...)
- desempenho
- robustez
- fácil utilização para o programador e para o usuário (facilitando o ensino de SMA)
- distribuição gratuita

Objetivo

- Apresentar uma visão geral do Saci:
 - propósitos
 - conceitos fundamentais
 - arquitetura de funcionamento
- Compará-lo com outras ferramentas

Saci

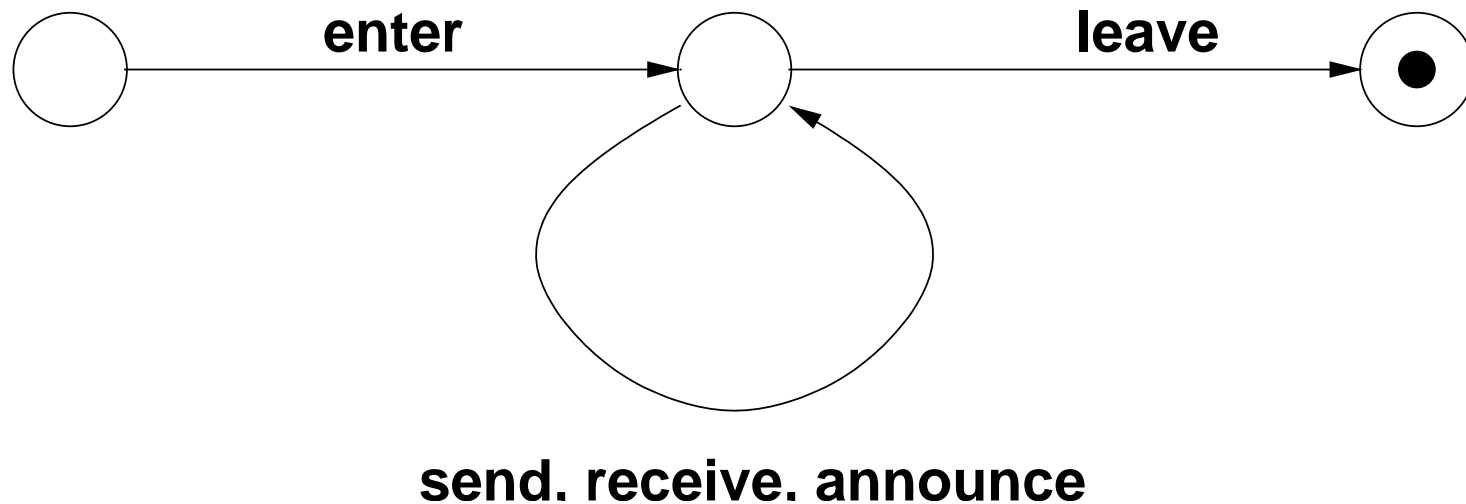
O Saci é uma ferramenta que torna a programação da **comunicação** entre agentes distribuídos mais fácil, em conformidade com um padrão, rápida e robusta. Compõem-se de

- Uma API para compor (parser), enviar (as/sincronamente) e receber (mail box) mensagens KQML
- Um conjunto de facilidades:
 - páginas brancas
 - páginas amarelas
 - execução remota de agentes
 - monitoramento da sociedade (eventos sociais podem ser obtidos para visualização ou para armazenamento e análise futura)
 - agentes podem ser applets e funcionar em um browser (nenhuma configuração é necessária no cliente)

Agentes

Para o Saci, os **agentes** possuem as seguintes propriedades:

- estão agrupados em sociedades
- possuem uma identificação única
- interagem com os demais agentes utilizando uma linguagem comum
- oferecem serviços aos demais agentes da sua sociedade
- tem o seguinte ciclo de vida



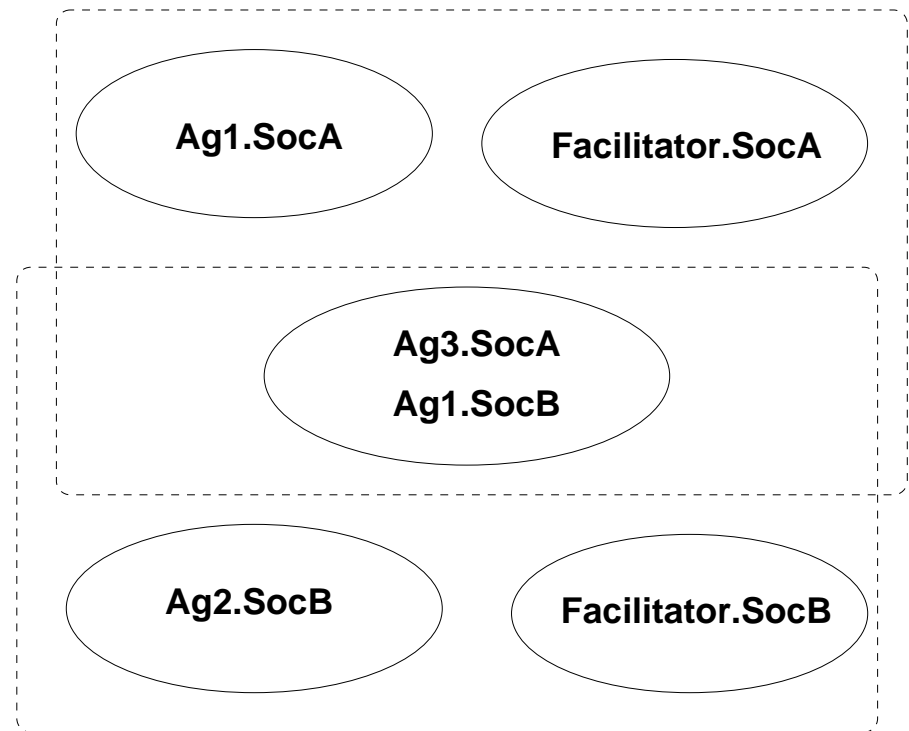
Arquitetura (Como o Saci funciona)

- Entrada e saída de Agentes
- Envio e recebimento de mensagens
- Anúncio de habilidades
- Protocolo de monitoramento

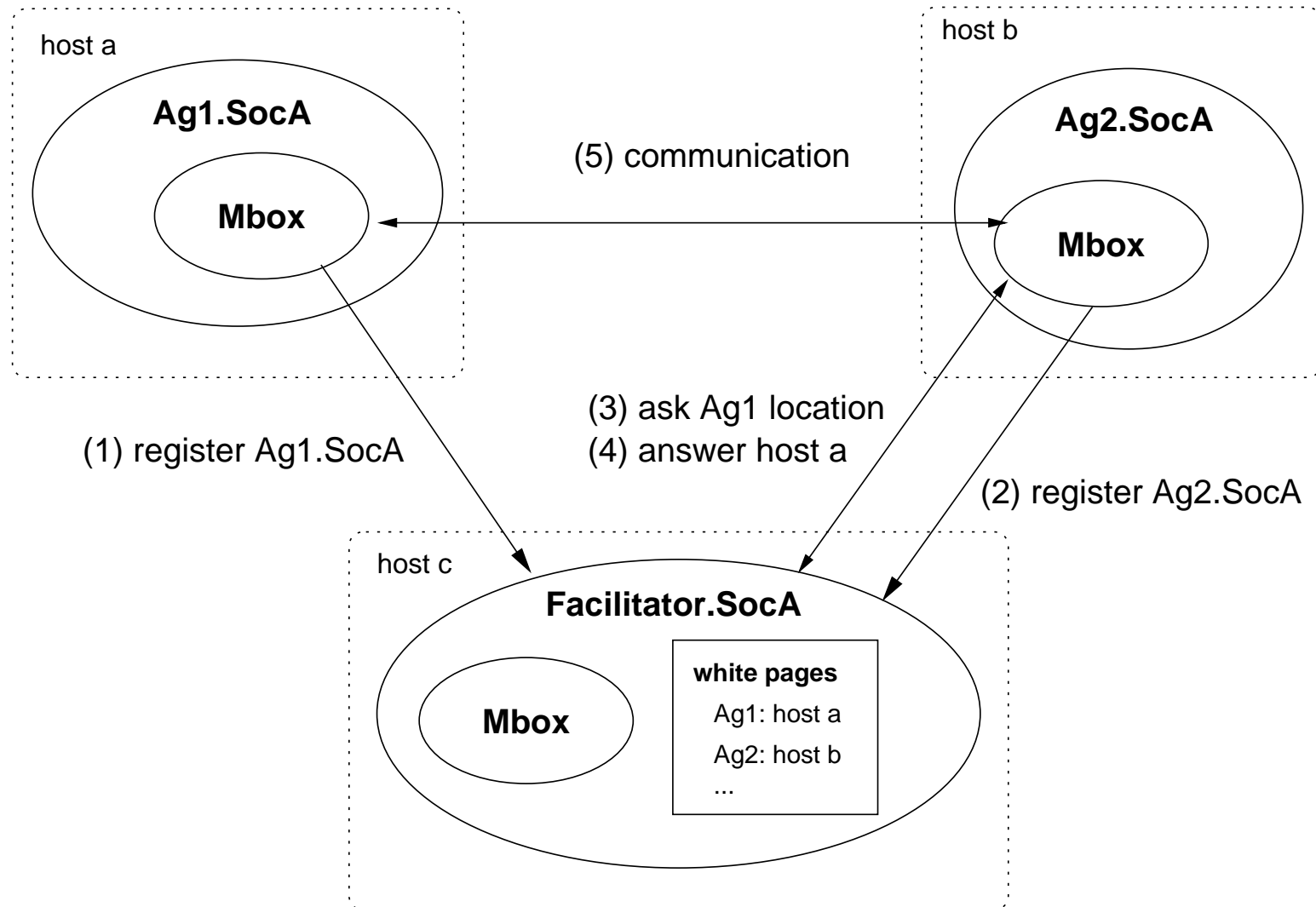
Entrada e saída da sociedade

Cada sociedade possui um **agente facilitador** que mantém sua estrutura no decorrer da sua história (sequência de eventos sociais).

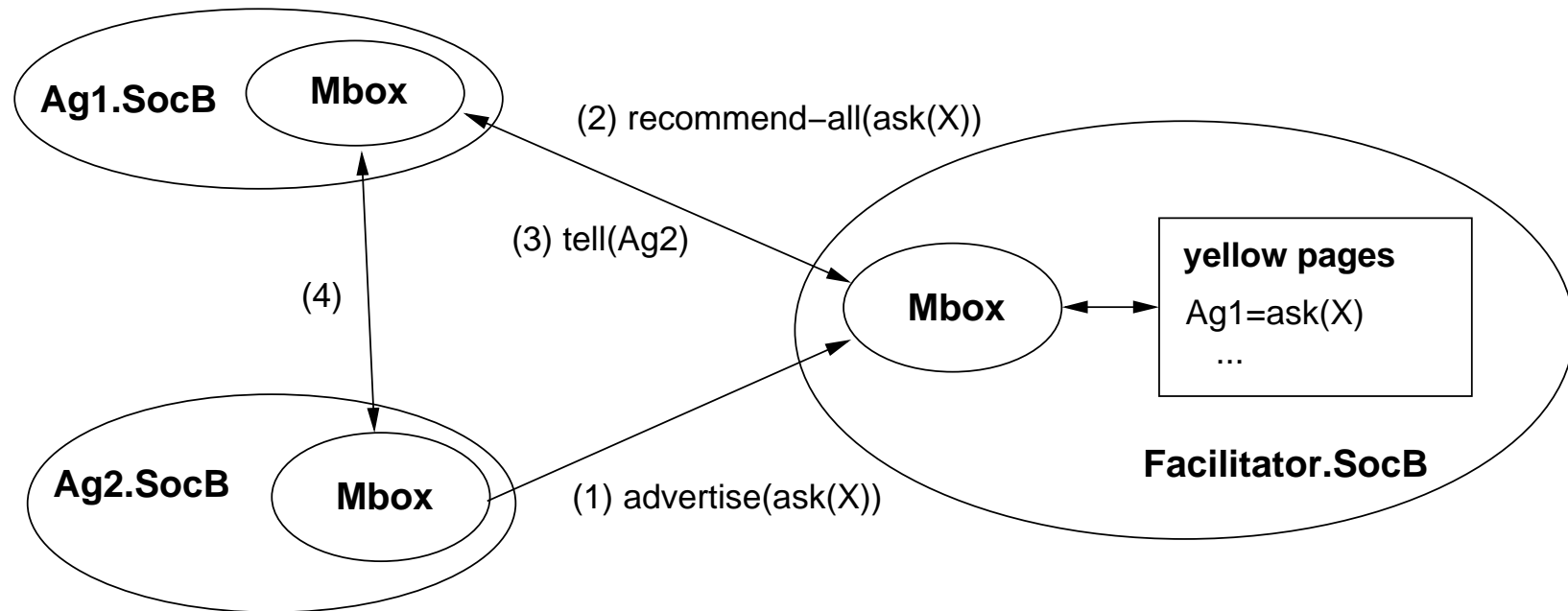
- Para entrar em uma sociedade um agente tem que registrar seu nome no facilitador e, antes de sair, deve avisar o facilitador.
- Os serviços que um agente deseja disponibilizar à sociedade devem ser anunciados ao facilitador.



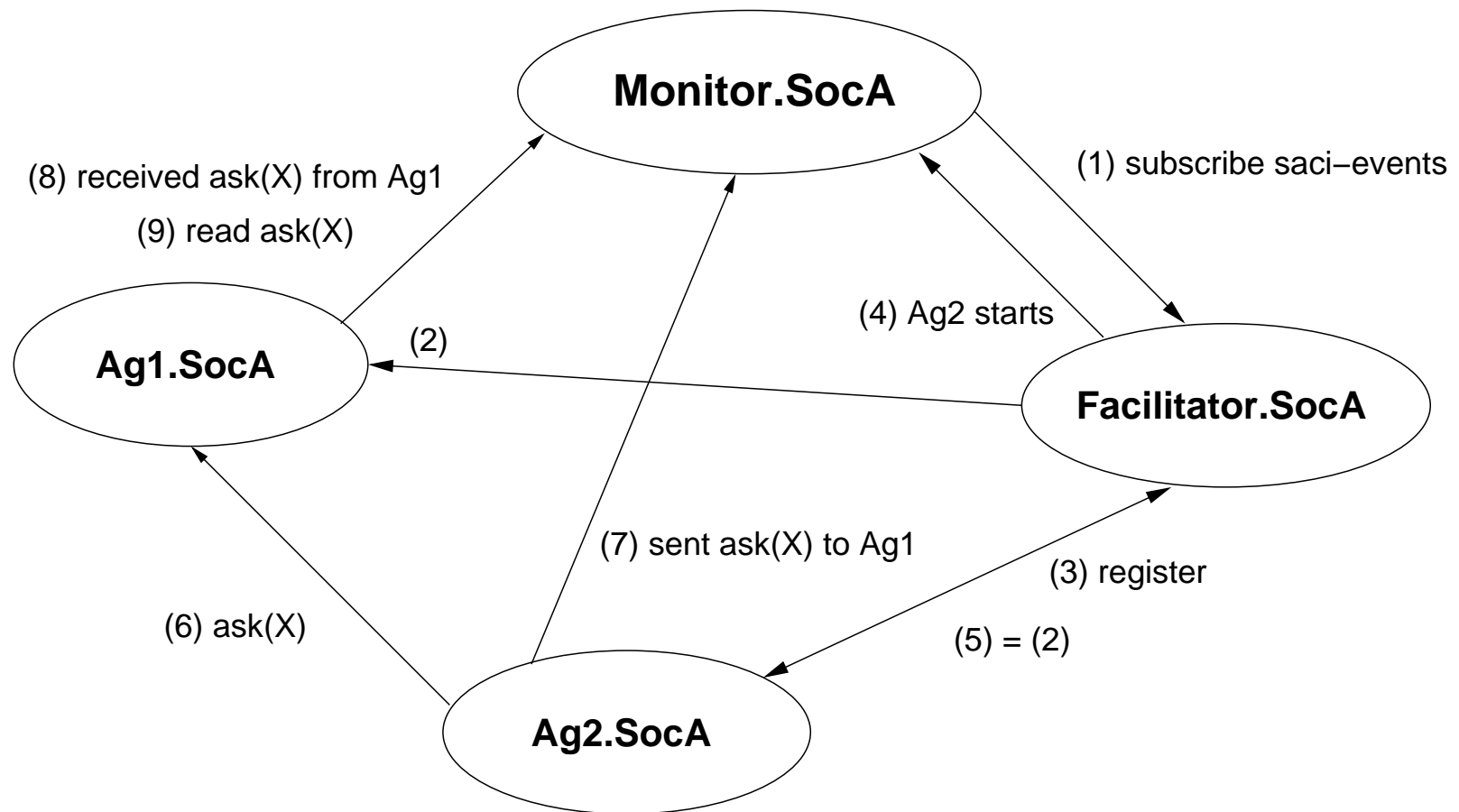
Envio e recebimento de mensagens



Anúncio de habilidades

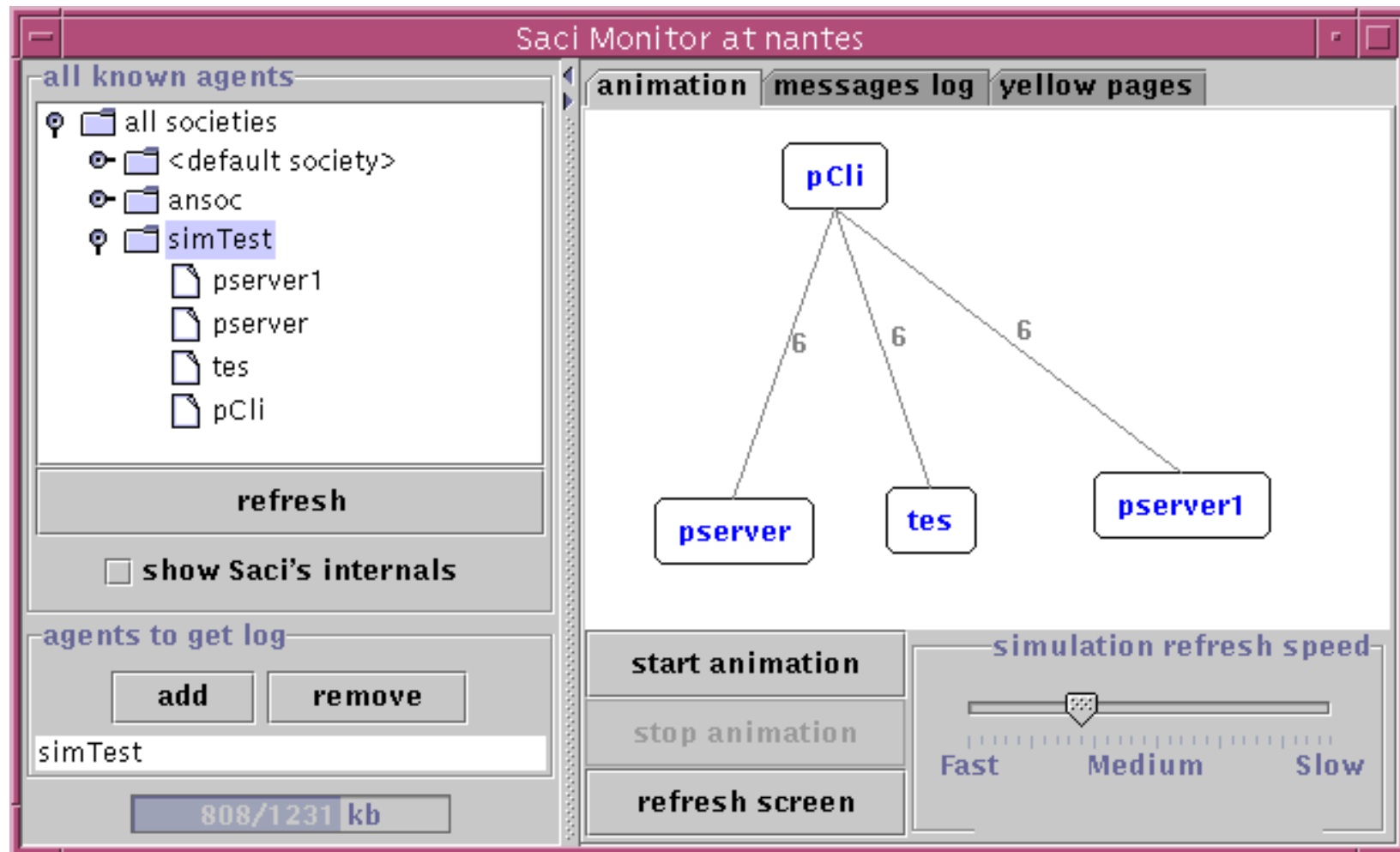


Monitoramento



(2) = ag Monitor subscribed saci-events

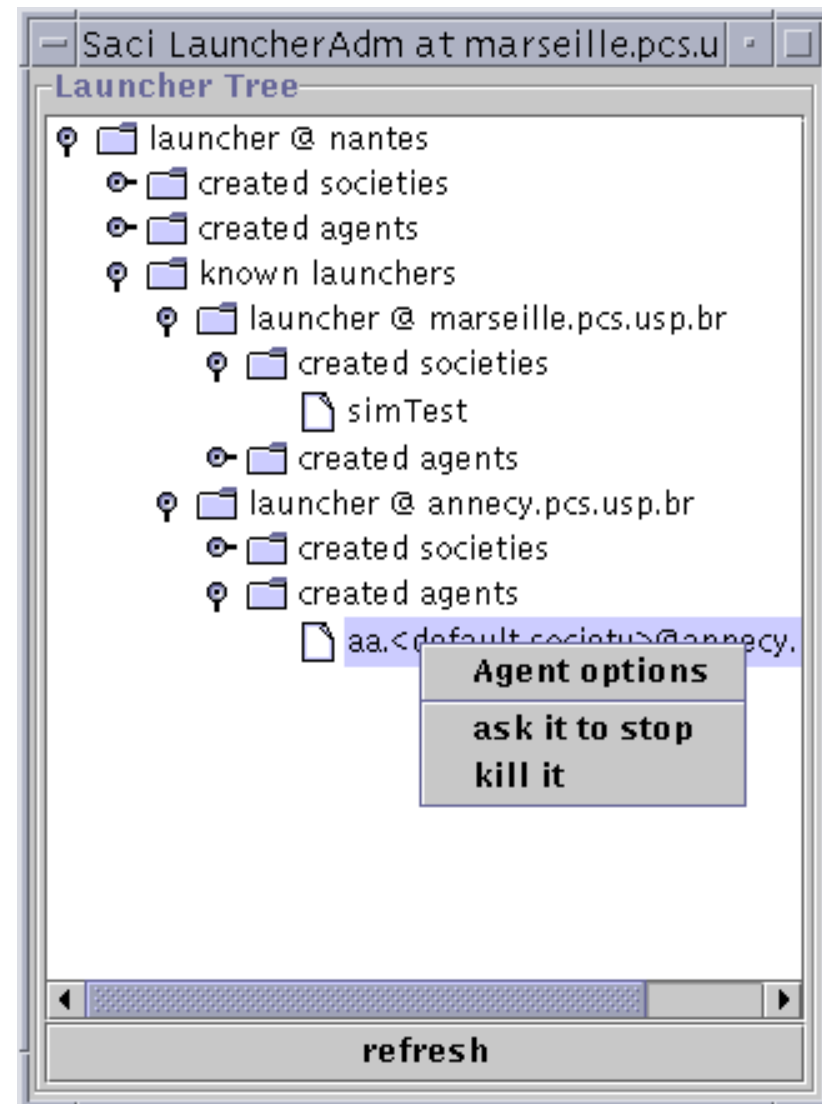
Exemplo de aplicação do monitoramento



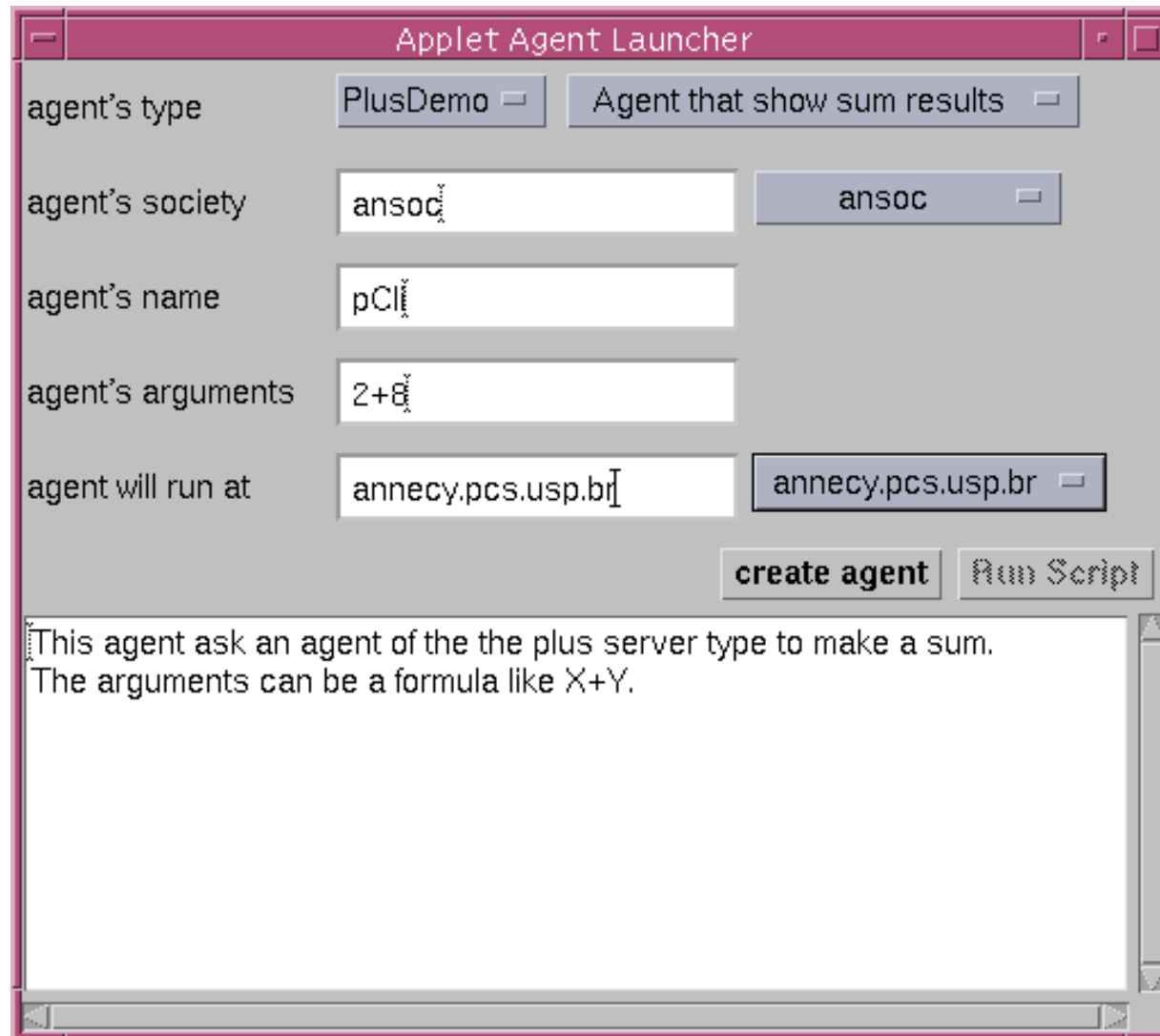
Ferramentas

Launcher Demon

- Facilitar a criação (remota) de agentes e facilitadores
- Execução de scripts
- Serviço de MailBox para Applets



AgentLauncher



The screenshot shows a window titled "Applet Agent Launcher" with the following fields and controls:

- agent's type:** A dropdown menu showing "PlusDemo" and a description "Agent that show sum results".
- agent's society:** A text input field containing "ansoc" and a dropdown menu also showing "ansoc".
- agent's name:** A text input field containing "pCl".
- agent's arguments:** A text input field containing "2+8".
- agent will run at:** A text input field containing "annecy.pcs.usp.br" and a dropdown menu also showing "annecy.pcs.usp.br".

At the bottom right, there are two buttons: "create agent" and "Run Script".

Below the input fields is a text area containing the following text:

This agent ask an agent of the the plus server type to make a sum.
The arguments can be a formula like X+Y.

Comparação com outras ferramentas

As seguintes ferramentas foram selecionadas para a comparação:

- Jackal (UMBC) [[Cost et al., 1998](#)]
- JKQML (IBM) [[Tsuchitami,](#)]
- FIPA-OS (Nortel Networks) [[Networks, 2000](#)]

Características

- c1. Os agentes poderem ser executados como applets
- c2. Mecanismo de monitoramento
- c3. Simplicidade de utilização. (a) o número de palavras do código fonte e (b) o número de parâmetros de configuração
- c4. Definição de protocolos de interação

Ferramenta	Critérios				
	c1	c2	c3(a)	c3(b)	c4
Saci	sim	sim	421	0	não
Jackal	não	não	382	6	sim
JKQML	não	não	1176	1	sim
FIPA-OS	não	sim	950	1	não

Desempenho

Foi medido o número médio de mensagens trocadas por segundo entre dois agentes:

- servidor: anuncia seu serviço, espera requisições e as responde.
- cliente: encontra um servidor através das páginas amarelas, e mede o tempo necessário para realizar n requisições.

Os testes foram realizados em três circunstâncias distintas:

- t1. Os dois agentes executando na mesma JVM como threads.
- t2. Os dois agentes executando em JVMs diferentes mas no mesmo computador.
- t3. Os dois agentes executando em computadores diferentes.

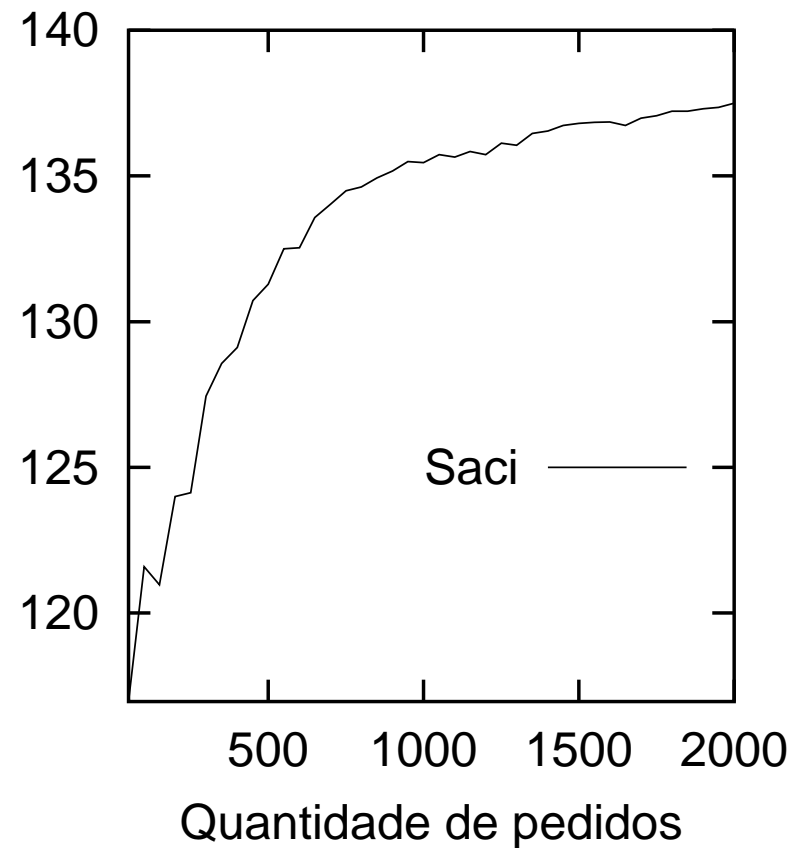
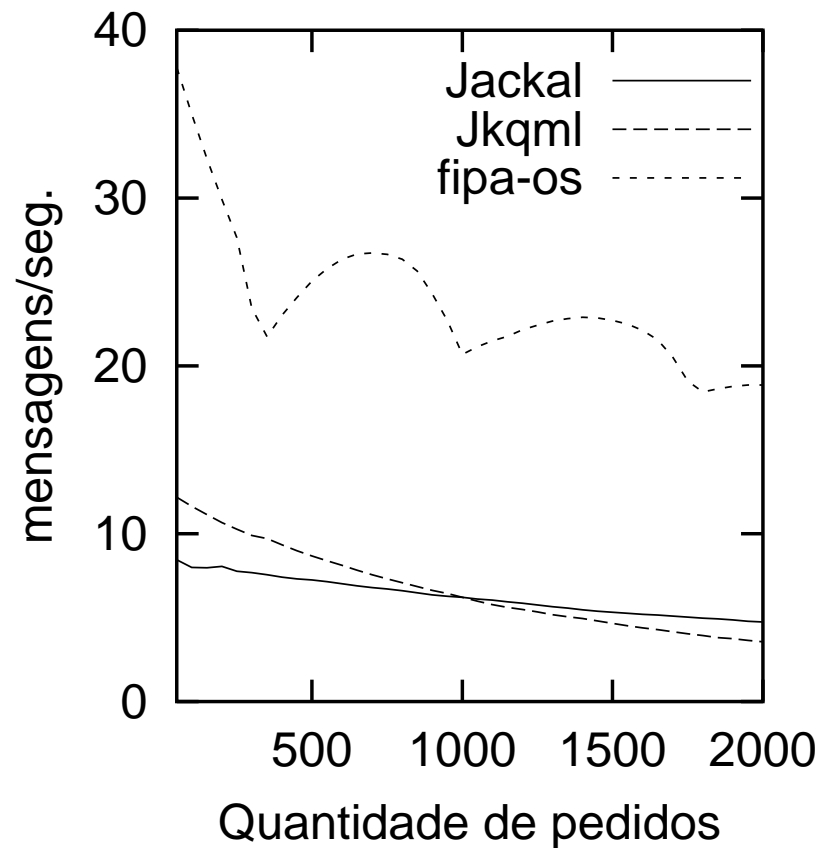
Resultado

Ferramentas	Testes					
	t1		t2		t3	
Saci	2.412,54		197,24		137,49	
Jackal	6,03	400x-	6,64	29x-	4,73	29x-
JKQML	1,46	1652x-	2,55	77x-	3,56	38x-
FIPA-OS	17,95	134x-	25,13	7x-	18,86	7x-

Observações:

- $nx-$: n vezes mais lento que o Saci
- No saci, a comunicação entre agentes na mesma JVM é 12 vezes mais rápida

Desempenho no tempo



Conclusões

O ambiente Saci mostrou-se uma boa solução para a implementação da comunicação em um SMA.

- portátil
- bom desempenho
- facilidade de uso (transparência da tecnologia de programação distribuída)
- robusto (o desempenho não diminui no decorrer do tempo)

Trabalhos futuros

- Protocolos de interação definidos pelo usuário
 - Linguagem de descrição de protocolos (transição de estados)
 - Como o agente irá usar o protocolo
- Mobilidade

Mais informações

Na página do Saci (<http://www.lti.pcs.usp.br/saci>) encontra-se:

- Manual de programação de agentes utilizando o Saci
 - Conceitos utilizados na especificação do Saci
 - Funcionamento do Saci
 - Programação de agentes Saci
 - Uso das facilidades do Saci
- Documentação das classes
- Programas exemplo

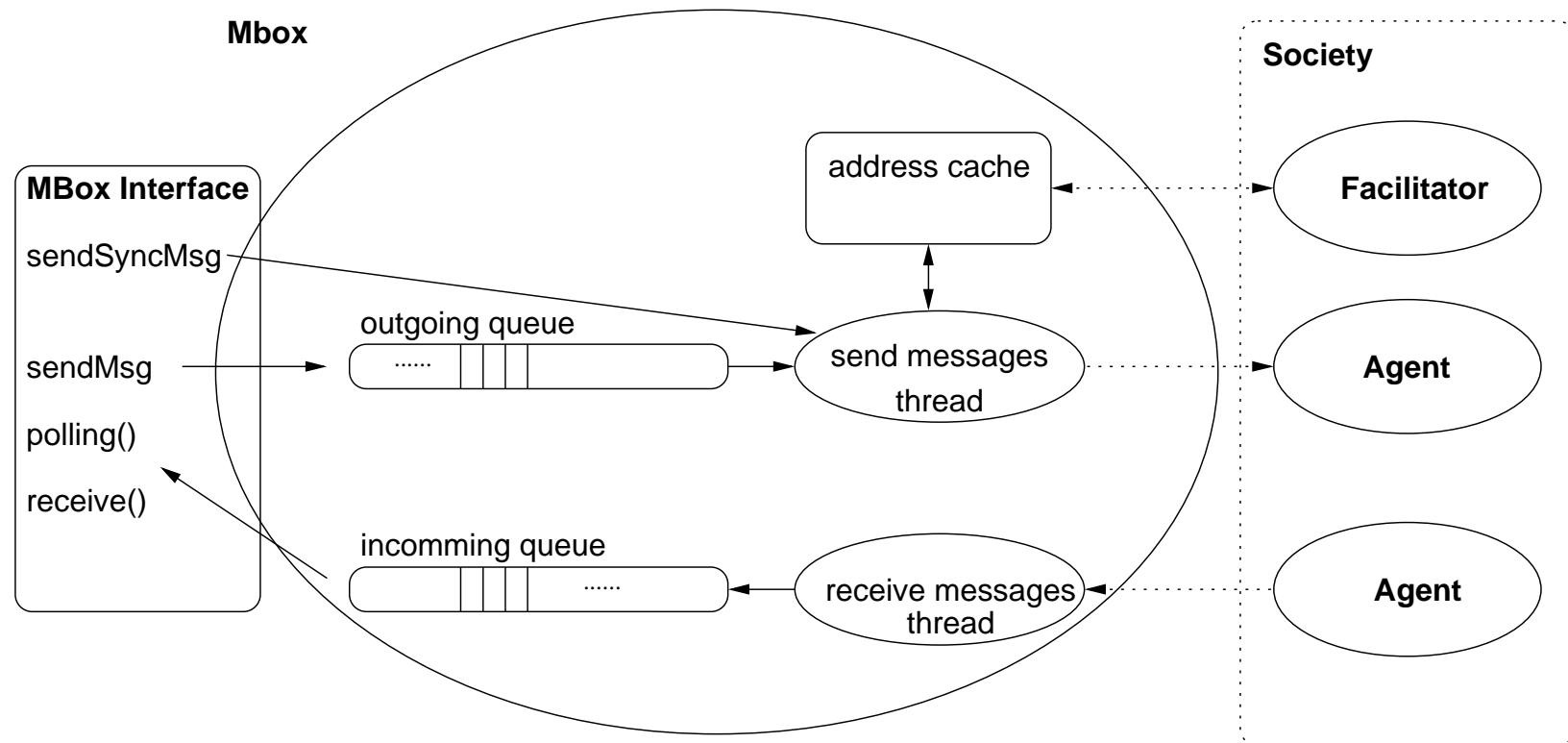
Referências

- [Cost et al., 1998] Cost, R. S., Finin, T., Labrou, Y., Luan, X., Peng, Y., Soboroff, I., Mayfield, J., and Boughannam, A. (1998). Jackal: A java-based tool for agent development. In *Working Notes of the Workshop on Tools for Developing Agents (AAAI Technical Report)*, AAAI 98. (<http://jackal.cs.umbc.edu/Jackal/>).
- [Demazeau, 1995] Demazeau, Y. (1995). From interactions to collective behaviour in agent-based systems. In *Proceedings of the European Conference on Cognitive Science*, Saint-Malo (France).
- [Networks, 2000] Networks, N. (2000). FIPA-OS.
(<http://www.nortelnetworks.com/products/announcements/fipa/>).
- [Tsuchitami,] Tsuchitami, H. Java KQML.
(<http://www.alphaworks.ibm.com/formula/JKQML>).

Slides para eventuais perguntas

- Funcionamento do MBox
- Como fica a programação
- Exemplo de funcionamento do LauncherD
- Razões para o alto desempenho do Saci

Arquitetura do MBox



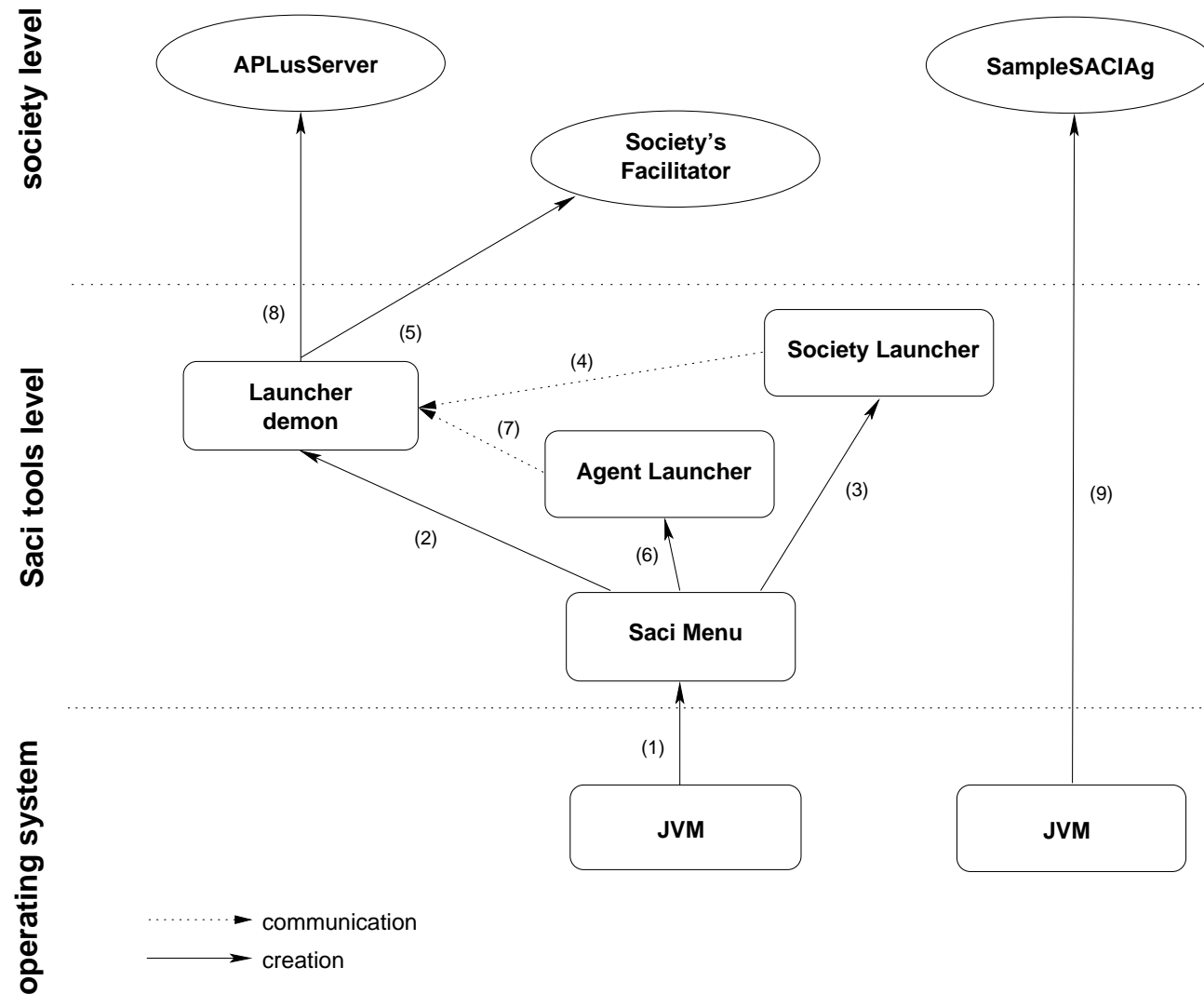
Exemplo de programação de um agente

```
import saci.*;  
public class SampleSACIAg extends Agent {  
    public static void main(String[] args) {  
        SampleSACIAg a = new SampleSACIAg();  
        if (a.enterSoc("SampleSACIAg")) {  
            a.run();  
            a.leaveSoc();  
        }  
    }  
    public void run() {  
        try {  
            Message m = new Message("(ask-one :content \"2+4\":receiver APlusServer)");  
            Message r = mbox.ask(m);  
            System.out.println("Answer is "+ r.get("content"));  
        } catch (Exception e) { .... }  
    }  
}
```

Um agente applet

```
import saci.*;  
  
public class SampleSACIApplet extends AppletAgent {  
    public void init() {  
        if (enterSoc("SampleAppletAgent")) {  
            run();  
            leaveSoc();  
        }  
    }  
    public void run() {  
        try {  
            Message m = new Message("(ask-one :content \"2+4\":receiver APlusServer)");  
            Message r = mbox.ask(m);  
            add (new java.awt.Label( "Answer is "+ r.get("content")));  
        } catch (Exception e) { .... }  
    }  
}
```

Exemplo de uso das ferramentas



Razões para o desempenho do Saci

1. Na implementação do Saci, houve preocupação com este aspecto:
 - (a) Parser KQML otimizado
 - (b) Cache de endereços dos agentes
 - (c) Comunicação na mesma máquina não usa rede
 - (d)
2. Assim como o Saci tem características que outras ferramentas não implementam, o Saci não implementa características que as outras tem e que eventualmente justifiquem a diferença de desempenho:
 - (a) Protocolos de interação
3. Portanto, uma avaliação mais conclusiva depende da implementação destas características no Saci

Lista de Slides

Contexto	1
Motivação	1.1
Objetivo	2
Saci	3
Agentes	3.1
Arquitetura (Como o Saci funciona)	4
Entrada e saída da sociedade	4.1
Envio e recebimento de mensagens	4.2
Anúncio de habilidades	4.3
Monitoramento	4.4
Exemplo de aplicação do monitoramento	4.5
Ferramentas	5

AgentLauncher	5.1
Comparação com outras ferramentas	6
Características	6.1
Desempenho	6.2
Resultado	6.3
Desempenho no tempo	6.4
Conclusões	7
Trabalhos futuros	7.1
Mais informações	7.2
Referências	7.3
Slides para eventuais perguntas	8
Arquitetura do MBox	8.1
Exemplo de programação de um agente	8.2

Um agente applet	8.3
Exemplo de uso das ferramentas	8.4
Razões para o desempenho do Saci	8.5
Lista de Slides	9