Multi-Scheduling in FunLoft

Pejman Attar

INRIA

29 juin 2010

DSLM(Dynamic Script Language with Memory)

We have 3 different level in DSLM :

- Scripts
- Agents
- Sites

Scripts

Script are composed of expression and instruction where expression should finished in one instant but the instruction can take several instant to finished.

Scripts

Syntax :
Expression

$$
\begin{aligned}
e \in Exp^{\tilde{}} ::= {}^{\tilde{}} \\
& x \mid v \mid f(\vec{e}) \\
\mid \quad & !e \mid e := e
\end{aligned}
$$

Scripts

$$
\begin{aligned}
s \in \mathit{Script}^{\sim} ::= {}^{\sim}\ & \mathit{Exp} \\
\mid\ & s; s \\
\mid\ & \texttt{let}\ x\ =\ e\ \texttt{in}\ s\ \texttt{end} \\
\mid\ & \texttt{if}\ e\ \texttt{then}\ s\ \texttt{else}\ s\ \texttt{end} \\
\mid\ & \texttt{loop}\ s\ \texttt{end} \mid \texttt{repeat}\ e\ \texttt{do}\ s\ \texttt{end} \\
\mid\ & \texttt{cooperate} \\
\mid\ & \texttt{launch}\ m\ (\vec{e}) \\
\mid\ & \texttt{generate}\ \mathit{ev}\ \texttt{with}\ v \mid \texttt{await}\ \mathit{ev}\ \texttt{with}\ l \mid \texttt{do}\ s\ \texttt{watching}\ e \\
\mid\ & \texttt{drop}\ s\ \texttt{in}\ \mathit{site} : \mathit{Ag} \\
\mid\ & \texttt{letagent}\ \mathit{Ag} = s\ \texttt{in}\ \mathit{site}\texttt{end} \\
\mid\ & \texttt{migrate}\ \mathit{Ag}\ \texttt{to}\ \mathit{site}
\end{aligned}
$$

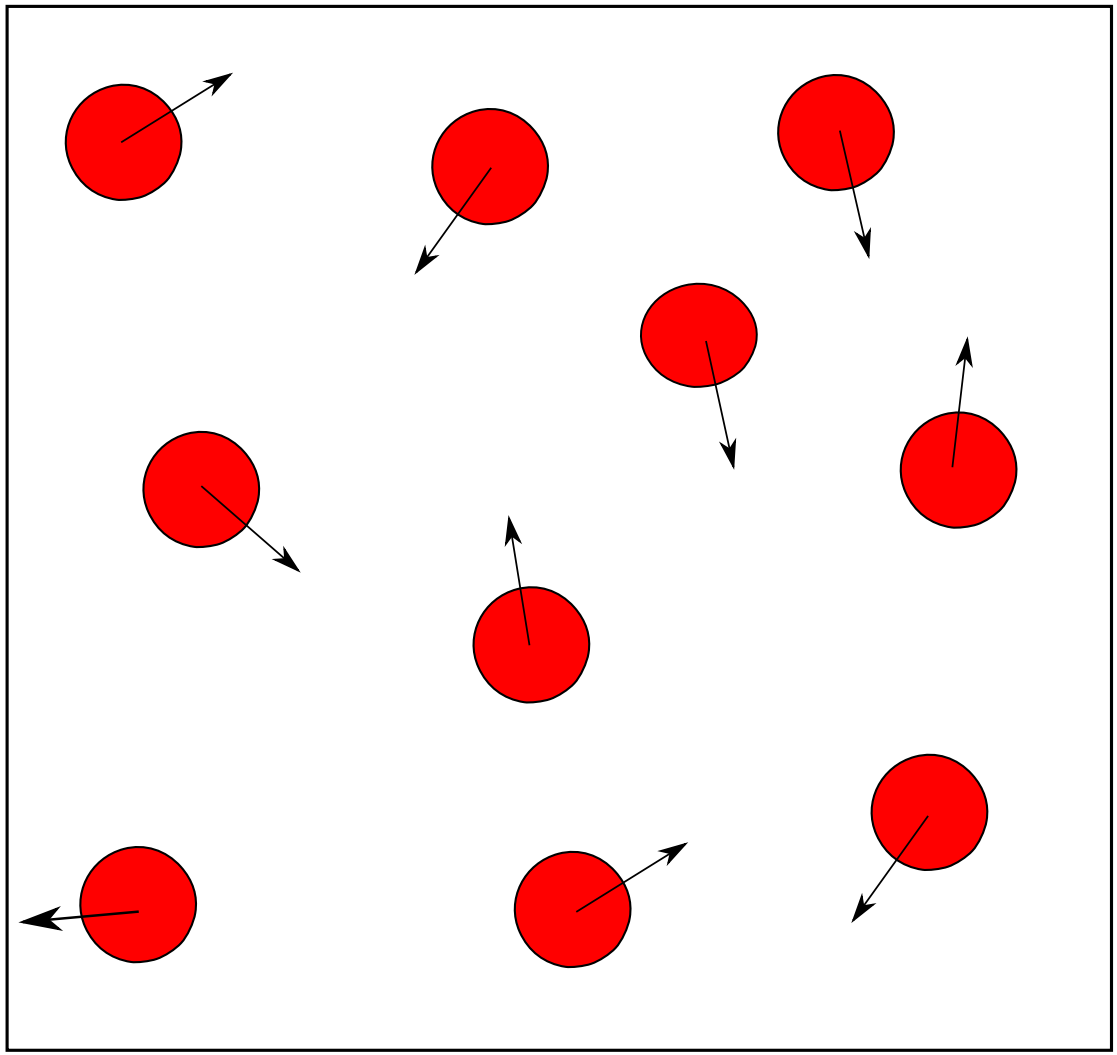
Agents

Each agent containe a script(parallel scripts). Each agent has its own memory, and a script that is not blonging to the agent cannot access to its memory.

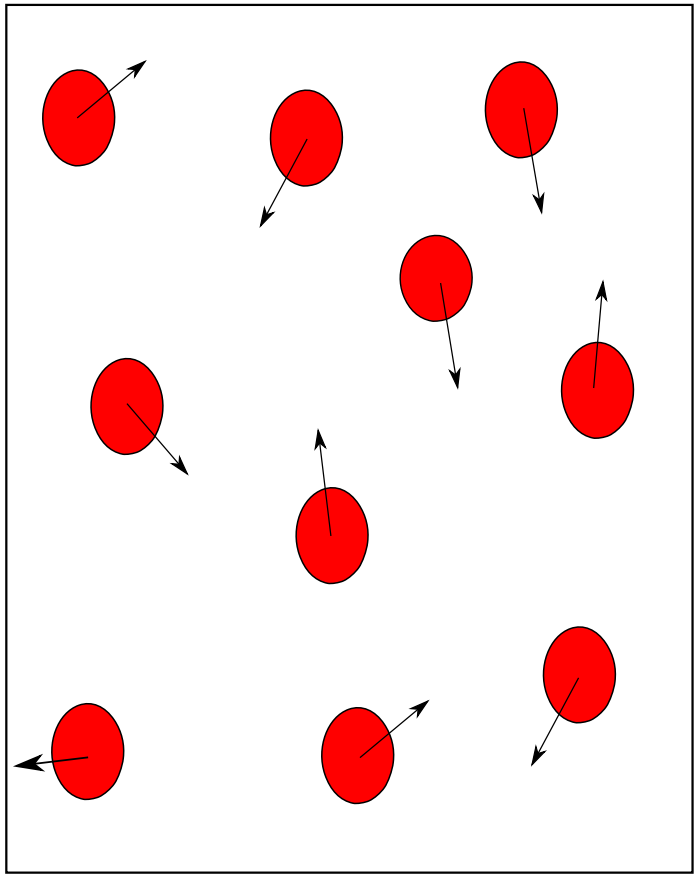
Sites

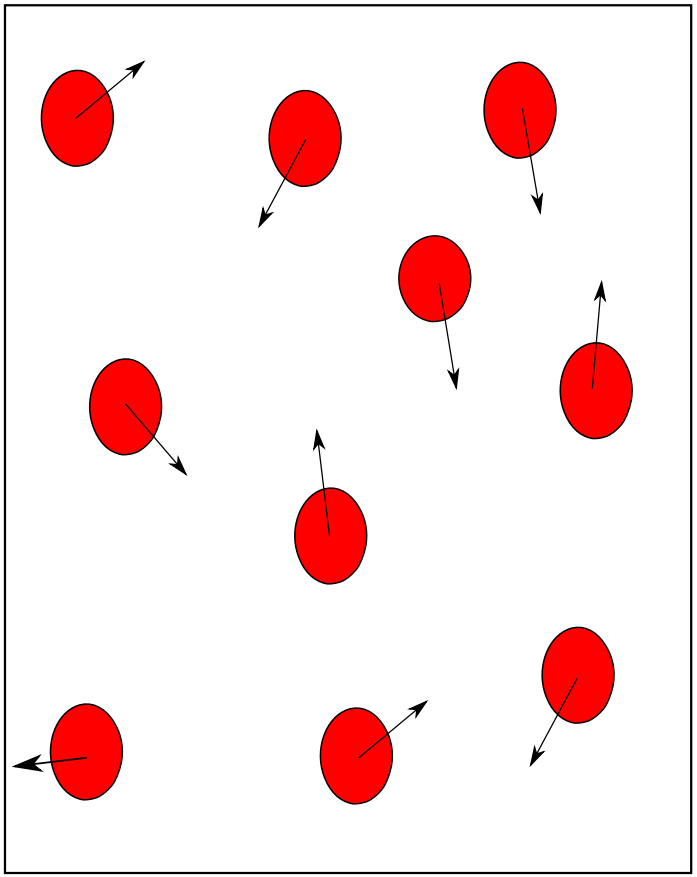
Sites is conainte a set of Agents which are executed in the same peace, and a set of events which are produce during the courant instant on the site.

Example : Simulation of gas molecules(1 room)

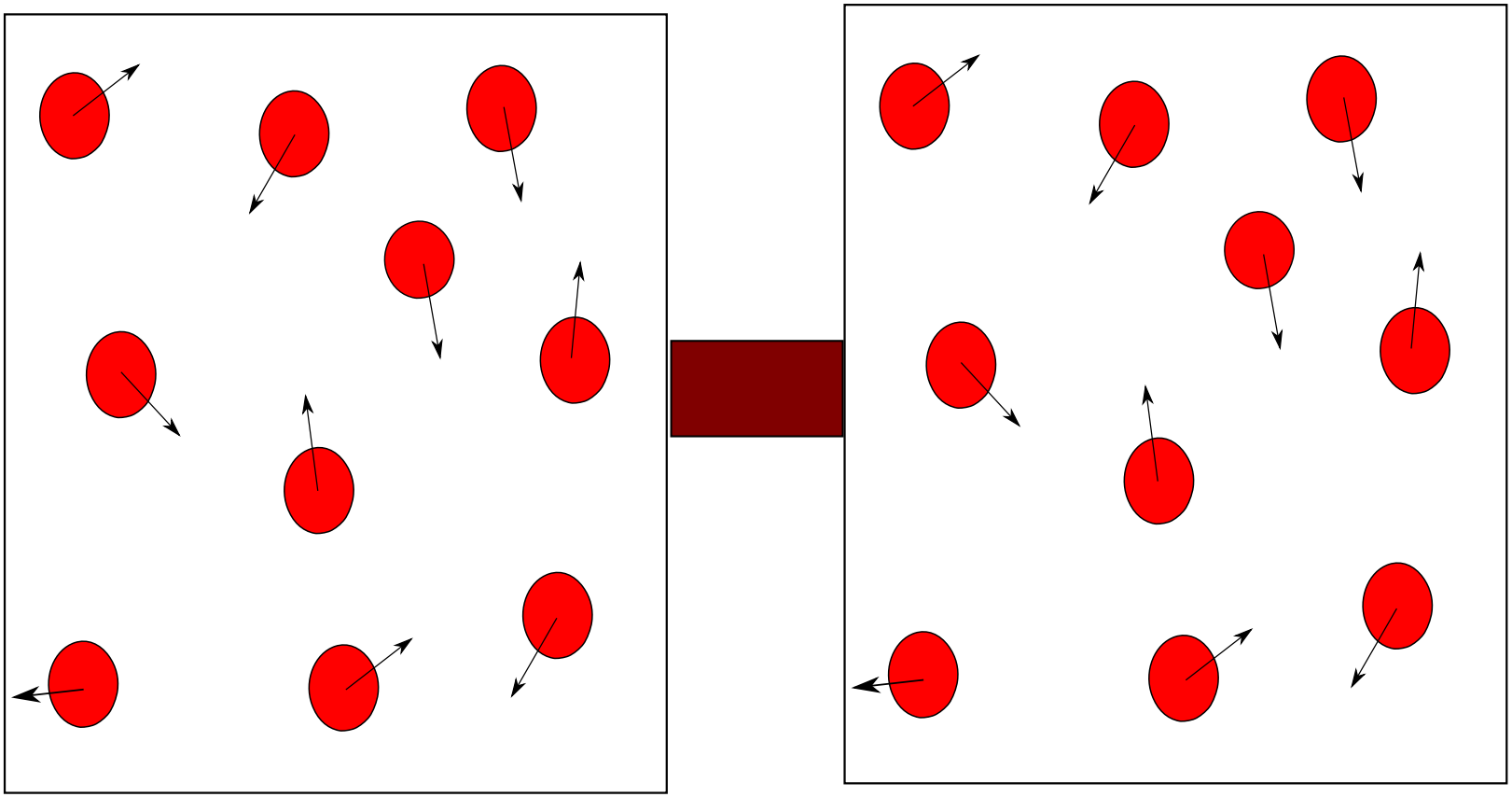


Example : Simulation of gas molecules(2 room)

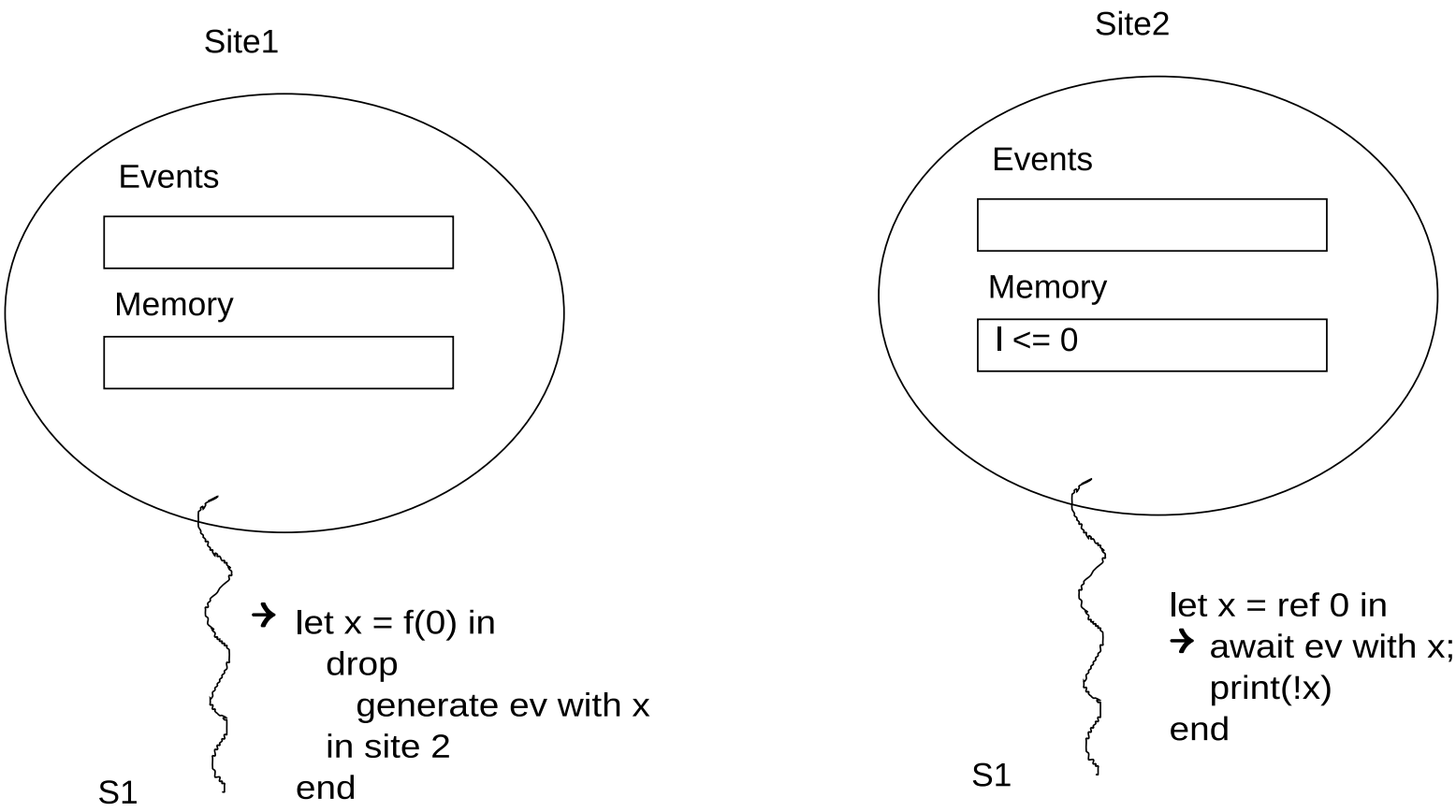




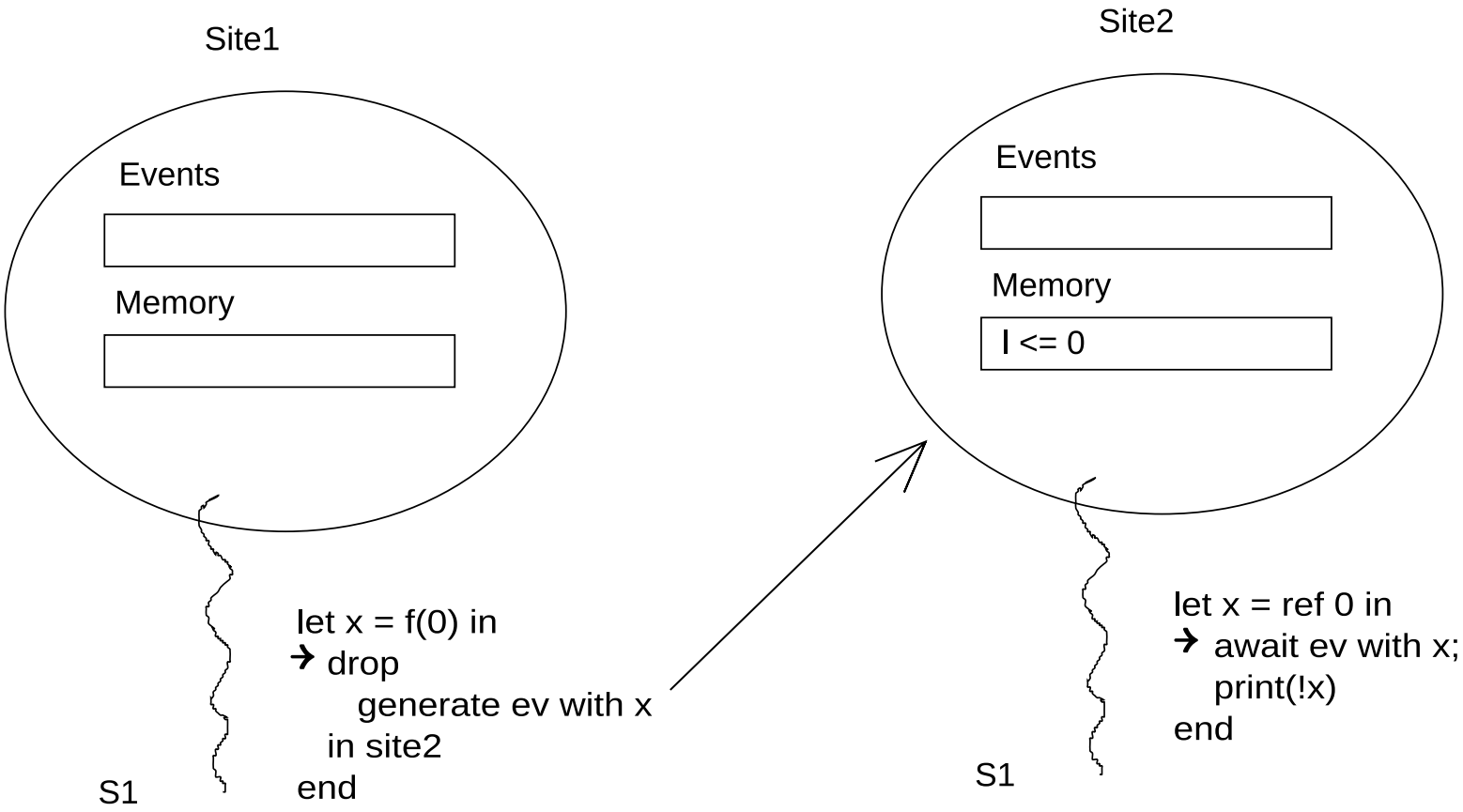
Example : Simulation of gas molecules(2 room with a connection)



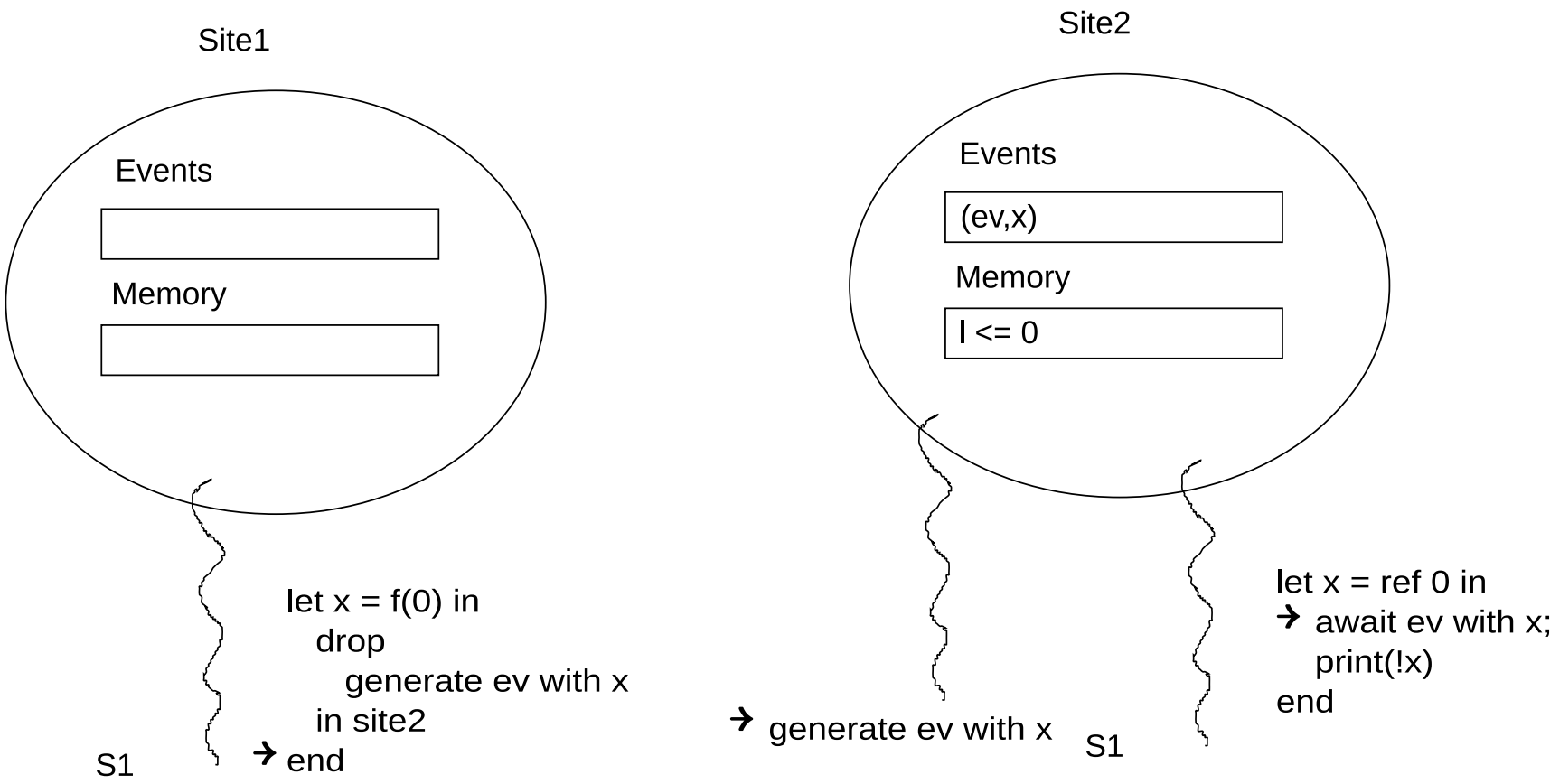
Example : Compute on a remote site



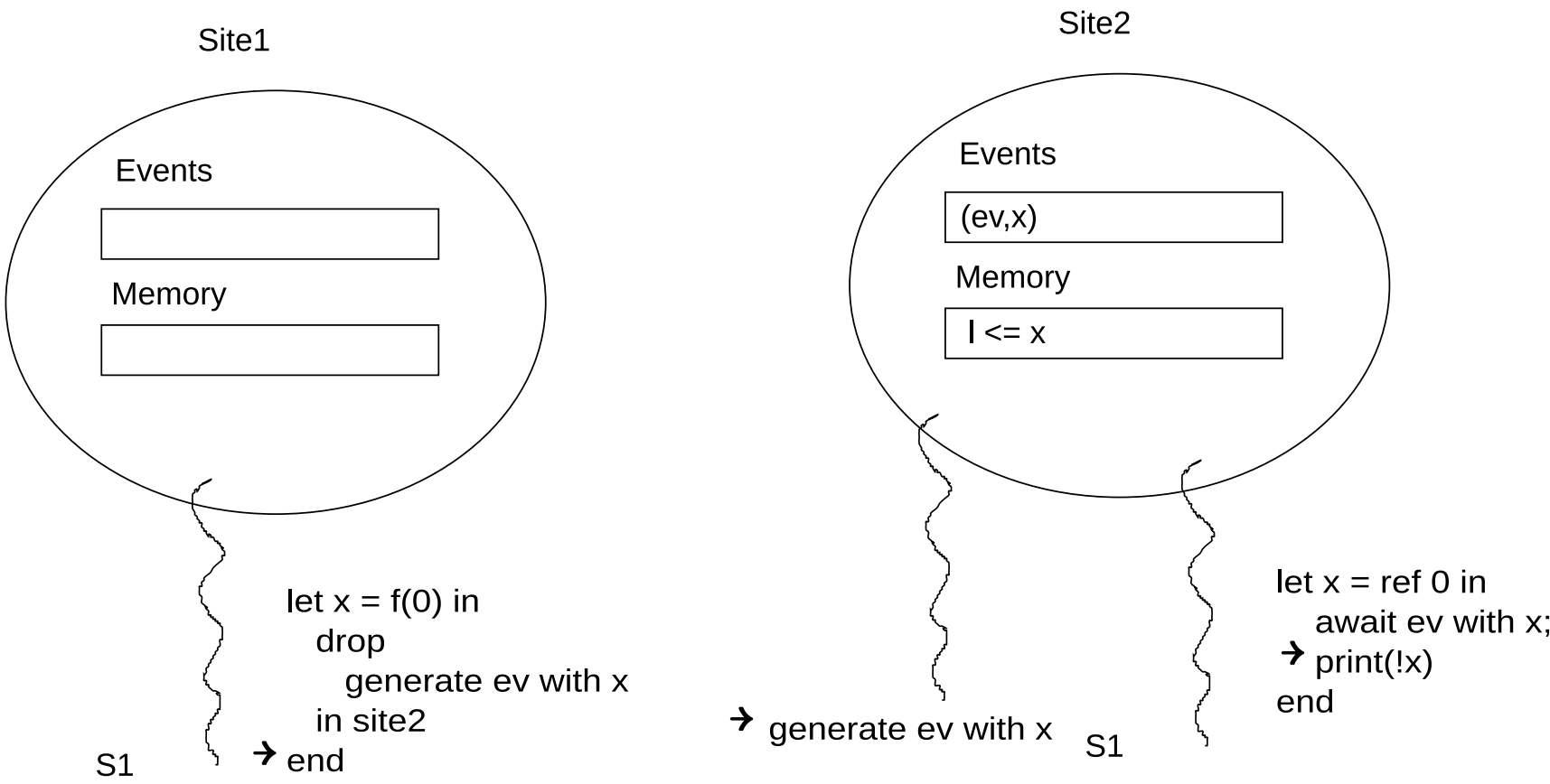
Example : Compute on a remote site



Example : Compute on a remote site



Example : Compute on a remote site



Dynamique scheduling

We want to use all the disponible resouces (cores). To resoulve this problem we propose:

- Expansion and contraction of a site
- Auto-Migration of Agents

Schedulers

Schedulers are the native thread which are mapped to each core. Each scheduler can containe one or several agents.

Sites

A site is a set of synchronized schedulers which means all the schedulers belonging to the same site are executed in a same pace.

Expansion and Contraction of a site

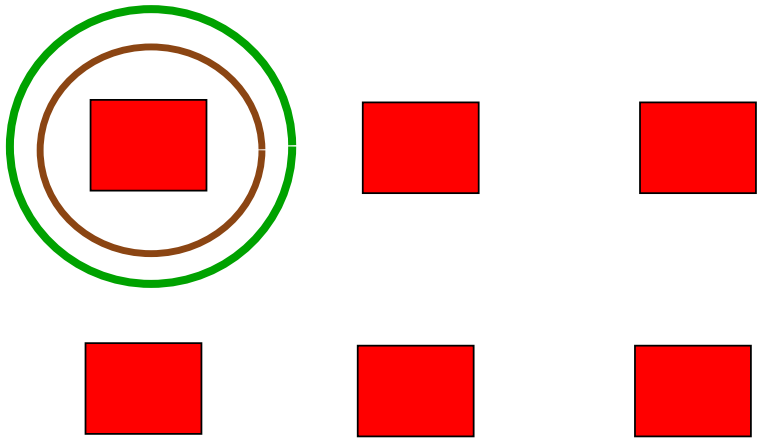
- Expansion of a site

$$\ldots, [sched : \{Ag, Ag', \ldots\}, \ldots], \cdots \rightarrow$$
$$\ldots, [sched : \{Ag, \ldots\}, sched' : \{Ag'\}, \ldots], \ldots$$

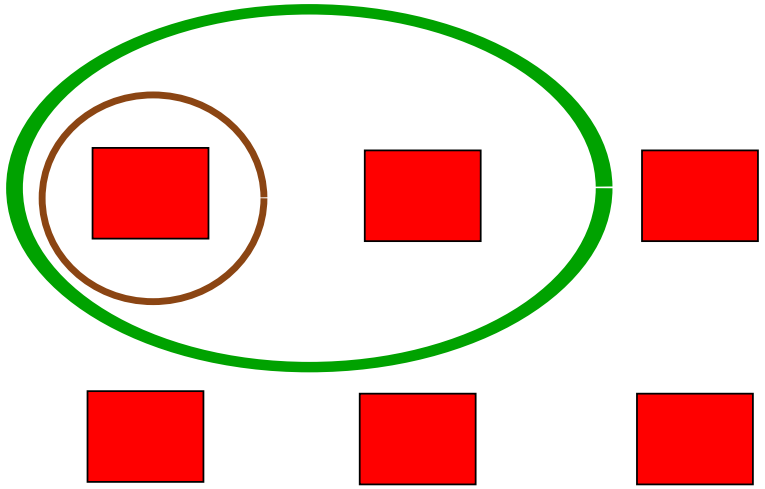
- Contration of a site

$$\ldots, [sched : \{Ag, \ldots\}, sched' : \{Ag'\}, \ldots], \cdots \rightarrow$$
$$\ldots, [sched : \{Ag, Ag', \ldots\}, \ldots], \ldots$$

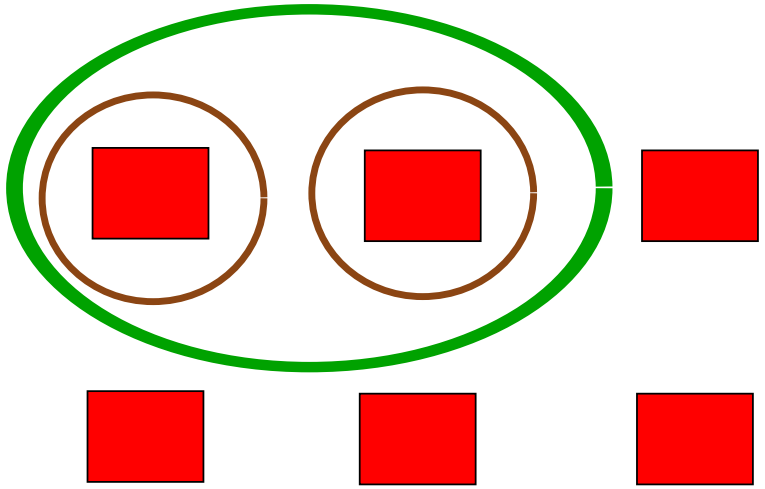
Expansion and Contraction of a site



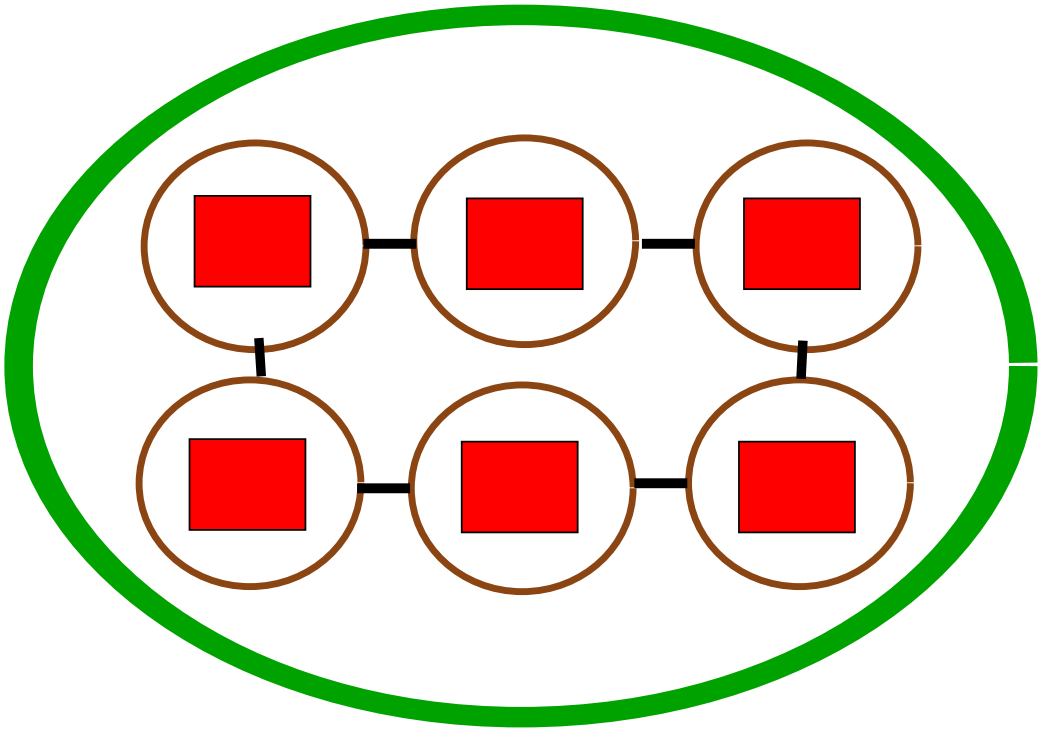
Expansion and Contraction of a site



Expansion and Contraction of a site



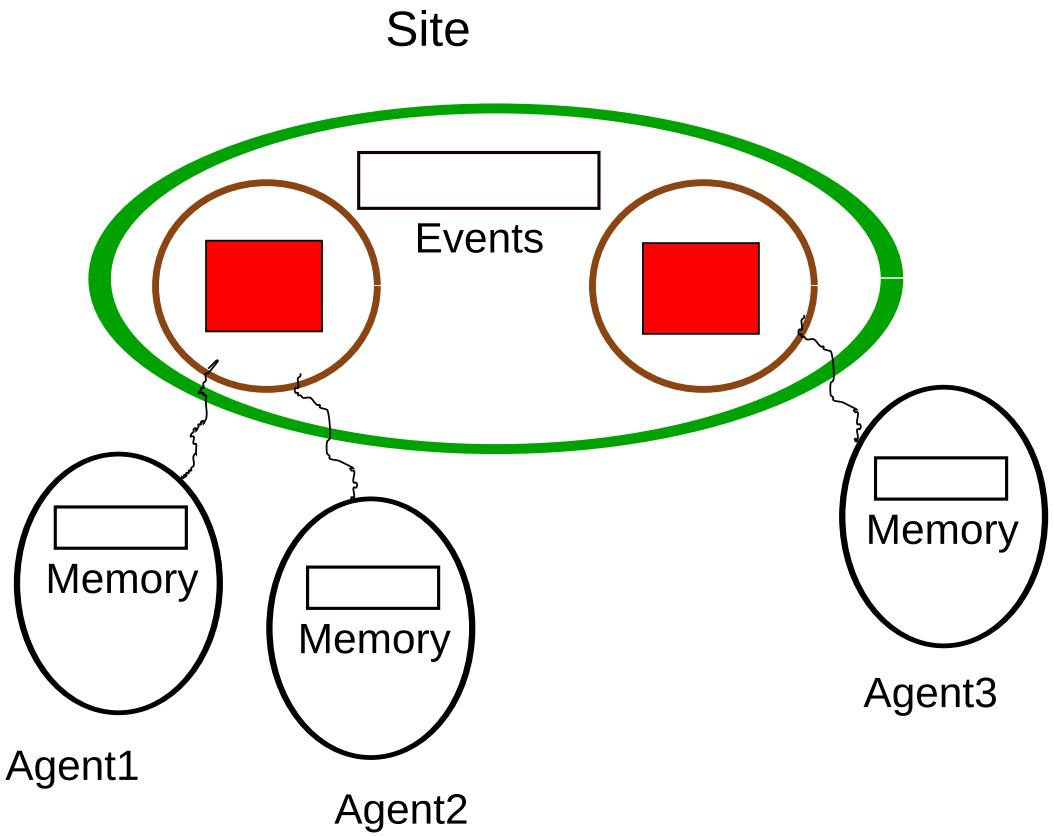
Expansion and Contraction of a site

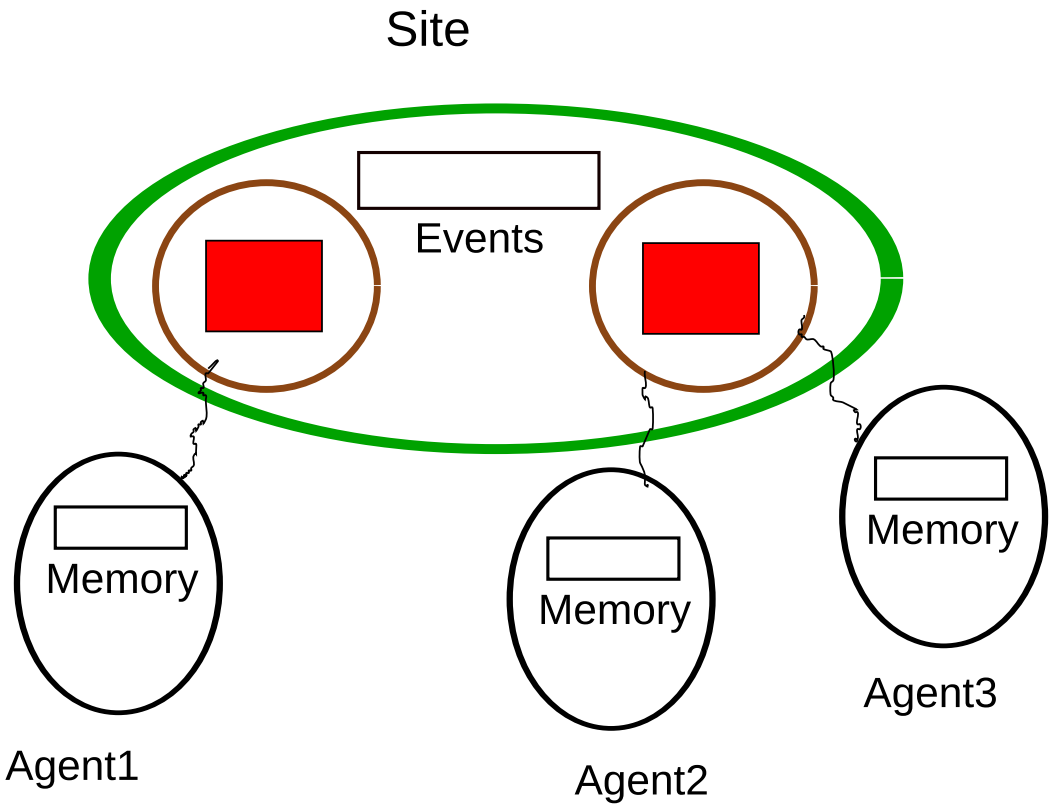


Auto migration of agents

- Automatic migration rule

$$\ldots, [sched : \{Ag, \ldots\}, sched' : \{Ag', \ldots\}, \ldots], \ldots$$
$$\rightarrow \ldots, [sched : \{Ag, Ag', \ldots\}, sched' : \{\ldots\}, \ldots], \ldots$$


Site
Events
Memory
Memory
Memory
Agent1
Agent2
Agent3


Site
Events
Memory
Memory
Memory
Agent1
Agent2
Agent3

Next Step

- Soundness of Typing
- Fixpoint
- Implementation