Agent-oriented Reasoning with Ω-ANTS

Christoph Benzmüller

joint work with: Volker Sorge, Manfred Kerber, Mateja Jamnik

Fachrichtung Informatik
Universität des Saarlandes
Saarbrücken, Germany

School of Computer Science
The University of Birmingham
Birmingham, England

Motivation – Cognitive Perspective

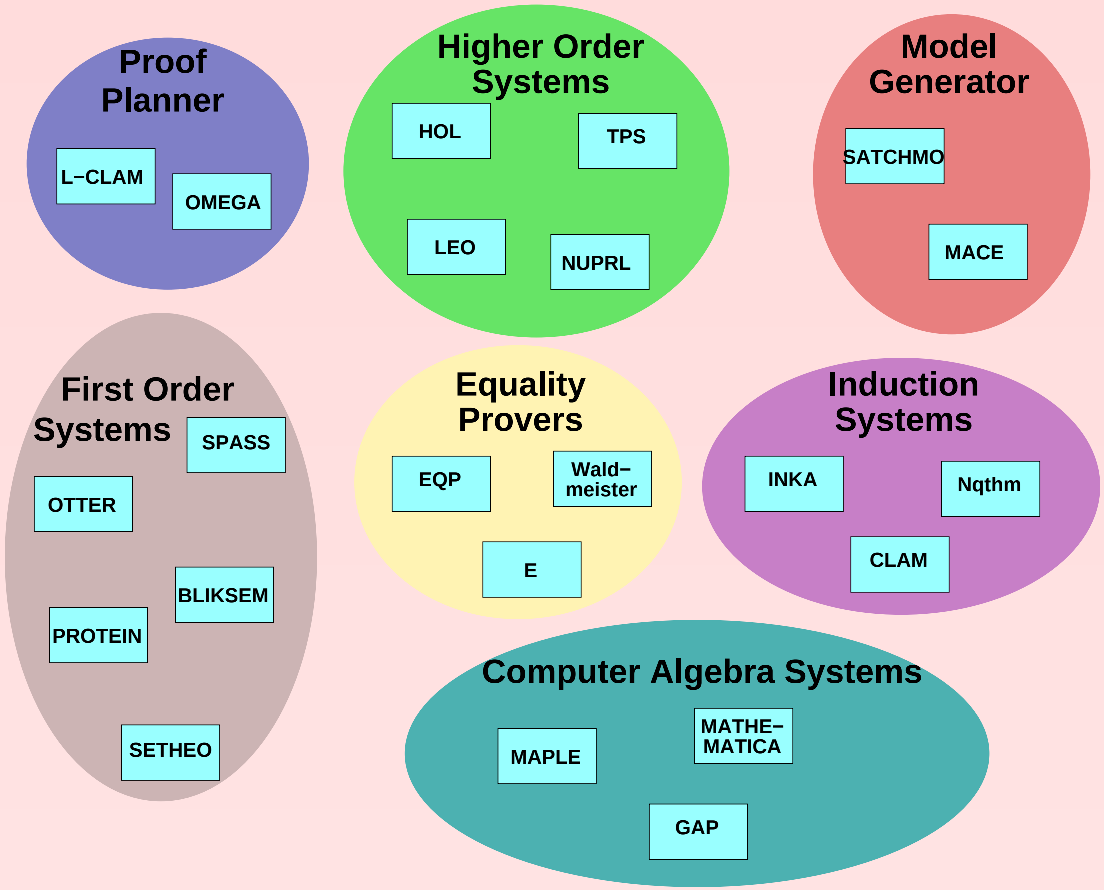
To solve complex mathematical problems

- different specialists may have to bring in their expertise and flexibly cooperate
- important: agreement on a common language

A single mathematician

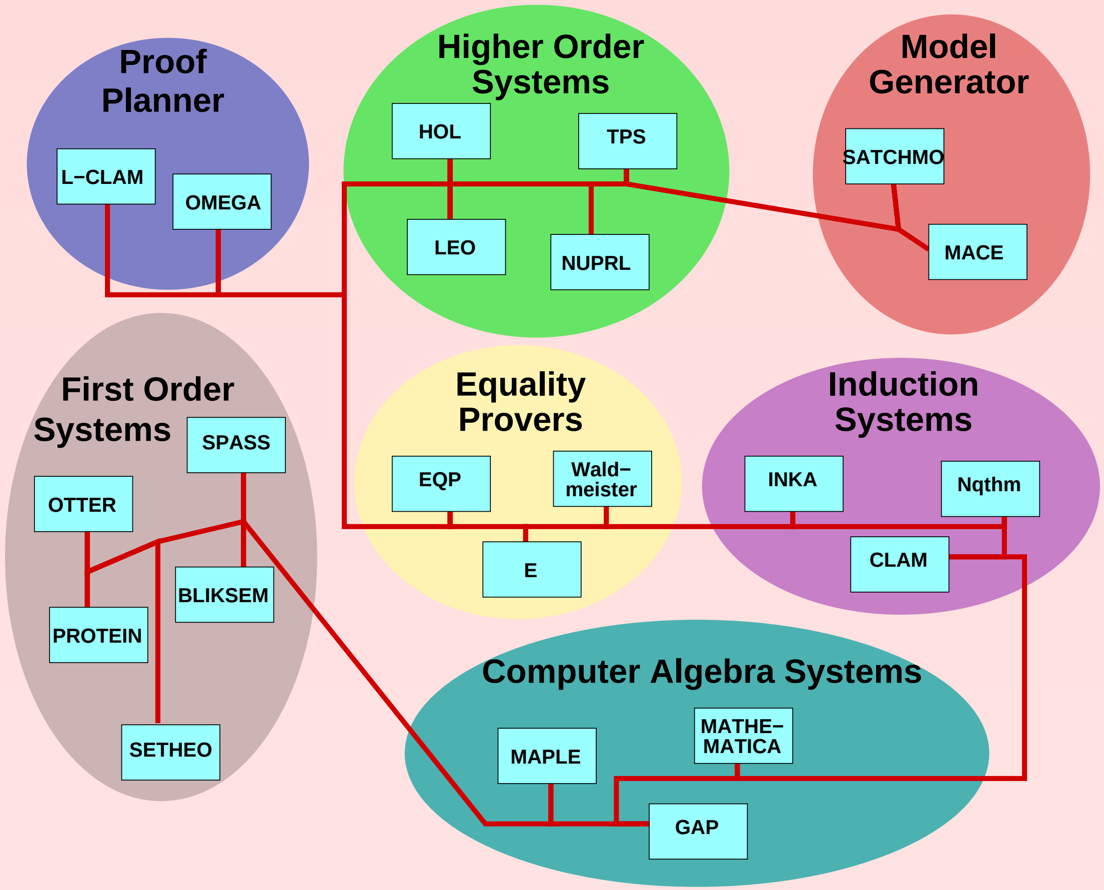
- possesses a large repertoire of specialized reasoning and problem solving techniques
- uses experience and intuition to flexibly combine them in an appropriate way

Motivation – Existing Systems



- heterogeneous
- different niches

Motivation – Existing Systems



- heterogeneous
- different niches
- hardwired integrations
- system networks: MATHWEB, PROSPER integration-infrastructure

How to realize a dynamic and flexible interplay?

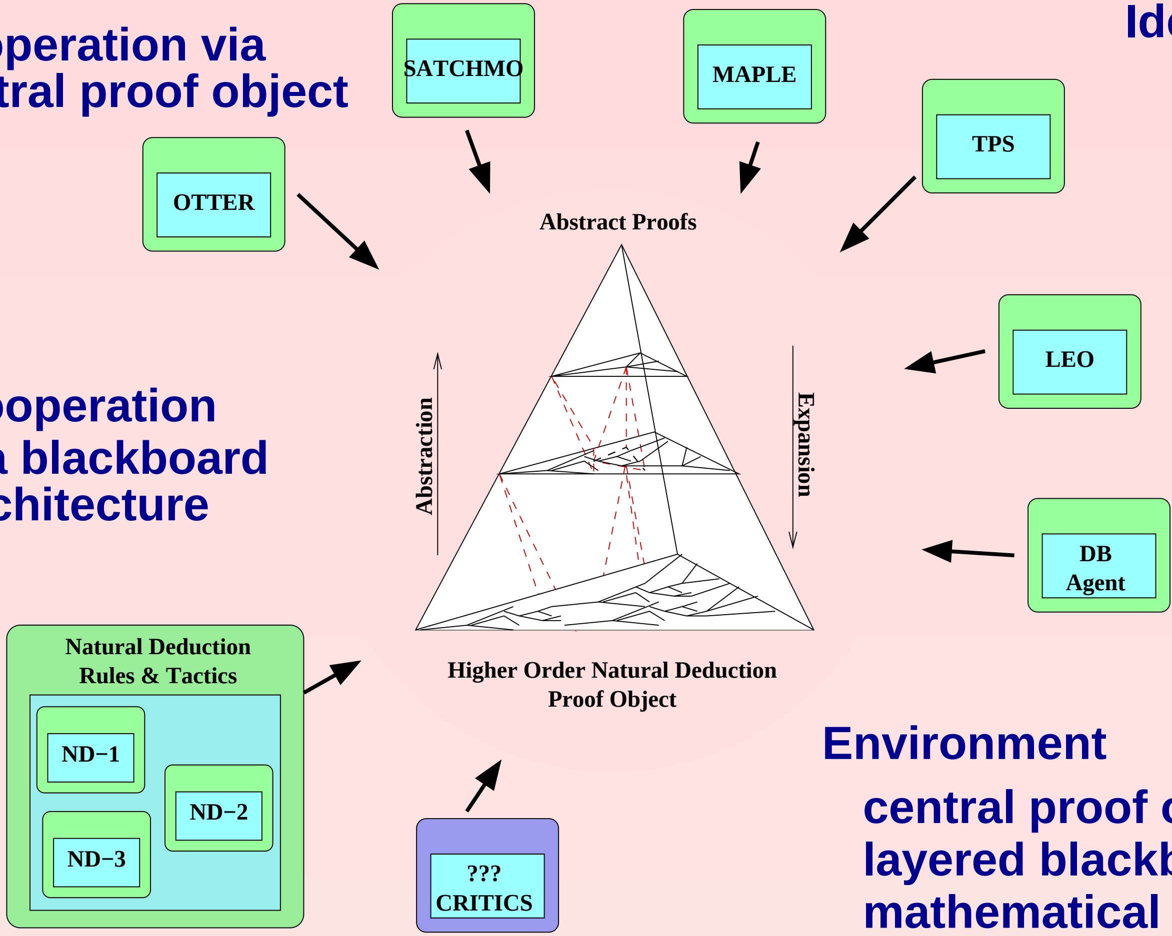
Talk Outline

- Sketch of the approach
- A motivating example: HO- and FO-ATP
- System realization: the overall architecture
- System realization: Ω-ANTS as core
- First experiments
- Conclusion and further work

The Approach – Abstract Perspective

Ωmega

Cooperation via
central proof object



Cooperation
via blackboard
architecture

ND−Backbone

Idea
- decentralised control

Agents
- reactive
- proactive
- heterogenous
- simple & complex
- cooperative & competetive
- distributed via MathWeb
- run−time definable
- resource adapted

Environment
- central proof object
- layered blackboards (local communication)
- mathematical database

Example – HO- and FO-ATP

favorite-numbers$(\lambda x. \text{odd}(x) \land \text{square}(x))$
$\Rightarrow$ favorite-numbers$(\lambda x. \text{square}(x) \land (\text{square}(x) \Rightarrow \text{odd}(x)))$???

Higher Order ATP with LEO

C_1: favorite-numbers$(\lambda x. \text{odd}(x) \land \text{square}(x))$

unifies (semantically) with

C_2: $\neg$favorite-numbers$(\lambda x. \text{square}(x) \land (\text{square}(x) \Rightarrow \text{odd}(x)))$

iff the following first order clauses can be contradicted

First Order ATP with OTTER

square(N)	$\neg$odd$(N) \lor \neg$square(N)
odd$(N) \lor \neg$square(N)	odd$(N) \lor$ square(N)

Example – HO- and FO-ATP

$\textsf{fn} = \textsf{favorite-numbers}$

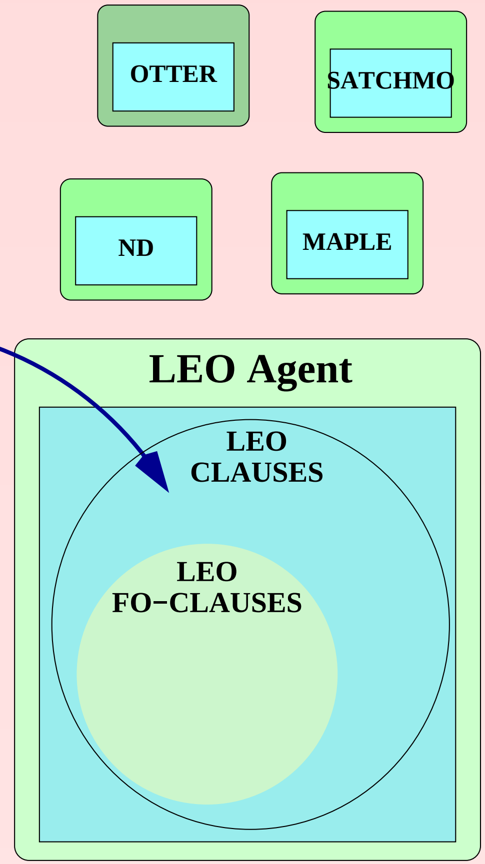
$\textsf{o} = \textsf{odd}$

$\textsf{s} = \textsf{square}$

$C_1 : \textsf{fn}(\lambda x.\, \textsf{o}(x) \land \textsf{s}(x))$

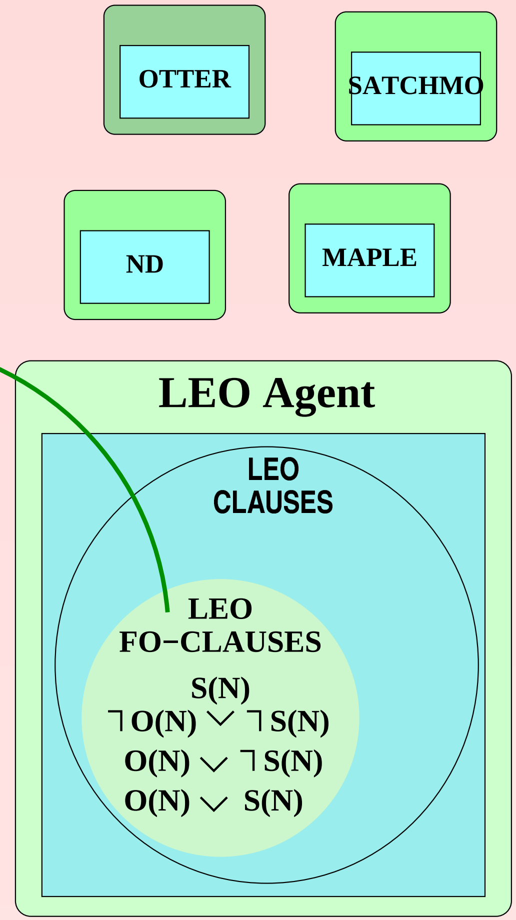
$C_2 : \neg\textsf{fn}(\lambda x.\, \textsf{s}(x) \land (\textsf{s}(x) \Rightarrow \textsf{o}(x)))$

$\textsf{fn}(\lambda x.\, \textsf{o}(x) \land \textsf{s}(x))$	*Hyp*
$\vdots$?	
$\textsf{fn}(\lambda x.\, \textsf{s}(x) \land (\textsf{s}(x) \Rightarrow \textsf{o}(x)))$	*open*



Example – HO- and FO-ATP

$$\frac{\dfrac{\mathsf{fn}(\lambda x.\, \mathsf{o}(x) \wedge \mathsf{s}(x)) \quad \neg\mathsf{fn}(\lambda x.\, \mathsf{s}(x) \wedge (\mathsf{s}(x) \Rightarrow \mathsf{o}(x)))}{\mathsf{s}(N) \wedge (\neg\mathsf{o}(N) \vee \neg\mathsf{s}(N)) \wedge (\mathsf{o}(N) \vee \neg\mathsf{s}(N)) \wedge (\mathsf{o}(N) \vee \mathsf{s}(N))}\ \mathit{LEO} \qquad \vdots\ ? \qquad \bot}{\mathsf{fn}(\lambda x.\, \mathsf{s}(x) \wedge (\mathsf{s}(x) \Rightarrow \mathsf{o}(x)))}\ \text{indirect}$$

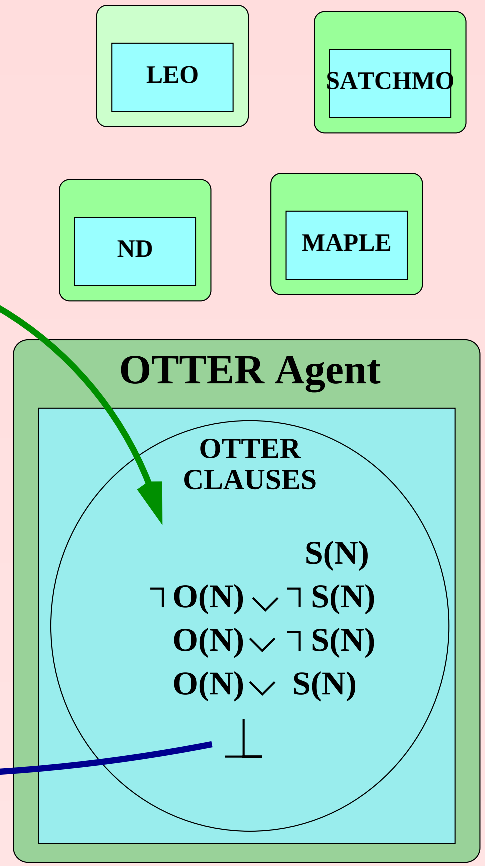


Example – HO- and FO-ATP

$$\frac{\dfrac{\mathsf{fn}(\lambda x.\, \mathsf{o}(x) \wedge \mathsf{s}(x)) \qquad \neg\mathsf{fn}(\lambda x.\, \mathsf{s}(x) \wedge (\mathsf{s}(x) \Rightarrow \mathsf{o}(x)))}{\mathsf{s}(N) \wedge (\neg\mathsf{o}(N) \vee \neg\mathsf{s}(N)) \wedge (\mathsf{o}(N) \vee \neg\mathsf{s}(N)) \wedge (\mathsf{o}(N) \vee \mathsf{s}(N))}\,LEO}{\perp}\,OTTER$$

$$\frac{\perp}{\mathsf{fn}(\lambda x.\, \mathsf{s}(x) \wedge (\mathsf{s}(x) \Rightarrow \mathsf{o}(x)))}$$

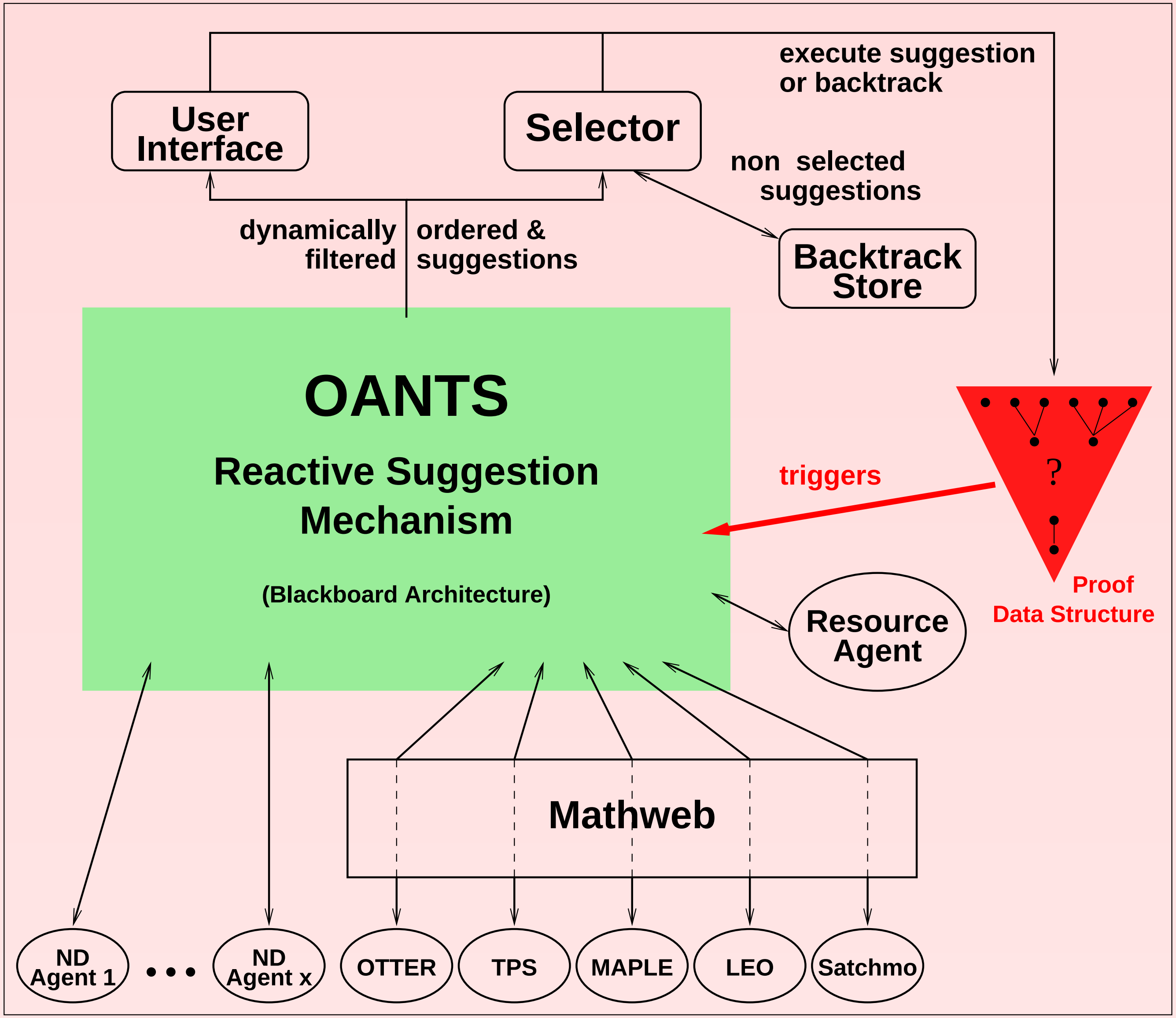
indirect



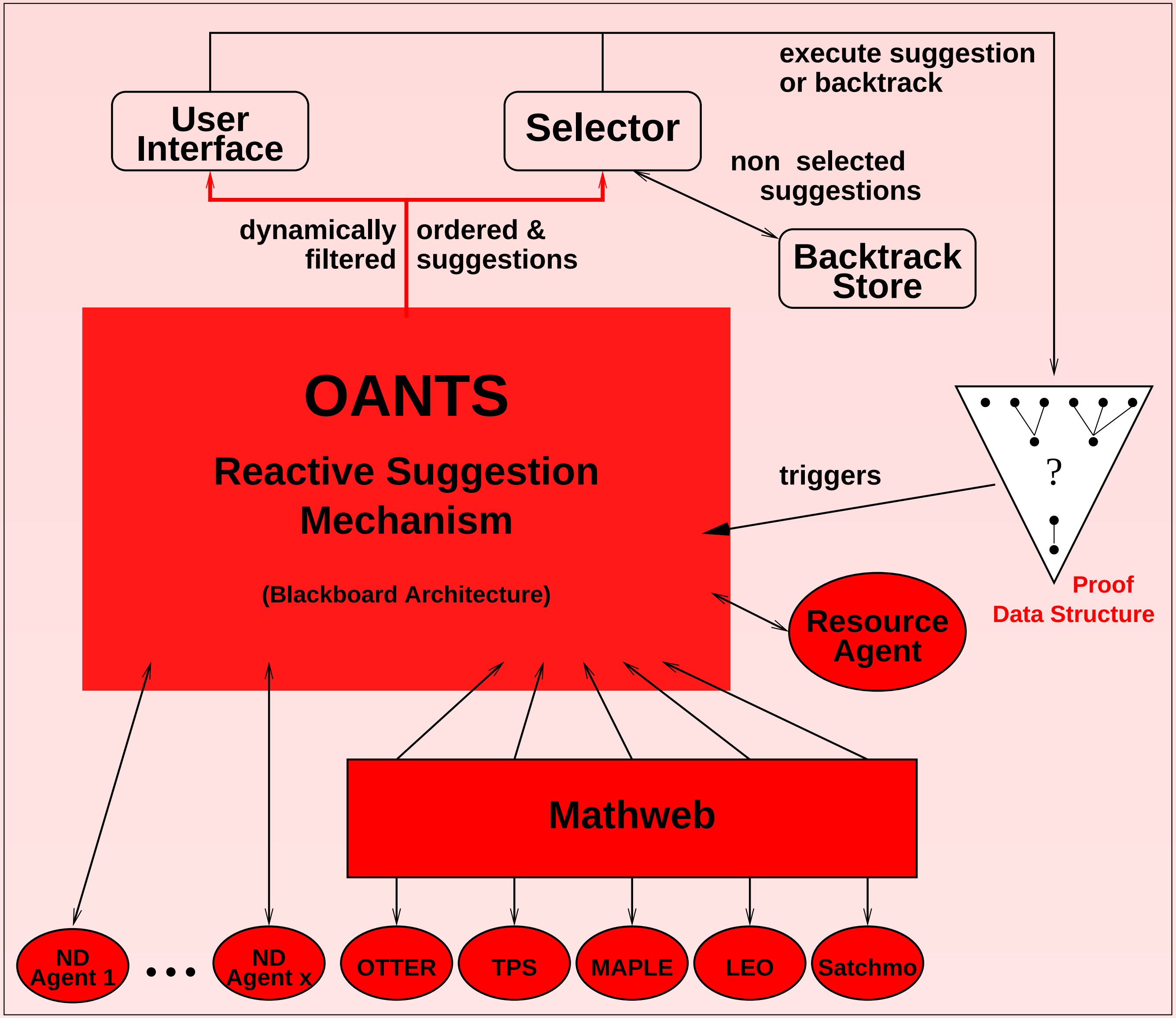
Realization – Main Components

- ΩMEGA's proof data structure (natural deduction with abstraction facilities)
- Ω-ANTS blackboard architecture
- MATHWEB-system
- Various external systems integrated via MATHWEB
- Translation modules like TRAMP ATP $\Longrightarrow$ ND) and SAPPER (CAS $\Longrightarrow$ ND)
- ND Intercalation proof search (NIC: Byrnes & Sieg)
- Implementation: concurrent CLOS

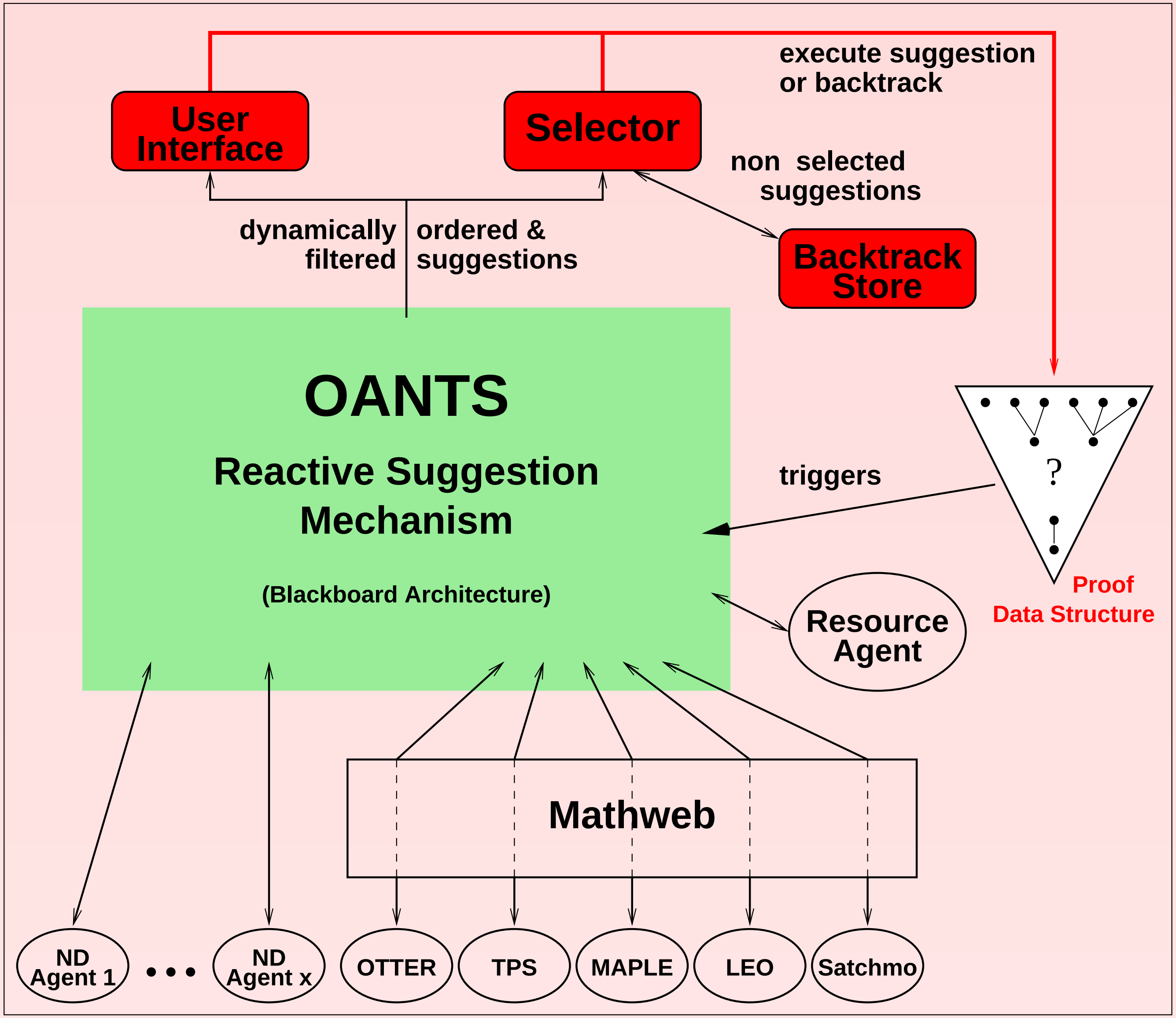
Realization – Architecture



Realization – Architecture



Realization – Automating Ω-ANTS



Realization – Resource Aspects

Resource adapted behavior

- clock speed **low** — automatic proof by single ATP
- clock speed **medium** — cooperative proof
- clock speed **high** — attack at ND level

Experiments: resource adaptivity (in interactive sessions)

agents decide to be inactive/active wrt varying clock speed

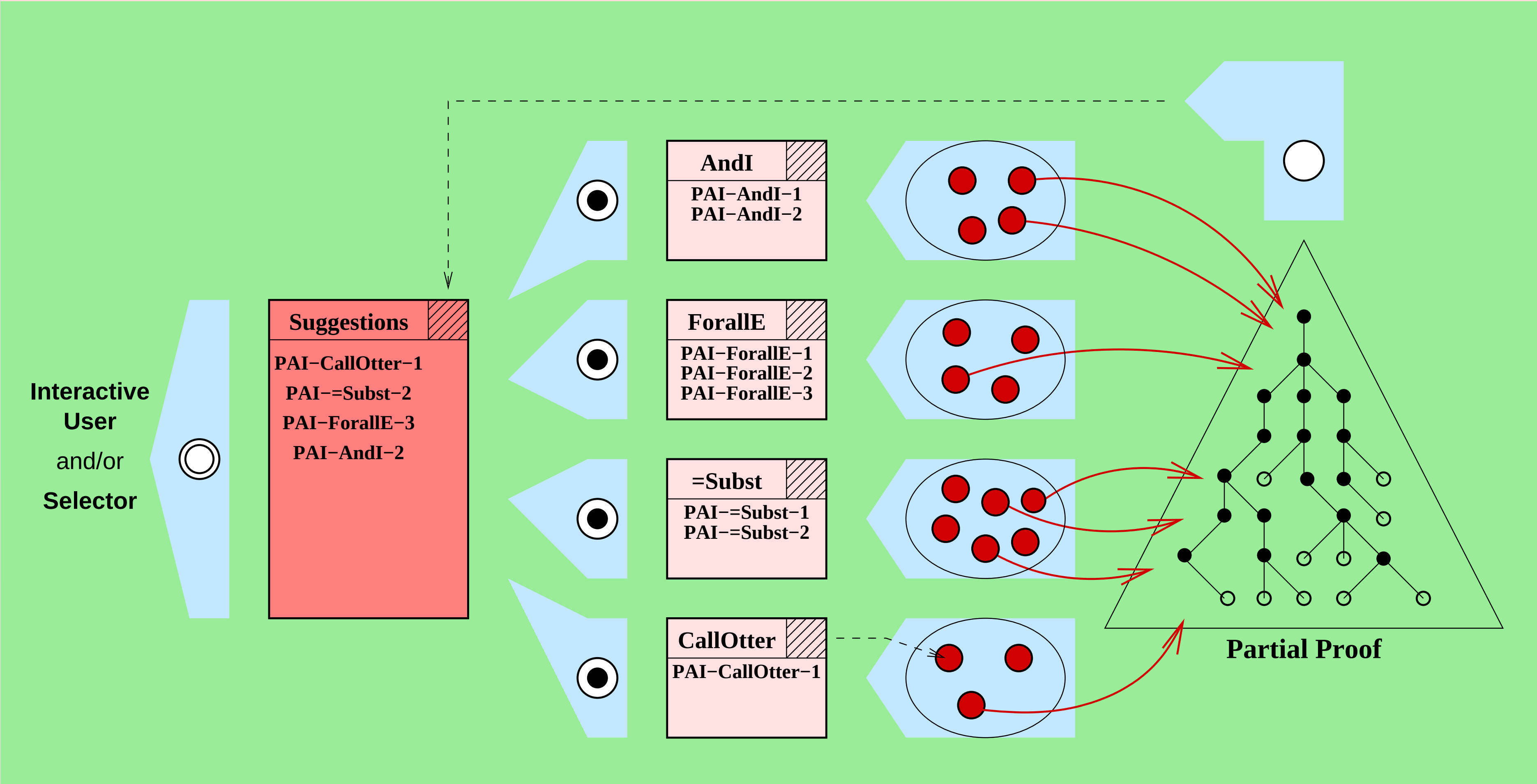
Ω-ANTS – Idea

$$\frac{\texttt{Left}:A \quad \texttt{Right}:B}{\texttt{Conj}:A \wedge B}\ \wedge\texttt{I}$$

$$\frac{\texttt{Ass}_1:A_1 \quad \ldots \quad \texttt{Ass}_n:A_n}{\texttt{Conc}:C}\ \texttt{OTTER}(\texttt{P}_1:f_1,\ldots,\texttt{P}_m:f_m)$$

- Distributed applicability checks for:
  proof rules, tactics, external systems, etc.
- Further distributed processes for single parameter or parameter combinations
- Two layered blackboard architecture
- Anytime character

Ω-ANTS – Blackboard Architecture



Task: **Determine the most promising commands (rules, tactics, external systems) in the current proof context**

Ω-ANTS – Parameter Agents

$$\begin{array}{c} \forall x.\, \mathbf{fn}(x) \\ \mathbf{fn}(N) \Rightarrow \mathbf{o}(N) \wedge \mathbf{s}(N) \\ \mathbf{o}(N) \\ \mathbf{s}(N) \\ \vdots\ ? \\ \mathbf{o}(N) \wedge \mathbf{s}(N) \end{array}$$

Blackboard: $\dfrac{\texttt{Left:}A \quad \texttt{Right:}B}{\texttt{Conj:}A \wedge B} \wedge\texttt{I}$

∧I-Agent-1
search for: `Conj`
required:
excluded: `Left,Right`
job: ***find open conjunctions***

∧I-Agent-2
search for: `Conj`
required: `Left`
excluded: `Right`
job: ***find open conjunctions with left conjunct*** **'Left'**

Blackboard: $\dfrac{\texttt{Left:}A \quad \texttt{Right:}B}{\texttt{Conj:}\mathbf{o}(N) \wedge \mathbf{s}(N)} \wedge\texttt{I}$, $\dfrac{\texttt{Left:}A \quad \texttt{Right:}B}{\texttt{Conj:}A \wedge B} \wedge\texttt{I}$

Ω-ANTS – Parameter Agents

$$\begin{array}{c} \forall x. \mathsf{fn}(x) \\ \mathsf{fn}(N) \Rightarrow \mathsf{o}(N) \land \mathsf{s}(N) \\ \mathsf{o}(N) \\ \mathsf{s}(N) \\ \vdots\ ? \\ \mathsf{o}(N) \land \mathsf{s}(N) \end{array}$$

BB: $\dfrac{\texttt{Left:}A \quad \texttt{Right:}B}{\texttt{Conj:}\mathsf{o}(N) \land \mathsf{s}(N)} \land\texttt{I}$, ...

∧I-Agent-3	∧I-Agent-4
search for: Left	**search for:** Right
required: Conj	**required:** Conj
excluded:	**excluded:**
job: ...	**job:** ...

BB: $\dfrac{\texttt{Left:}\mathsf{o}(N) \quad \texttt{Right:}\mathsf{s}(N)}{\texttt{Conj:}\mathsf{o}(N) \land \mathsf{s}(N)} \land\texttt{I}$, $\dfrac{\texttt{Left:}A \quad \texttt{Right:}\mathsf{s}(N)}{\texttt{Conj:}\mathsf{o}(N) \land \mathsf{s}(N)} \land\texttt{I}$, $\dfrac{\texttt{Left:}\mathsf{o}(N) \quad \texttt{Right:}B}{\texttt{Conj:}\mathsf{o}(N) \land \mathsf{s}(N)} \land\texttt{I}$, ...

Ω-ANTS – Adding Components

Adding an external system (or rule, tactic, method):

- **provide a command**

$$\frac{\texttt{Ass-1}:A_1 \quad \ldots \quad \texttt{Ass-n}:A_n}{\texttt{Conc}:C}\ \texttt{OTTER}(\texttt{P-1}:f_1,\ldots,\texttt{P-m}:f_m)$$

- **define parameter agents**
- **adapt heuristic filter (utility function)**
- **command agent & blackboard created automatically**
- **optional (but important for sceptical approach): result transformation in ND**

Ω-ANTS – Further Aspects

- **Employ structure of natural deduction proof objects**
  guide search along current focus
- **Declarative agent specification language**
  uniform way to define new agents
  run-time modifiability of agent societies & heuristic filters
- **Attempt to a formal semantics**
  mapping agent declarations to simply typed λ-calculus
  some system properties can be modeled
- **Self-evaluation of agents**
  learn about own performance
  broadcast this information via blackboards
  explicit resource reasoning on informed layer

Experiments – Example Classes

Ex1 **Higher order ATP and first order ATP**

$$\forall x, y, z.\, (x = y \cup z) \Leftrightarrow (y \subseteq x \wedge z \subseteq x \wedge \forall v.\, (y \subseteq v \wedge z \subseteq v) \Rightarrow (x \subseteq v))$$

Ex2 **ND based TP, propositional ATP, and model generation**

$\forall x. \forall y. \forall z. ((x \cup y) \cap z) = (x \cap z) \cup (y \cap z)$ **10000 Examples**

$\forall x. \forall y. \forall z. ((x \cup y) \cup z) = (x \cap z) \cup (y \cap z)$ **988 valid / 9012 invalid**

Ex3 **CAS and higher order ATP**

$$\{x | x > gcd(10, 8) \wedge x < lcm(10, 8)\} = \{x | x < 40\} \cap \{x | x > 2\}$$

Ex4 **Tactical TP, first-order ATP, CAS, and higher order ATP**

. . . *group-definition-1* . . . ⇔ . . . group-definition-2 . . .

Experiments – ND, PL-ATP, Models

Ωmega

Conc	$\vdash \forall x. \forall y. \forall z. ((x \cup y) \cap z) = (x \cap z) \cup (y \cap z)$	Forall-I L1
	$\dots$	
L3	$\vdash ((X \cup Y) \cap Z) = (X \cap Z) \cup (Y \cap Z)$	Set-Ext L4
L4	$\vdash \forall e. e \in ((X \cup Y) \cap Z) \leftrightarrow e \in (X \cap Z) \cup (Y \cap Z)$	Forall-I L5
L5	$\vdash E \in ((X \cup Y) \cap Z) \leftrightarrow E \in (X \cap Z) \cup (Y \cap Z)$	Def L6
	$\dots$	
L8	$\vdash ((E \in X \vee E \in Y) \wedge E \in Z) \leftrightarrow ((E \in X \wedge E \in Z) \vee (E \in Y \wedge E \in Z))$	OTTER

Theorem

Experiments – ND, PL-ATP, Models

Conc	$\vdash \forall x. \forall y. \forall z. ((x \cup y) \cup z) = (x \cap z) \cup (y \cap z)$	Forall-I L1
	$\dots$	
L3	$\vdash ((X \cup Y) \cup Z) = (X \cap Z) \cup (Y \cap Z)$	Set-Ext L4
L4	$\vdash \forall e. e \in ((X \cup Y) \cup Z) \leftrightarrow e \in (X \cap Z) \cup (Y \cap Z)$	Forall-I L5
L5	$\vdash E \in ((X \cup Y) \cup Z) \leftrightarrow E \in (X \cap Z) \cup (Y \cap Z)$	Def L6
	$\dots$	
L8	$\vdash ((E \in X \vee E \in Y) \vee E \in Z) \leftrightarrow ((E \in X \wedge E \in Z) \vee (E \in Y \wedge E \in Z))$	**SATCHMO**

Countermodel: $G \in Z \wedge G \notin X \wedge G \notin Y$

Experiments – ND, CAS, HO-ATP

$$\{x|x > gcd(10,8) \wedge x < lcm(10,8)\} = \{x|x < 40\} \cap \{x|x > 2\}$$

Conc	$\vdash (\lambda x\text{.}\, x > gcd(10,8) \wedge x < lcm(10,8)) = (\lambda x\text{.}\, x < 40) \cap (\lambda x\text{.}\, x > 2)$	**CAS** L1
L1	$\vdash (\lambda x\text{.}\, x > 2 \wedge x < 40) = (\lambda x\text{.}\, x < 40) \cap (\lambda x\text{.}\, x > 2)$	Def L3
L3	$\vdash (\lambda x\text{.}\, x > 2 \wedge x < 40) = (\lambda x\text{.}\, x < 40 \wedge x > 2)$	**LEO**

Related Work

- Parallel & distributed theorem proving [Bonacina 2000]
- TECHS & TEAMWORK approach [Denzinger/Fuchs 1999]
  - filtered exchange of clauses between first-order provers
  - no higher-order systems and no CAS
  - no explicit proof object
  - no user orientation
- Concurrent theorem proving [Fisher 1997]
  METATEM (temporal logics) [Fisher 1994]
- Multi agent proof-planning [Fisher/Ireland 1998]
- . . . agent based architectures, layered architectures . . .

Conclusion

- No opposition to classic ATP (benefit from their strengths!)
- New: active (vs passive) character of integrated systems
- New: flexible, dynamic combination of reasoning tools

$$\text{Automate} \longleftarrow \Omega\text{-ANTS}(rc_1, \ldots, rc_n)$$

$$\text{Interact with} \longleftarrow \Omega\text{-ANTS}(rc_1, \ldots, rc_n)$$

- Parameterised calls Ω-ANTS$(rc_1, \ldots, rc_n)$ contrast fixed call-hierachies of traditional tactics and methods
- Relevance:
  - high for interactive TP (no waste of resources, active vs passive, …)
  - for ATP ???, more work needed

Problems and Future Work

- Short-term goals
  - adaption to a new system environment
  - counterexamples: illustration (Venn-diagrams) & early backtracking
  - more & better agents; more case studies
- Long-term goals
  - (partial) decentralisation
  - dynamic clustering
  - communication bottleneck
  - agent interlingua
  - or-parallelism
  - integration with proof planning
  - critical (reflecting) agents