A Planning-Based Knowledge Acquisition Methodology

Eder Mateus Nunes Gonçalves and Guilherme Bittencourt

[eder|gb]@das.ufsc.br

University Federal of Santa Catarina - UFSC
Automation and Systems Department - DAS
Florinanópolis - Brazil
Topics

- Problem: the acquisition knowledge process building in hierarchical knowledge-based systems.
Topics

- Problem: the acquisition knowledge process building in hierarchical knowledge-based systems.
- Methodology: knowledge base description using Hierarchical Petri Nets and Planning
Topics

• Problem: the acquisition knowledge process building in hierarchical knowledge-based systems.

• Methodology: knowledge base description using Hierarchical Petri Nets and Planning

• Application: Soccer robots in Soccerserver.
Topics

- **Problem**: the acquisition knowledge process building in hierarchical knowledge-based systems.
- **Methodology**: knowledge base description using Hierarchical Petri Nets and Planning
- **Application**: Soccer robots in Soccerserver.
- **Presentation Topics**:
  - Introduction
  - Knowledge Acquisition as a Planning Problem
  - Plan Representation using Petri Nets
  - Petri Net Representation using Expert Systems
  - Conclusion
Introduction

- Problem: a methodology for acquisition knowledge process in cognitive multi-agent systems applicable in distributed domain.
Introduction

- Problem: a methodology for acquisition knowledge process in cognitive multi-agent systems applicable in distributed domain.

- Constraints:
Introduction

• Problem: a methodology for acquisition knowledge process in cognitive multi-agent systems applicable in distributed domain.

• Constraints:
  • The problem complexity can be split in a hierarchical set of plans.
Introduction

- Problem: a methodology for acquisition knowledge process in cognitive multi-agent systems applicable in distributed domain.

- Constraints:
  - The problem complexity can be split in a hierarchical set of plans.
  - The plans set can be implemented in one or more Expert Systems.
Introduction

- Problem: a methodology for acquisition knowledge process in cognitive multi-agent systems applicable in distributed domain.

- Constraints:
  - The problem complexity can be split in a hierarchical set of plans.
  - The plans set can be implemented in one or more Expert Systems.
  - Even the expert systems can be split in several knowledge bases according the different intelligence aspects (coordination, classification, etc...).
Introduction (continuation)

- Problem relevance:
Problem relevance:

- The acquisition knowledge difficulties plus the complexity in design a cognitive multi-agent systems.
  → *Intelligent coordination between agents.*
Introduction (continuation)

• Problem relevance:
  • The acquisition knowledge difficulties plus the complexity in design a cognitive multi-agent systems.
    → *Intelligent coordination between agents.*
  • Communication problem between the Knowledge Engineer and the Expert.
    → *There is no common vocabulary.*
Introduction (continuation)

- Propose approach: Petri Nets formalism as a problem representation tool:
Propose approach: Petri Nets formalism as a problem representation tool:

- The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
Propose approach: Petri Nets formalism as a problem representation tool:

- The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
- The level is solved as a planning problem;
Propose approach: Petri Nets formalism as a problem representation tool:

- The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
- The level is solved as a planning problem;
- The planning is express in a ordinary Petri Net, works like a communication language between the engineer and the expert;
Propose approach: Petri Nets formalism as a problem representation tool:

- The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
- The level is solved as a planning problem;
- The planning is expressed in an ordinary Petri Net, works like a communication language between the engineer and the expert;
- With a deal, the ordinary model is translated into a High Level Petri Net that uses knowledge bases as tokens;
Propose approach: Petri Nets formalism as a problem representation tool:

- The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
- The level is solved as a planning problem;
- The planning is expressed in an ordinary Petri Net, working like a communication language between the engineer and the expert;
- With a deal, the ordinary model is translated into a High Level Petri Net that uses knowledge bases as tokens;
- If necessary, a new hierarchy level is created, and the process is repeated;
Introduction (continuation)

- Propose approach: Petri Nets formalism as a problem representation tool:
  - The Knowledge Engineer and the Expert, for each abstract level, determine the goals;
  - The level is solved as a planning problem;
  - The planning is express in a ordinary Petri Net, works like a communication language between the engineer and the expert;
  - With a deal, the ordinary model is translated in a High Level Petri Net that uses knowledge bases as tokens;
  - If necessary, a new hierarchy level is created, and the process is repeated;
  - In the end, the high level Petri nets resulting is transformed in knowledge bases.
Knowledge Acquisition as a Planning Problem

- Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.
Knowledge Acquisition as a Planning Problem

- Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.

- A *Plan* is a sequence of actions $\{a_1, \ldots, a_n\} \in A$ that leads $s_0$ to $f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) \in S_G$. 

Knowledge Acquisition as a Planning Problem

• Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.

• A *Plan* is a sequence of actions \( \{a_1, \ldots, a_n\} \in A \) that leads \( s_0 \) to

\[
f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) \in S_G.
\]

• Example: a simple multi-agent strategy in robot soccer.
Knowledge Acquisition as a Planning Problem

- Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.

- A Plan is a sequence of actions \{a_1, \ldots, a_n\} ∈ A that leads \(s_0\) to
  
  \[f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) ∈ S_G.\]

  
  - \(S = \{\text{not – ball – control, ball – control, attacking, goal}\}\)
Knowledge Acquisition as a Planning Problem

- Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.

- A *Plan* is a sequence of actions \( \{a_1, \ldots, a_n\} \in A \) that leads \( s_0 \) to

\[
f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) \in S_G.
\]

  - \( S = \{\text{not \ – ball \ – control}, \text{ball \ – control}, \text{attacking}, \text{goal}\} \)
  - \( s_0 = \text{not \ – ball \ – control} \)
Knowledge Acquisition as a Planning Problem

- Hypothesis: The rules of the expert systems can be elicited more easily if they are seen as actions in plans in a social context.

- A *Plan* is a sequence of actions \( \{a_1, \ldots, a_n\} \in A \) that leads \( s_0 \) to \( f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) \in S_G \).

  - \( S = \{\text{not} - \text{ball} - \text{control}, \text{ball} - \text{control}, \text{attacking}, \text{goal}\} \)
  - \( s_0 = \text{not} - \text{ball} - \text{control} \)
  - \( S_G = \{\text{goal}\} \)
Knowledge Acquisition as a Planning Problem

• Hypothesis: The rules of the expert systems can be elicit more easily if they are seen as actions in plans in a social context.

• A Plan is a sequence of actions \(\{a_1, \ldots, a_n\} \in A\) that leads \(s_0\) to \(f(\ldots f(f(s_0, a_1), a_2), \ldots, a_n) \in S_G\).

• Example: a simple multi-agent strategy in robot soccer.
  * \(S = \{\text{not ball control, ball control, attacking, goal}\}\)
  * \(s_0 = \text{not ball control}\)
  * \(S_G = \{\text{goal}\}\)
  * \(A = \{\text{get ball control(-1), loose ball control(1), right attack(0), center attack(0), left attack(0), kick to goal(-1)}\}\)
Solving the problem we find the following plans:

\[ \langle -1, \{ (\text{get-ball-control}, \neg \text{ball-control}), (\text{right-attack}, \text{ball-control}), (\text{kick-to-goal}, \text{attacking}) \} \rangle \]

\[ \langle -1, \{ (\text{get-ball-control}, \neg \text{ball-control}), (\text{center-attack}, \text{ball-control}), (\text{kick-to-goal}, \text{attacking}) \} \rangle \]

\[ \langle -1, \{ (\text{get-ball-control}, \neg \text{ball-control}), (\text{left-attack}, \text{ball-control}), (\text{kick-to-goal}, \text{attacking}) \} \rangle \]

\[ \langle 0, \{ (\text{get-ball-control}, \neg \text{ball-control}), (\text{right-attack}, \text{ball-control}), (\text{loose-ball-control}, \text{attacking}) \} \rangle \]
Plan Representation Using Petri Nets

- Plans as Petri Nets:
Plan Representation Using Petri Nets

- Plans as Petri Nets:
  - $P = S$ the set of places correspond to the set of states in one abstraction level.
  - $T = A$ the set of transitions correspond to the set of actions in one abstraction level.
  - $\text{Pre}(p; t) = w$ if $p = a_2(s)$; $t = s$ such that $f(s; a) = s_0$ with $\text{Post}(s_0; a) = 1$ and $w$ is the weight of the minimum weight plan $= h$ in which $(s; a) \subseteq$. 
  - $\text{Post}(p; t) = 1$ if $p = a; t = s$ such that $f(s_0; a) = s$ with $\text{Pre}(s_0; a) = w$.
Plan Representation Using Petri Nets

- Plans as Petri Nets:
  - $P = S$ the set of places correspond to the set of states in one abstraction level.
  - $T = A$ the set of transitions correspond to the set of actions in one abstraction level.
Plan Representation Using Petri Nets

- Plans as Petri Nets:
  - \( P = S \) the set of places correspond to the set of states in one abstraction level.
  - \( T = A \) the set of transitions correspond to the set of actions in one abstraction level.
  - \( \text{Pre}(p, t) = w \), if \( p = a \in \alpha(s) \), \( t = s \) such that \( f(s, a) = s' \) with \( \text{Post}(s', a) = 1 \) and \( w \) is the weight of the minimum weight plan \( \pi = \langle w, \sigma \rangle \) in which \( (s, a) \in \sigma \).
Plan Representation Using Petri Nets

- Plans as Petri Nets:
  - $P = S$ the set of places correspond to the set of states in one abstraction level.
  - $T = A$ the set of transitions correspond to the set of actions in one abstraction level.
  - $Pre(p, t) = w$, if $p = a \in \alpha(s), t = s$ such that $f(s, a) = s'$ with $Post(s', a) = 1$ and $w$ is the weight of the minimum weight plan $\pi = \langle w, \sigma \rangle$ in which $(s, a) \in \sigma$.
  - $Post(p, t) = 1$, if $p = a, t = s$ such that $f(s', a) = s$ with $Pre(s', a) = w$. 
Plan Representation Using Petri Nets (cont.)

- Net example:
Plan Representation Using Petri Nets (cont.)

- In an Expert Systems, there are two ways to access a knowledge base:
Plan Representation Using Petri Nets (cont.)

- In a Expert Systems, there are two ways to access a knowledge base:
  - $\text{Tell} : KB \times T^* \rightarrow KB$
    to include new facts;
Plan Representation Using Petri Nets (cont.)

- In a Expert Systems, there are two ways to access a knowledge base:
  - $Tell : KB \times T^* \rightarrow KB$
    to include new facts;
  - $Ask : KB \times T \rightarrow S$
    to query to knowledge base.
Plan Representation Using Petri Nets (cont.)

- In a Expert Systems, there are two ways to access a knowledge base:
  - $\text{Tell} : \mathcal{KB} \times \mathcal{T}^* \rightarrow \mathcal{KB}$
    - to include new facts;
  - $\text{Ask} : \mathcal{KB} \times \mathcal{T} \rightarrow S$
    - to query to knowledge base.

- The Petri net elements are articulated in the sense to represents these expert systems components:
Plan Representation Using Petri Nets (cont.)

• In a Expert Systems, there are two ways to access a knowledge base:
  • $Tell : KB \times T^* \rightarrow KB$
    to include new facts;
  • $Ask : KB \times T \rightarrow S$
    to query to knowledge base.

• The Petri net elements are articulated in the sense to represents these expert systems components:
  • The *token* is defined as an element of the set $KB$, i.e., represents a knowledge base;
In an Expert Systems, there are two ways to access a knowledge base:

- **Tell**: $KB \times T^* \rightarrow KB$
  to include new facts;
- **Ask**: $KB \times T \rightarrow S$
  to query to knowledge base.

The Petri net elements are articulated in the sense to represent these expert systems components:

- The *token* is defined as an element of the set $KB$, i.e., represents a knowledge base;
- The transitions have preconditions and effects, that are specified like:
  
  $Cond : T \times KB \rightarrow S$
  
  $Act : T \times KB \times S \rightarrow KB$
Plan Representation Using Petri Nets (cont.)

- The previous example in ordinary model can be represented by a high level Petri net like this:
Petri Net Representation Using Expert Systems

- From the previous high level model, we can extract the following rule, referent to transition $t_4$.

(rule_004
  (if  (logic (global_goal current rws_attack_play))
       (logic (global_goal status active))
       (logic (local_goal current ?lg1))
       (logic (local_goal status ?lg2)))
  (filter (= ?lg1 kick)
            (= ?lg2 sucess))
  (then  (logic (global_goal status sucess))))
Conclusions

- The main advantages in the use of Petri nets are:
Conclusions

- The main advantages in the use of Petri nets are:
  - it is possible to describe a partial order between events;
Conclusions

- The main advantages in the use of Petri nets are:
  - it is possible to describe a partial order between events;
  - the system states, as the events, can be represented explicitly;
Conclusions

- The main advantages in the use of Petri nets are:
  - it is possible to describe a partial order between events;
  - the system states, as the events, can be represented explicitly;
  - Petri nets provide a tool family to specification, model, analysis, evaluation, and implementation;
Conclusions

• The main advantages in the use of Petri nets are:
  • it is possible to describe a partial order between events;
  • the system states, as the events, can be represented explicitly;
  • Petri nets provide a tool family to specification, model, analysis, evaluation, and implementation;
  • a precise and formal description of the agent synchronization in the environment is possible;
Conclusions

- The main advantages in the use of Petri nets are:
  - it is possible to describe a partial order between events;
  - the system states, as the events, can be represented explicitly;
  - Petri nets provide a tool family to specification, model, analysis, evaluation, and implementation;
  - a precise and formal description of the agent synchronization in the environment is possible;
  - the description of the control flow in expert systems;
Conclusions

- The main advantages in the use of Petri nets are:
  - it is possible to describe a partial order between events;
  - the system states, as the events, can be represented explicitly;
  - Petri nets provide a tool family to specification, model, analysis, evaluation, and implementation;
  - a precise and formal description of the agent synchronization in the environment is possible;
  - the description of the control flow in expert systems;
  - a common language between the Knowledge Engineer and the Expert in the Expert System development;
Conclusions (cont.)

- Future works include:
Conclusions (cont.)

- Future works include:
  - The tool formalization to system simulation, analysis and validation;
Conclusions (cont.)

- Future works include:
  - The tool formalization to system simulation, analysis and validation;
  - Formal coordination between the social and individuals levels using the same methodology.
Conclusions (cont.)

- Future works include:
  - The tool formalization to system simulation, analysis and validation;
  - Formal coordination between the social and individuals levels using the same methodology.

Thanks for your attention!