Designing Expert Systems

10. Design Principles for Learning Assistants

Gheorghe Tecuci
tecuci@gmu.edu
http://lac.gmu.edu/

Learning Agents Center
and Computer Science Department
George Mason University
Agent Development Approaches

Without Learning Assistant

Model the reasoning of SME
Create object ontology
Define reasoning rules
Verify and update rules

Instruct SME to explicitate reasoning
Import and develop initial ontology
Provide and explain solutions
Analyze Agent’s solutions

Develop reasoning trees
Specify instances and features
Learn ontological elements
Learn reasoning rules
Explain errors
Refine rules

With Learning Assistant

With Learning Assistant

KE
SME
Agent
KE
SME
Agent
SME
Agent
SME
Agent
SME
Agent
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
P1. Spiral Development with SMEs and End-Users

Intelligence analysis, Center of gravity determination, Course of action critiquing, Emergency response planning, Workaround reasoning, PhD advisor selection, Teaching higher order thinking skills.

Development of systematic approach to expert problem solving

Working closely with subject matter experts to model their reasoning

Development of the Disciple theory for agent teaching by non-computer experts

Integration of many areas of Artificial Intelligence

Army War College
Air War College
George Mason University

Development and application of Disciple agents

Working closely with end users to receive crucial and timely feedback

DISCIPLE
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Develop a general problem solving paradigm which is:
○ natural for a human user;
○ appropriate for an automated agent.

"I Keep Six Honest..."
I keep six honest serving-men
(They taught me all I knew);
Their names are What and Why and When
And How and Where and Who.

Rudyard Kipling
“One of the most highly developed skills in contemporary Western civilization is dissection; the split-up of problems into their smallest possible components. We are good at it. So good, we often forget to put the pieces back together again.”

The reduction representation of a class of problems is a quadruple \((P, S, RO, OS)\) where:

- \(P\) - class of problems;
- \(S\) - solutions of problems;
- \(RO\) - reduction operators, each reducing a problem to sub-problems and/or solutions,
- \(SO\) - synthesis operators, each synthesizing the solution of a problem from the solutions of its sub-problems.

A problem \(P_1\) is solved by:

- successively reducing it to simpler problems through the application of the reduction operators;
- finding the solutions of the simplest problems;
- successively combining these solutions through the application of synthesis operators until the solution of the initial problem is obtained.
Sample Agent: Analyst’s Cognitive Assistant

Analytic Assistance
Empowers the analysts through mixed-initiative reasoning for hypotheses analysis, collaboration with other analysts and experts, and sharing of information.

Learning
Rapid acquisition and maintenance of subject matter expertise in intelligence analysis which currently takes years to establish, is lost when experts separate from service, and is costly to replace.

Tutoring
Helps new intelligence analysts learn the reasoning processes involved in making intelligence judgments and solving intelligence analysis problems.
Hypothesis Analysis through Problem Reduction

1) A complex hypothesis analysis problem is successively reduced to simpler problems that either have known solutions or can be solved through evidence analysis.

2) Potentially relevant pieces of evidence for the unsolved problems are identified.

3) The pieces of evidence are analyzed to obtain solutions for the unsolved problems.

4) The solutions of the simplest problems are successively combined to obtain the solution of the initial problem.

Assess whether Al Qaeda has nuclear weapons.

It is likely that Al Qaeda has nuclear weapons.

<table>
<thead>
<tr>
<th>Remote</th>
<th>Unlikely</th>
<th>Even chance</th>
<th>Probably, Likely</th>
<th>Almost certainly</th>
</tr>
</thead>
<tbody>
<tr>
<td>National Intelligence Council’s standard estimative language</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Demo: Analytic Assistance

Disciple-LTA Demo: Analytic Assistance

collaborate

Disciple Agent

KB
Problem Reduction: Abstract and Detailed Reasoning

Main problems

Reduction of a main problem to its main subproblems

Abstract tree

Detailed tree
Problem Reduction and Solution Synthesis

Detailed evidence and source analysis

EVD-Dawn-Mir-01-01c

The following interpretation:
Al Qaeda has nuclear weapons and may use them to defend itself of the following statement by Osama bin Laden:
We [Al Qaeda] have chemical and nuclear weapons [...] We ourselves are the target of killings, destruction and atrocities. We are only defending ourselves. This is defensive Jihad. We want to defend our people and our land. That is why I say that if we dont get security, the Americans, too would not get security.

from the document EVD-Dawn-Mir01:
Mir, H. (2001, November 10). Osama claims he has nukes: If US uses N-arms it will get same response. Dawn - Pakistan English Newspaper
Disciple-LTA makes very clear:
The analysis logic; What evidence was used and how; What assumptions have been made; What is not known.
Assumptions-based Analysis

Disciple-LTA allows for: Assumptions checking; Rapid updating of large analysis trees based on new intelligence data and assumptions.

1. Analyst's assumption challenged by Disciple-LTA

2. Revised, assumption-based, solution

Over 1700 reasoning nodes
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
P3. Multi-Agent and Multi-Domain Problem Solving

Develop a general problem solving paradigm that facilitates:
○ collaboration between users assisted by their agents;
○ solving problems requiring multi-domain expertise.
**Problem:** Assess whether Al Qaeda has nuclear weapons.

**Subproblem:** Assess whether Al Qaeda makes credible claims to have nuclear weapons.

**Solution:** It is almost certain that the Al Qaeda claims of having nuclear weapons are credible.
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Structure the knowledge base into two parts:
- o its more general and reusable components;
- o its more specific components.

Disciple: Knowledge Base Structuring

- The object ontology which may be reused from existing knowledge repositories;
- The problem solving rules which are learned from the subject matter expert.
Knowledge Base = Object Ontology + Rules

The ontology is a hierarchical description of the domain objects.

Interpretation: Al Qaeda has chemical and nuclear weapons as deterrent.
Knowledge Base = Ontology + Rules

Rules specify general problem reduction or solution synthesis steps

Analysis Tree

To assess whether there are states that may be willing to sell nuclear weapons to an actor, one has to consider each nuclear state and assess whether that state may be willing to sell nuclear weapons to that actor, except for the case in which the nuclear state is an enemy of that actor and also except for the case when the nuclear state opposes the proliferation of nuclear weapons.
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Control of Modeling, Learning and Problem Solving

- Problem
- Mixed-Initiative Problem Solving
- Reasoning Tree
- Ontology + Rules

- Accept Reasoning Step
  - Explain Examples
  - Refined Rules
  - Refined Ontology

- Reject Reasoning Steps
  - Explain Examples
  - Rules Refinement

- Extend Reasoning Tree
  - Explain Examples

- Rules Learning
Develop agent teaching and learning methods where the subject matter expert helps the agent to learn (e.g. by giving examples, hints and explanations), and the agent helps the expert to teach it (e.g. by asking relevant questions).
Integrated Teaching and Learning in Disciple

Analogy and Hint Guided Explanation

Example of problem reduction step

Analogy-based Generalization

Rule with Plausible Version Space Condition

Incomplete explanation

Knowledge Base

Example of problem reduction step
Reasoning Rules Learned from Analyst’s Solution

1. The analyst extends the analysis logic

Assess whether there are states with nuclear weapons that may be willing to sell nuclear weapons to Al Qaeda.

Which is a nuclear state?
North Korea

Assess whether North Korea may be willing to sell nuclear weapons to Al Qaeda.

What might be a possible reason for North Korea to sell nuclear weapons to Al Qaeda?
United States is perceived as a common enemy of North Korea and Al Qaeda.

Assess to what extent the perception that United States is a common enemy of North Korea and Al Qaeda might be a good reason for North Korea to sell nuclear weapons to Al Qaeda.

2. Disciple learns reasoning rules

To assess whether there are states that may be willing to sell nuclear weapons to an actor, one has to consider each nuclear state and assess whether that state may be willing to sell nuclear weapons to that actor.
## Overview: Design Principles

<table>
<thead>
<tr>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spiral Development with SMEs and End-Users</td>
</tr>
<tr>
<td>Problem Solving Paradigm for User-Agent Collaboration</td>
</tr>
<tr>
<td>Multi-Agent and Multi-Domain Problem Solving</td>
</tr>
<tr>
<td>Knowledge Base Structuring for Knowledge Reuse</td>
</tr>
<tr>
<td>Integrated Teaching and Learning</td>
</tr>
<tr>
<td>Multistrategy Learning</td>
</tr>
<tr>
<td>Learning with an Evolving Representation Space</td>
</tr>
<tr>
<td>Mixed-Initiative Modeling, Learning and Problem Solving</td>
</tr>
<tr>
<td>Plausible Reasoning with Partially Learned Knowledge</td>
</tr>
<tr>
<td>User Tutoring in Problem Solving</td>
</tr>
<tr>
<td>Agent Architecture for Generality-Power Tradeoff</td>
</tr>
</tbody>
</table>
P6. Multistrategy Learning

Develop multistrategy learning methods that integrate complementary learning strategies to take advantage of their strengths to compensate for each other’s weaknesses.

Knowledge Base

Learning by Analogy and Experimentation

Failure explanation

Learning from Explanations

Example of problem reductions generated by the agent

Incorrect example
Correct example

Learning from Examples

IF we have to solve
<Problem>

Main
PVS Condition

Except-When
PVS Condition

THEN solve
<Subproblem 1>
...
<Subproblem m>
France will not sell nuclear weapons to Al Qaeda because it perceives it as an enemy.

To assess whether there are states that may be willing to sell nuclear weapons to an actor, one has to consider each nuclear state and assess whether that state may be willing to sell nuclear weapons to that actor, except for the case in which the nuclear state is an enemy of that actor.
Rules Refined based on Analyst’s Critique

1. Disciple applies the refined rule

Assess whether there are states with nuclear weapons willing to sell nuclear weapons to Al Qaeda.

Which is a nuclear state which is not an enemy of Al Qaeda?
North Korea
Assess whether North Korea is willing to sell nuclear weapons to Al Qaeda.

Which is a nuclear state which is not an enemy of Al Qaeda?
Iran
Assess whether Iran is willing to sell nuclear weapons to Al Qaeda.

Which is a nuclear state which is not an enemy of Al Qaeda?
Pakistan
Assess whether Pakistan is willing to sell nuclear weapons to Al Qaeda.

Which is a nuclear state which is not an enemy of Al Qaeda?
Russia
Assess whether Russia is willing to sell nuclear weapons to Al Qaeda.

2. The analyst critiques the reasoning

This is wrong!
Russia will not sell nuclear weapons to Al Qaeda because it opposes the proliferation of nuclear weapons.

To assess whether there are states that may be willing to sell nuclear weapons to an actor, one has to consider each nuclear state and assess whether that state may be willing to sell nuclear weapons to that actor, except for the case in which the nuclear state is an enemy of that actor and also except for the case when the nuclear state opposes the proliferation of nuclear weapons.

3. Disciple refines the rule with a new except-when condition

Refined Rule

DECOMPOSITION RULE DDR_0025 FORMAL DESCRIPTION

IF:
Assess whether there are states with nuclear weapons willing to sell nuclear weapons to ?O1

Q:
Which is a nuclear state which is not an enemy of ?O1?

A:

MAIN CONDITION

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O1 (terrorist group)</td>
<td>(actor)</td>
<td></td>
</tr>
<tr>
<td>?O2 (nuclear state)</td>
<td>(actor)</td>
<td></td>
</tr>
</tbody>
</table>

EXCEPT WHEN CONDITION 1

<table>
<thead>
<tr>
<th>Var</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O2 (nuclear state)</td>
<td>(actor)</td>
<td></td>
</tr>
<tr>
<td>?O1 (terrorist group)</td>
<td>(actor)</td>
<td></td>
</tr>
</tbody>
</table>

EXCEPT WHEN CONDITION 2

<table>
<thead>
<tr>
<th>Var</th>
<th>Relationship</th>
<th>Var</th>
</tr>
</thead>
<tbody>
<tr>
<td>?O2</td>
<td>perceives as enemy</td>
<td>?O1</td>
</tr>
</tbody>
</table>

Assess whether ?O2 is willing to sell nuclear weapons to ?O1.
Disciple-LTA Demo:
Solving, Modeling, and Learning
Discussion: Rule Refinement with Negative Example

Rule Condition C

- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)

- Specialize Upper Bound of Main Cond (C, Ex)
- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)

- Generalize Lower Bound of Except When Cond (C, Ex)
- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)

- Generalize Lower Bound of Except When Condition (C, Ex)
- Specialize Upper Bound of Main Cond (C, Ex)
- Learn Except When Condition (C, Ex)
- Keep as Negative Exception (C, Ex)
<table>
<thead>
<tr>
<th>Overview: Design Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spiral Development with SMEs and End-Users</td>
</tr>
<tr>
<td>Problem Solving Paradigm for User-Agent Collaboration</td>
</tr>
<tr>
<td>Multi-Agent and Multi-Domain Problem Solving</td>
</tr>
<tr>
<td>Knowledge Base Structuring for Knowledge Reuse</td>
</tr>
<tr>
<td>Integrated Teaching and Learning</td>
</tr>
<tr>
<td>Multistrategy Learning</td>
</tr>
<tr>
<td>Learning with an Evolving Representation Space</td>
</tr>
<tr>
<td>Mixed-Initiative Modeling, Learning and Problem Solving</td>
</tr>
<tr>
<td>Plausible Reasoning with Partially Learned Knowledge</td>
</tr>
<tr>
<td>User Tutoring in Problem Solving</td>
</tr>
<tr>
<td>Agent Architecture for Generality-Power Tradeoff</td>
</tr>
</tbody>
</table>
Develop methods that allow continuous adaptation of the previously learned rules to the evolution of the ontology.

IF <Problem>

PVS Condition

Except-When PVS Condition

THEN <Subproblem 1> ...
    <Subproblem m>

PVS Condition

Except-When PVS Condition

Develop methods that allow continuous adaptation of the previously learned rules to the evolution of the ontology.
Characterization of the Disciple Learning Methods

Uses the explanation of the first positive example to generate a much smaller version space than the classical version space method.

Conducts an efficient heuristic search of the version space, guided by explanations, and by the maintenance of a single upper bound condition and a single lower bound condition.

Will always learn a rule, even in the presence of exceptions.

Learns from a few examples and an incomplete knowledge base.

Uses a form of multistrategy learning that synergistically integrates learning from examples, learning from explanations, and learning by analogy, to compensate for the incomplete knowledge.

Uses mixed-initiative reasoning to involve the expert in the learning process.

Is applicable to complex real-world domains, being able to learn within a complex and evolving representation language.
Develop mixed-initiative methods where modeling, learning, and problem solving mutually support each other.
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Develop reasoning methods based on confidence levels that allow efficient use of partially learned rules for modeling expert’s reasoning, learning and problem solving.

**P9. Plausible Reasoning with Partially Learned Knowledge**

- Rule’s conclusion is (most likely) incorrect
- Rule’s conclusion is (most likely) correct
- Rule’s conclusion is plausible
- Rule’s conclusion is (most likely) incorrect
- Rule’s conclusion is not plausible

**PVS Condition**

**Except-When PVS Condition**
Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Develop approaches to user tutoring that allow the agent to easily and rapidly teach the user its problem solving paradigm, to facilitate their collaboration.

Disciple-LTA Demo: Tutoring
Lesson Fragment: Hypothesis support from a piece of evidence

Access to what extent the piece of evidence favors the hypothesis, assuming that we believe the information provided by the piece of evidence.

Access to what extent the piece of evidence favors the hypothesis.

The information provided by the piece of evidence and the extent to which it is believable.

The piece of evidence is testimonial evidence obtained at second hand.

Access the believability the reporter of the piece of evidence.

Access the believability the source of the piece of evidence.

Lesson on Evidence

Abstract reduction strategy

Automatically generated illustration of the abstract strategy
Lesson Fragment: Hypothesis support from a piece of evidence

Abstract synthesis strategy

Automatically generated illustration of the abstract strategy
### Overview: Design Principles

- Spiral Development with SMEs and End-Users
- Problem Solving Paradigm for User-Agent Collaboration
- Multi-Agent and Multi-Domain Problem Solving
- Knowledge Base Structuring for Knowledge Reuse
- Integrated Teaching and Learning
- Multistrategy Learning
- Learning with an Evolving Representation Space
- Mixed-Initiative Modeling, Learning and Problem Solving
- Plausible Reasoning with Partially Learned Knowledge
- User Tutoring in Problem Solving
- Agent Architecture for Generality-Power Tradeoff
Structure the architecture of the agent into two parts:
- a reusable domain-independent learning agent shell;
- domain specific modules.

Disciple Agent

Disciple Learning Agent Shell

- Graphical User Interface
- Learner
- Knowledge Base Manager
- Problem Solver

Domain Independent Modules

- Customized User Interface
- Customized Problem Solver

Domain Dependent Modules

Knowledge Repository

Knowledge Base
Overall Architecture of the Disciple Shell

Ontology Elicitation, Learning and Refinement
- Ontology Viewers and Editors
- Ontology Graphical Browsers
- Scenario Elicitation, Script Editor
- Ontology Learning and Refining

Disciple Learning Agent Shell
- Mixed-initiative, Multi-agent Framework
- Multi-Agent Framework
- Mixed-Initiative Reasoner
- Task Agenda Modules
- Interaction Model Learning and Refining

Problem Solving
- Problem Solving Modules
- Assumptions Modules

Knowledge Management, Verification and Validation
- Knowledge Management Module
- System Verification Modules
- Knowledge Base Validation Modules

Rule Learning and Refinement
- Task and Rule Learning Modules
- Plausible Explanation Generation Modules
- Rule Refinement Modules
- Rule Analysis Modules
- Control Wizards for Rule Refinement

Knowledge Repository Management
- Management of Distributed Knowledge Repository
- Knowledge Base Versioning

Knowledge Integration, Import, and Export
- Knowledge Integration Tools
- Import Tools
- Export Tools

Knowledge Base Validation Modules
Customization of the Disciple Shell

Customization for each type of user

Subject Matter Expert
Field Application
Education and training

Disciple COG

Ontology Elicitation, Learning and Refinement
- Ontology Viewers and Editors
- Ontology Graphical Browsers
- Scenario Elicitation, Script Editor
- Ontology Learning and Refining

Disciple Learning Agent Shell
- Mixed-initiative, Multi-agent Framework
- Problem Solving Modules
- Assumptions Modules

Rule Learning and Refinement
- Task and Rule Learning Modules
- Plausible Explanation Generation Modules
- Rule Refinement Modules
- Rule Analysis Modules
- Control Wizards for Rule Refinement

Problem Solving
- Problem Solving Modules
- Mixed-Initiative Reasoner
- Task Agenda Modules
- Interaction Model Learning and Refining

Knowledge Management, Verification and Validation
- Knowledge Management Module
- System Verification Modules
- Knowledge Base Validation Modules

Knowledge Repository Management
- Management of Distributed Knowledge Repository
- Knowledge Base Versioning

Customization for each domain

COG Report Generator
Dedicated modules

System knowledge base
Domain knowledge base

Knowledge Integration, Import, and Export
- Knowledge Integration Tools
- Import Tools
- Export Tools

Knowledge Base Versioning
Conclusion: Research Vision for Learning Assistants

Mainframe Computers

Personal Computers

Learning Assistants

Disciple

Global Grid Computing and Communication Infrastructure