

Domain Based Problem Solving Process

Teodor Rus
The University of Iowa
Iowa City, IA 52242, USA

December 30, 2012

Evolution of Computer Based Problem Solving Process

This paper discusses the evolution of Computer Based Problem Solving Process in the framework provided by three parameters:

1. Mechanism used to execute the problem solving algorithm.
2. Major problems raised by algorithm execution.
3. Evolution mechanism that eliminates the problems raised by algorithm execution.

Major steps we identify are:

- ▶ Polya problem solving methodology.
- ▶ Computer based problem solving methodology, CBPSP.
- ▶ Web based problem solving methodology, WBPSP.
- ▶ Domain based problem solving methodology, DBPSP.

Polya problem solving methodology

Polya 4 step problem solving process:

1. Formulate the problem;
 2. Develop a solution algorithm;
 3. Execute the algorithm;
 4. Check the result.
- ▶ Algorithm Execution: problem solver executes the algorithm.
 - ▶ Major Problems: there are problems where algorithm complexity cannot be managed by hand; algorithm effectivity is hampered by execution speed.
 - ▶ Evolution Mechanism: create automatons that perform the algorithm execution. These are today computers!

Map the problem solving algorithm into a program and let the computer execute the program (*programming + program execution tools*).

- ▶ Algorithm Execution: computer executes the program, i.e., *problem and its solution algorithm are embedded into machine language*.
- ▶ Major problems: complexity of program development and execution.
- ▶ Evolution mechanism: develop problem and algorithm dedicated gadgets. **Example: cell-phone**

Note: this resolves programming problem but raises other problems:

- ▶ domain experts need to rely on computer experts;
- ▶ problem domains evolve indefinitely with human cognition while gadgets are finite/static objects.

Develop the Web and standardize problem solving algorithms as expressions of Web Services (WS):

1. Algorithm Execution:
 - ▶ Computer is a node into a computer network;
 - ▶ Develop standards that resolve problems raised by computer-interoperation. **Examples:** HTML, XML, WSDL, SOAP, UDDI;
 - ▶ Implement client-server paradigm of problem solving process.
2. Problems: this is adequate for computer experts but excludes non-experts from WS development process.
3. Evolution: domain based problem solving tools

Develop *Domain Algorithmic Languages* and let problem solving algorithms be DAL expressions developed by domain experts.

1. Algorithm Execution: domain expert using domain oracles (i.e., computer is embedded in DAL language).
 - ▶ DAL is specified by a domain ontology where concepts are associated with WS implementing them.
 - ▶ Oracles are Domain Dedicated Virtual Machines that perform DAL algorithms on the WEB;
2. Problems: how to resolve problems raised by the dynamic nature and ambiguity of natural language.
3. Evolution Mechanism: software tools that support three-way collaborations during problem solving process:
 - ▶ Domain expert: develop problem solving algorithms using DAL;
 - ▶ Computer expert: use computer languages to implement DAL ontologies and DAL oracles;
 - ▶ Web experts: develop tools that allow DAL algorithm execution on the Web.

NLD System is a *Proof of Concept* that represents a crude implementation of the DBPSP where:

- ▶ Problem domain: is the arithmetics.
- ▶ Domain specification: computational emancipation arithmetic.
- ▶ Implementation: (1) computationally emancipate a small number of domain concepts; and (2) automatic procedures that embed solution algorithms as *new computational emancipated domain concepts* into domain ontology.
- ▶ Problems: the experiment concerns a well formalized problem domain. Need to expand it to other problem domains.