A 2020 Computer User

(2020's Computer Based Problem Solving Process)

Teodor Rus and Cuong Bui
The University of Iowa,
Iowa City, IA 52242, USA

August 26, 2010

Alain Chesnais, ACM President

Are you ready for 2020?

Topics

- Computer Based Problem Solving Process;
- Web Based Problem Solving Process;
- Cloud Computing Supporting WBPSP;
- NLD System: a Cloud Model for WBPSP;
- Example NLD application.

Computer Based Problem Solving Process

Conditions:

- Problem Domain (PD) is a repository of universal knowledge!
- Computer is a universal problem solving tool!

Problem Solving Process

Assumption: problem solver has access to a computer C.

- Formulate the problem;
- Develop a solution algorithm;
- Program the algorithm, i.e. encode the problem and the algorithm into a Program ;
- Map Program into Machine Language Program (MLP) of C;
- Run computer C on MLP and its Data stored in C's memory;
- Retrieve the solution generated by the computer run.

Limitations

- Problem solver must be both:
  a problem domain expert (to develop the algorithm)
  and a computer expert (to develop MLP).
- If computer C changes, MLP changes, hence problem solver cannot reuse the program as a component of another program;
- Problem solver cannot reuse the algorithm as a component in the solution of another problem solving algorithm.

CBPSP does not support problem solver's knowledge evolution.

Technology Trends

Technology produced myriad of computer based gadgets such as phones, pads, keys, etc., all of which are characterized by:

- Each gadget provides solutions to a class of problems;
- Problems are identified by buttons labeled by NL terms;
- To solve a problem (example, make a phone call) the user clicks the button labeled by the appropriate term;
- No formal specification and/or programming is required.

Side-effect on Computer Education

Computer user looses interest in computer education:

- Universality of the computer as a problem solving tool is lost. Computer becomes a gadget dedicated to a problem;
- Computer users are interested in solving their problems by clicking buttons rather than by programming;
- Gadget design and use lacks the intellectual substance capable to refresh CS curricula;

What can we do to revive CS excitement?

Answer

1. We can observe that current CBPSP is dedicated to computer experts. (Computer user is a programmer!)
2. We can transform the trend into a science by setting the background for **non-computer expert dedicated technology**.

Problem Domain (PD) In 2020

In 2020, PD will be computationally emancipated, i.e.:

- PD will be specified by a Domain Ontology (DO);
- Concepts in DO are associated with Web Services that implement them;
- PD will be provided with a Domain Dedicated Virtual Machine (DDVM) which is an abstraction similar to a real-computer;
- Instructions of the DDVM are computer processes performing the computer artifacts that implement domain concepts.

DDVM Formally

$DDVM = \langle CC, Execute, Next \rangle$ where:

1. CC is a concept counter that runs on DO, similar to a computer Instruction Counter (IC) which runs on memory;
2. Execute is a device that executes the computer process associated with (CC), similar to the CPU of a real computer;
3. Next check concepts in DO, similar to the mechanism used by CPU to check the instructions it is asked to execute.

2020 Computer Based Problem Solving Process

Assumption: problem solver has access to the DDVM:

- Formulate the problem;
- Develop a solution algorithm;
- Input the algorithm to DDVM which performs:

```
1 CC = First Concept of the algorithm;
2 While(CC in not End of the algorithm)
        Execute(CC); CC := Next(CC);

3 If CC is End of the algorithm DDVM outputs the solution.
```

Contrast

2020 CBPSP becomes a service oriented business:

- Computer user does not write programs!
  Hence she is not required to be a computer expert.
- Problem solving is a DDVM-based business.
- The problem solver work is completely reusable; (algorithm developed by problem solver is executed by DDVM).
- Problem domain evolves with PSP (DO is expanded with concepts executed by DDVM). This mimics human-learning process.

Conclusion:

Computational Emancipation of Application Domain (CEAD) transforms the computer from a

number-crunching tool

into a

domain dedicated cognitive machine.

What is CEAD?

CEAD is a process similar to the process of domain formalization performed by mathematical applications:

- Domain formalization transforms domain concepts into mathematically well-defined abstractions;

  CEAD transforms domain concepts into computer artifacts;

- Domain formalization develops the "mathematical language of the domain" used by mathematicians to prove theorems.

  CEAD develops the "computational language of the domain" used by domain experts to express problems and algorithms.

Natural Language of the Domain

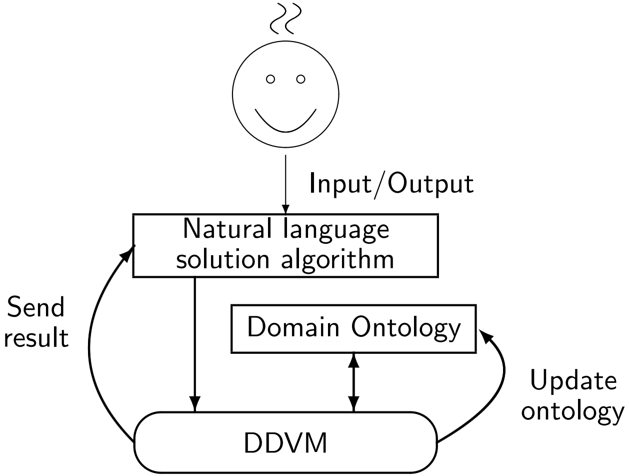
Computational language of a domain can be formalized thus providing a Natural (Human) Programming Language, further referred to as the Natural Language of the Domain, (NLD).

A 2020 user will use NLD and DDVM as follows:

- Develop the solution algorithm using the NLD;
- Input the algorithm and data to the DDVM;
- DDVM compute and return result.

Note: no programming as usual is needed.

Using DDVM


Input/Output
Natural language solution algorithm
Domain Ontology
Send result
Update ontology
DDVM

WBPSP

Web Based Problem Solving Process (WBPSP) is the process of using DDVM by AD experts to solve their problems without programming as usual.

Questions

1. Does this mean that programming as usual disappear?
2. What are the technological implications of WBPSP?
3. Who develop the DDVM for various domains and how?
4. Do we have examples of WBPSP-s?

Does programming as usual disappear?

By the contrary:

Programming as usual receives a new dimension with WBPSP!

WBPSP sets bases for a new mechanism of interdisciplinary collaboration which bridges the semantic gap between AD experts and IT experts!

Implications of WBPSP

WBPSP generates an unlimited and unrestricted domain of new Computer Technology:

- Each AD needs to be provided with its own DDVM.
- WBPSP makes the universality of conventional computer match the diversity of human universe of discourse!
- Problem solving by One-Pattern-Fits-All (computer programming) is no longer valid! Hence:
- WBPSP may resolve problems raised by software complexity!
- System Software receive new dimensions:
  dedication to problem domains,
  evolution with problem solving process, etc.

Who and where manages WBPSP?

WBPSP is managed by Cloud Computing!

Cloud Computing: is a wonderland populated by:
computer networks (clients and servers) and
people who develop and sell web services!

Web Services

A web service is a program (as usual) provided with three new mechanisms:

1. A WSDL expression that describes its functionality as a program run on a node in the network;
2. A SOAP expression that describes the mechanism of accessing the web service;
3. A UDDI registry that allow people interested to discover and integrate the web service in their applications.

Examples of WBPSP

- Service Oriented Architecture (SOA) is the best example of WBPSP!
- BPEL (Busines Process Expression Language) is another example;
- NLD System provides a model of cloud business that allows people to buy DDVM-s dedicated to their AD-s.

Note: with SOA and BPEL in addition to programming as usual, computer user needs to develop WSDL, SOAP, UDDI too!

What is NLD System?

NLD System consists of three components that are integrated and run in the cloud:

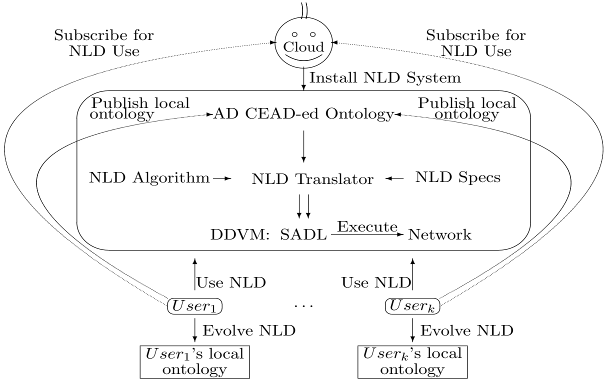
1. NLD, used by computer user to develop AD algorithms.
2. OWL file representing the CEAD-ed Domain Ontology.
3. A translator that map NLD algorithms into DDVM running them in the cloud.

Using NLD System

Computer users can use the NLD system by the following pattern:

1. Get a Cloud subscription (Subscribe2NLD) to use the NLD system. Cloud manager install NLD system on user laptop!
2. The user uses the NLD system (UseNLD) by developing and running NLD algorithms.
3. The user evolves the NLD with new concepts by Evolve NLD which generate a local OWL File for the new concepts;
4. The user can remove a concept from her ontology by Remove Concept ;
5. NLD users can share the ontology concepts they develop by publishing their local ontologies (PublishOWL).

NLD System Pictorial



Example: High School Algebra

Problem Domain: Solving equations.

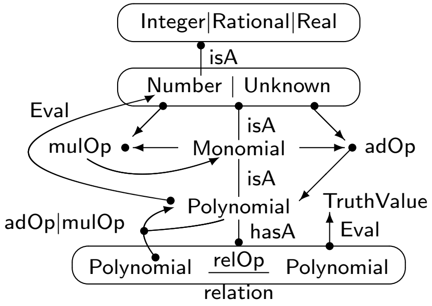
Domain concepts:

- Numbers (integers, real)
- Unknowns (variables denoted by letters)
- Monomials
- Polynomials
- Relations

Domain Ontology

A highgraph (D. Harel, Comm. ACM 31, #5) whose nodes represent concepts and edges represent conceptual relationships.

Assumption: concepts, properties, and actions used in the ontology are associated with web services implementing them.



Example: solving second-degree equations

- Problem formalization: $ax^2 + bx + c = 0$ where $a, b, c \in R$ and $a \neq 0$;
- Solution algorithms: $x_{1,2} = \frac{-b+|-\sqrt{b^2-4ac}}{2a}$;
- Solving equations:
  (1) input solution algorithm to DDVM;

  (2) call the DDVM and engage in the DDVM dialog.
- Evolving the domain:
  add equation and solver to the domain ontology.

Demos

1. The CEAD-ed Arithmetic Ontology;
2. Evolving Arithmetic Ontology by NLD reimplementation:
   adding new data (complex type)
   adding new web services (square root);
3. Using Arithmetic Ontology:
   equation solver (SolverR, SolverC);
4. Adding, using, removing concepts:
   add2Ont SolverR, add2Ont SolverC
   use SolverR, use SolverC, remove SolverR, remove SolverC
5. Evolving Arithmetic Ontology to a Vector space:
   adding abstractions to NLD (vector algebra)

Tools Used for NLD Implementation

- protégé (`http://protege.stanford.edu/`) is an Editor and Knowledge Acquisition System;
- OWL (web ontology language), for ontology development and RDF (Resource Description Framework) as the Ontology Reasoning Tool.
- Apache extensible interactive system (axis) is the Apache server used to implement software services interoperability.

Java Architecture for XML Binding (JAXB) is a newer and more convenient way to process XML content using Java objects by binding its XML schema to Java representation.