GSoC project in Tcl/Tk Community

A Business Rule Management System based on the high-level object oriented scripting language XOTcl

http://code.google.com/soc/2008/tcl/appinfo.html?csaid=2D63BF1F53E58EA1

http://wiki.tcl.tk/20832

Project Members

- **Mentor**
    - Univ.Prof. Dr. Gustaf Neumann
    - Institute for New Media (University of Business Administration, Vienna)
    - http://wi.wu-wien.ac.at/nm/en/neumann
    - Main Developer of XOTcl
- **Student**
    - Mag. Franz Wirl
    - http://wi.wu-wien.ac.at/nm/en/wirl
    - Email: franz.wirl@wu-wien.ac.at

Project Idea

High level object oriented scripting languages like XOTcl can be perfectly used to implement Charles Forgy's Rete algorithm. An algorithm that has been developed and tested to match between more than a thousand patterns and objects. Implementing this fast algorithm into/with XOTcl will provide a fast and dynamic Rete library in XOTcl.
An object oriented implementation allows a natural expression of rules. Object oriented interfaces to the algorithm improve its flexibility and allows usage in many different domains.

RETE

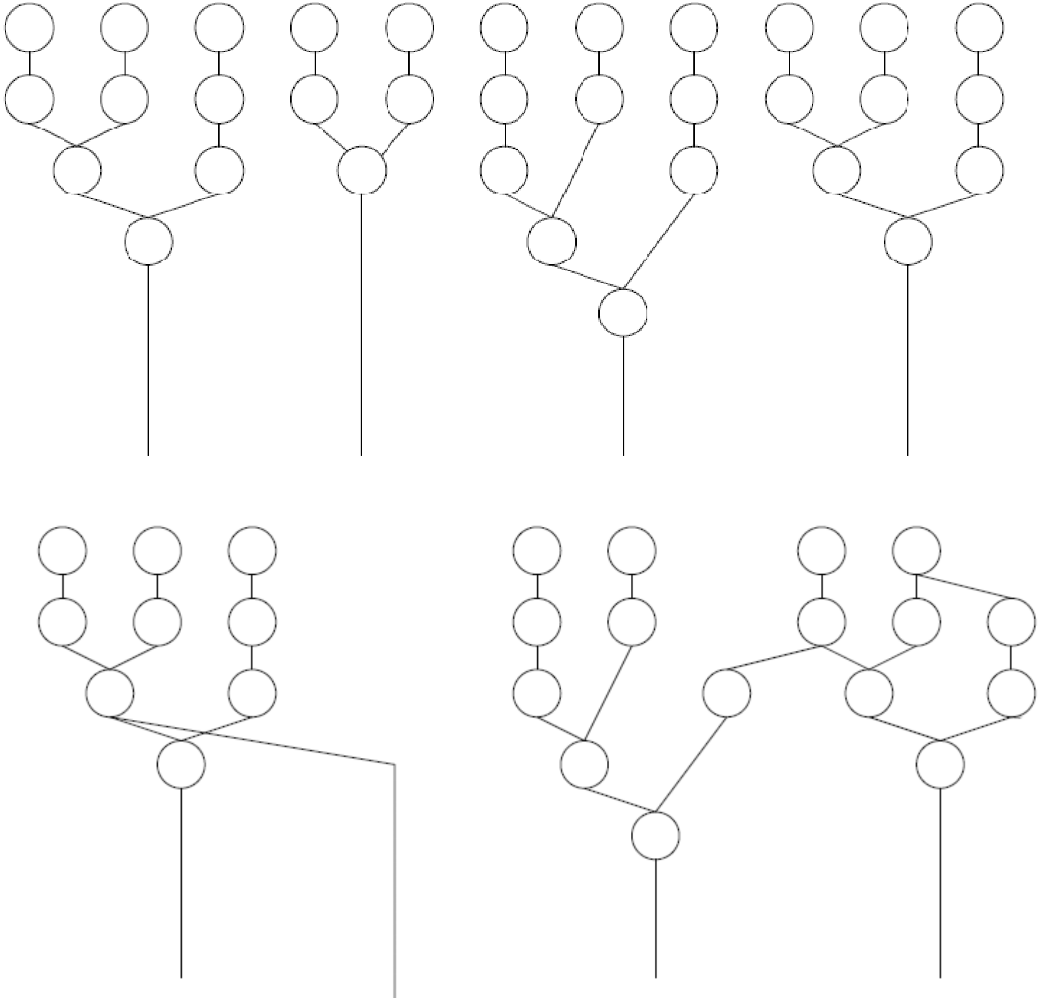
- **Rete /[net] A fast algorithm for the many pattern/many object pattern match problem**
- **By Charles Forgy**
- **Artificial Intelligence, Vol 19(1), Sept. 1982 http://dx.doi.org/10.1016/0004-3702(82)90020-0**
- **Usage of RETE**
  - DROOLS (JBoss Rules) [JAVA]
    http://www.jboss.org/drools/
  - CLIPS [C]
    Expert System NASA 1984
    http://clipsrules.sourceforge.net/

What is RETE

- RETE is an algorithm for production systems interpreters
- many pattern/many object pattern matching less expensive
- The RETE network consists of *facts* and a set of *rules* which operate on them

- Rule-based programming allows knowledge to be represented as heuristics, or "rules of thumb“, which specify a set of actions to be performed for a given situation. (CLIPS)

RETE – As simply as possible



Defining facts and rules

- **Putting facts into workspace**
- **Define rules**
  - A special notation of ‘if then’
  - The if part is called LHS, then part RHS
- **In OPS5 (Forgy’s choice)**

  -
    ```
    (P Time 0x
        (Goal ↑Type Simplify ↑Object <x>)
        (Expression ↑Name<x> ↑Arg1 0 ↑OP *)
    -->
    (MAKE Expression ↑Name Expr1 ↑Arg1 1))
    ```

Defining facts and rules in CLIPS

- **Define facts**

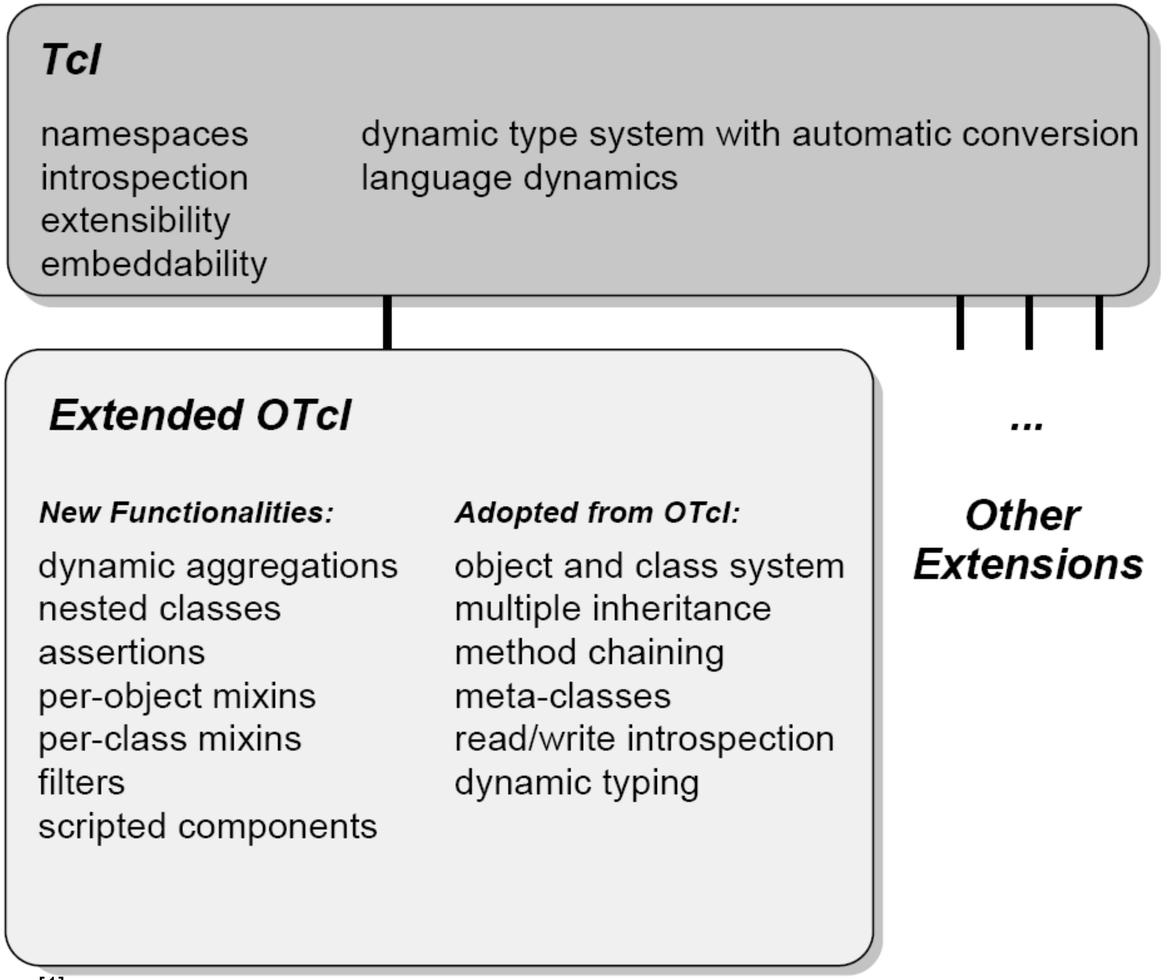
  -
    ```
    (assert (weather sunny))
    (assert (temperature high))
    ```
- **Define rules**

  -
    ```
    (defrule warm-day
     (and
         (weather sunny)
         (temperature high)
     )
    =>
    (assert (use-shorts true))
    )
    ```
- **Many others**
  - Drools Rule Language (DRL),…

XOTcl

- http://www.xotcl.org
- XOTcl = Extended OTcl = “Exotickle”
- XOTcl is used from the tclsh
  - package require XOTcl
    namespace import ::xotcl::*
- XOTcl is an OO Glue-Language for components
- XOTcl extends Tcl commands
  - **Object**: creates new generic objects
    Class: defines user classes
    …

Architecture of XOTcl



Tcl

namespaces
introspection
extensibility
embeddability

dynamic type system with automatic conversion
language dynamics

Extended OTcl

New Functionalities:

dynamic aggregations
nested classes
assertions
per-object mixins
per-class mixins
filters
scripted components

Adopted from OTcl:

object and class system
multiple inheritance
method chaining
meta-classes
read/write introspection
dynamic typing

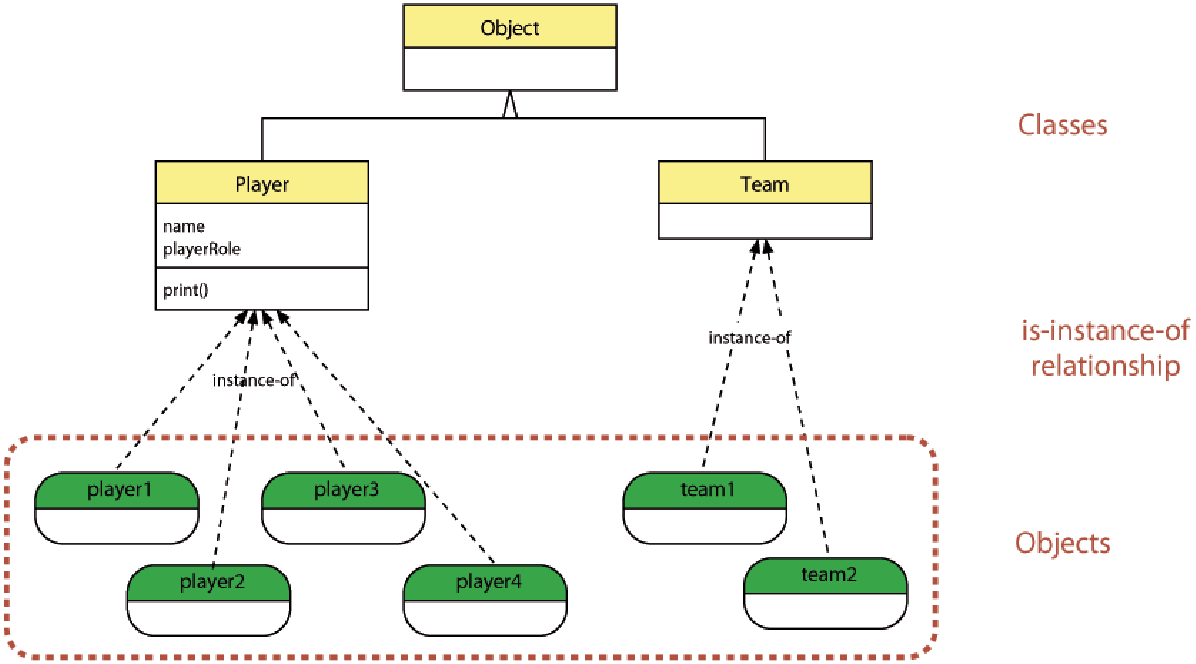
...

Other Extensions

see [1]

Object Oriented ?

- **Class Player -superclass Object**
- **Player player1 -name “Franz Wirl"**



see [1]

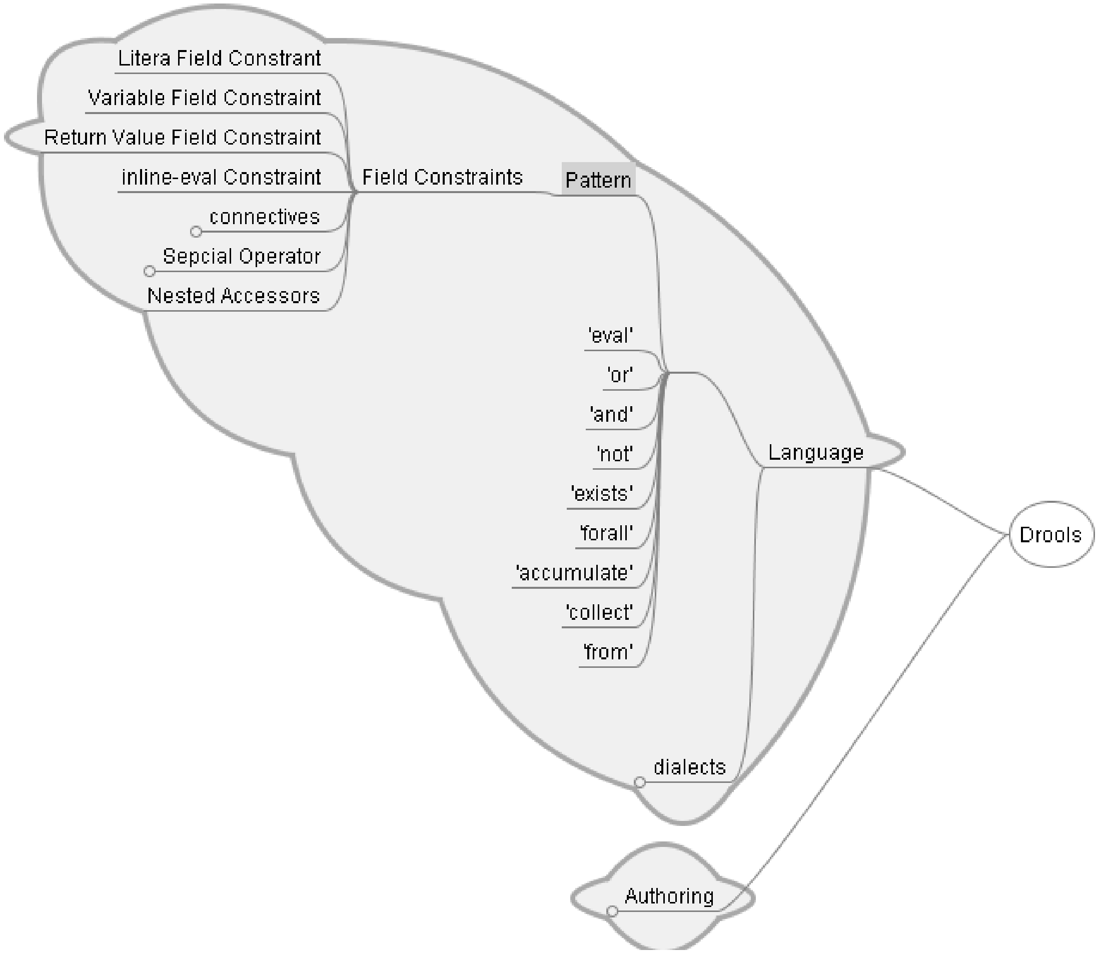
Why XOTcl

- **Using XOTcl as implementation language allows to focus on architectural design patterns of the Rete algorithm.**
- **Understanding the breakthrough of Rete in "matching data tuples (‘facts’) against productions (‘rules’) in a pattern-matching production system" and how Rete can improve language specific design patterns is one of the challenging tasks that have to be fulfilled.**

Nice, but what are you going to do?

- **Providing an interface for queries using the XOTcl based Rete algorithm.**
- **Define rules and facts with XOTcl**
  - Define generic classes for rules and facts
  - Provide access to operators and field constraints
    - (Define your own operators and constraints)
  - Handle bunch of rules and facts
  - Rule assertions are again XOTcl classes

Language constructs (Drools)



See [3]

‚but still?

- **Natural language processing**
  - Define business rules in natural language
  - Access to XOTcl classes in your own domain language
  - Mapping of DSL
  - DRL, OPS5
  -
- **Problems?**
  - Facts from DB // (materialized) Views
  - Read only, updateable, writeable

Queries with XOTcl (1)

- **Model**

```
Class Company -slots {
    Attribute name
    Attribute location
  }

Class Person -slots {
    Attribute name
    Attribute age
    Attribute employed -type::Company
  }
```

- **Facts**

```
Company c1 -name KM -location Vienna
Company c2 -name AS -location US
Company c3 -name WU -location Vienna

Person p1 -name Bernd -age 31 -employed ::c1
Person p2 -name Gustaf -age 77 -employed ::c3
Person p3 -name Franz -age 28 -employed ::c3
Person p4 -name Jeff -age 55 -employed ::c2
```

Queries with XOTcl (2)

- **Simple output**

  -
    ```
    set r [Company select {$location eq "Vienna"}]
    puts "companies in Vienna -> $r\n"
    ```

  -
    ```
    set r [Person select {[$employed location] eq "Vienna"}]
    puts "people working for a company in Vienna -> $r\n“
    ```
- **Using Ordered Composites**

  -
    ```
    # First, define a meta class "View“
    Class View -superclass Class
    View instproc init {as definition} {
      my eval $definition
    }
    ```

Queries with XOTcl (3)

```
#
# Define a materialized View Workplace...
#
View Workplace as {
  forall {Person p Company c} {
    my set name [$p name]
    my set location [$c location]
  } where {
    [$p employed] eq $c && [$c location] eq "Vienna"
  }
}

#
# Show the results
#
foreach w [Workplace info instances] {
  puts stderr "$w location [$w set location] name [$w set name]"
}
```

Thanks

- **Enjoy the prizes and have a nice journey back again.**

References

[1]..Gustaf Neumann, Uwe Zdun, “XOTcl an object-oriented scripting language” http://nm.wu-wien.ac.at/research/publications/xotcl-tclconf.pdf

[2]..CLIPS Tutorial, http://iweb.tntech.edu/bhuguenard/ds6530/ClipsTutorial/tableOfContents.htm

[3].. JBoss Drools Documentation, http://downloads.jboss.com/drools/docs/