

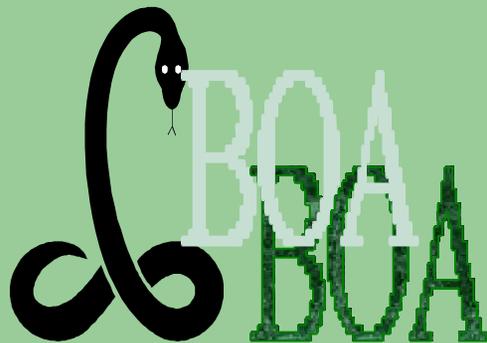


# Framework for Automated Builds

Natalia Ratnikova

CCS Meeting

Feb 2003 CMS Week





# Project Goals

---

- Facilitate software maintenance, help to improve software quality in the areas of software development, release management and distribution processes with the aids of the automated builds.
- Systematize available tools and components and put them all to work together in a highly automated fashion
- This activity is a part of the CMS software quality improvements project (see CMS internal note 2002/050, task 3.2–b specifications).



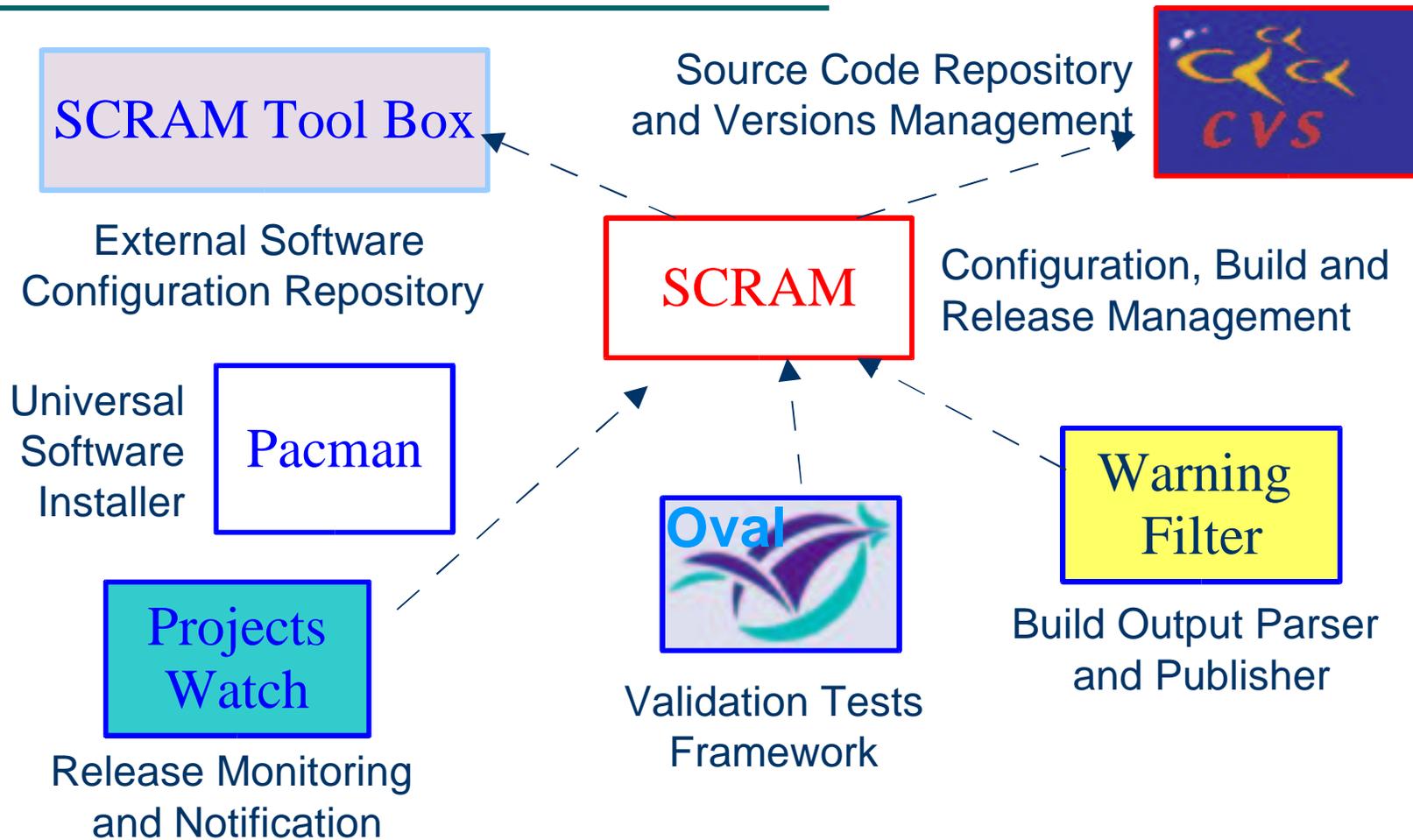
# What Is BOA

---

- Acronym for **B**uild and **O**utput **A**nalyzer
- Framework that allows to generate, control and analyze various types of automated software builds and tests, such as:
  - Regular rebuilds of the development code
  - Software integration builds for releases
  - Installation and build of the released versions
- Use cases and requirements presented and discussed on the CCS TC meeting 11/22/2002.



# Related Components and Tools





# Task Complexity and Conditions

---

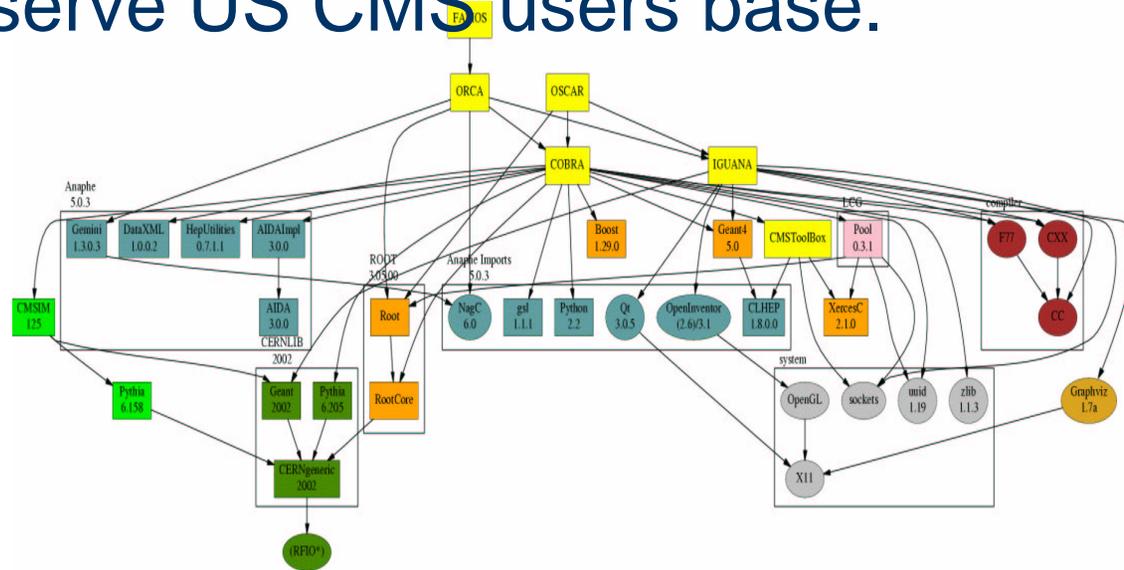
- Multiple software projects with anisochronous release schedule.
- Plenty of detailed specific information.
- Changing requirements and infrastructure.
- Lack of standardization for regular operations.
- Complicated and dynamic dependencies between components
- Site specific configuration management (most tedious and error-prone operations)



# Some Illustrations

- Over 300 versions released since Dec 2001, including pre-releases:  
ORCA:112, COBRA:57, OSCAR:48, CMSToolBox:32, FAMOS:22, IGUANA:21, Geometry:6
- Over 60 releases built at FNAL for the same period to serve US CMS users base.

Projects  
Dependencies:





# Methodologies

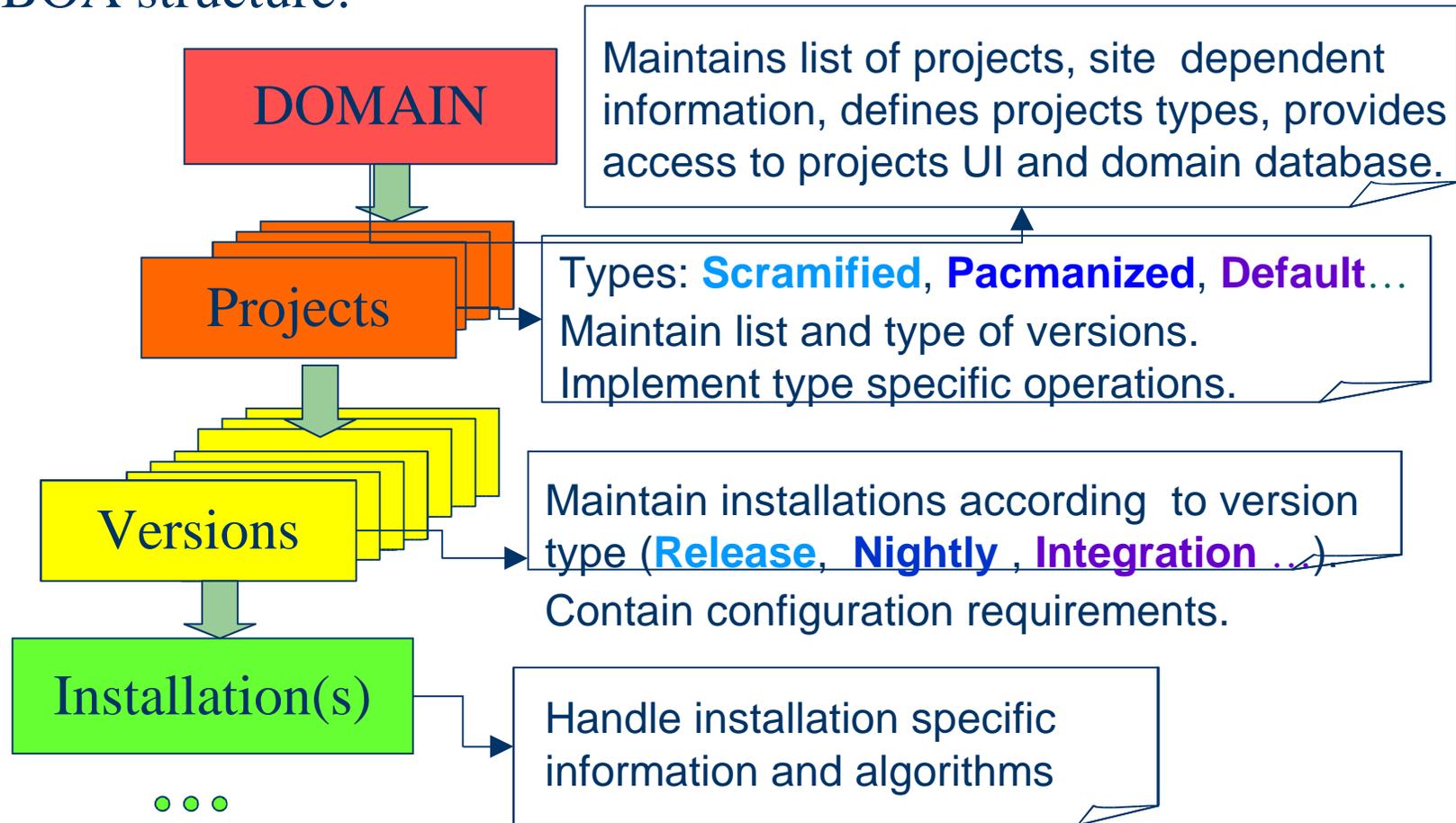
---

- **Model:**
  - Look for invariants in the complex system
  - Abstract the structure from the functionality, and the functionality from the implementation
- **Implementation:**
  - Accumulate and systematize the knowledge base
  - Provide standard interfaces to inter-changeable components
- **Process:**
  - Cyclic development and early prototyping
  - Constantly testing and documenting



# System Architecture

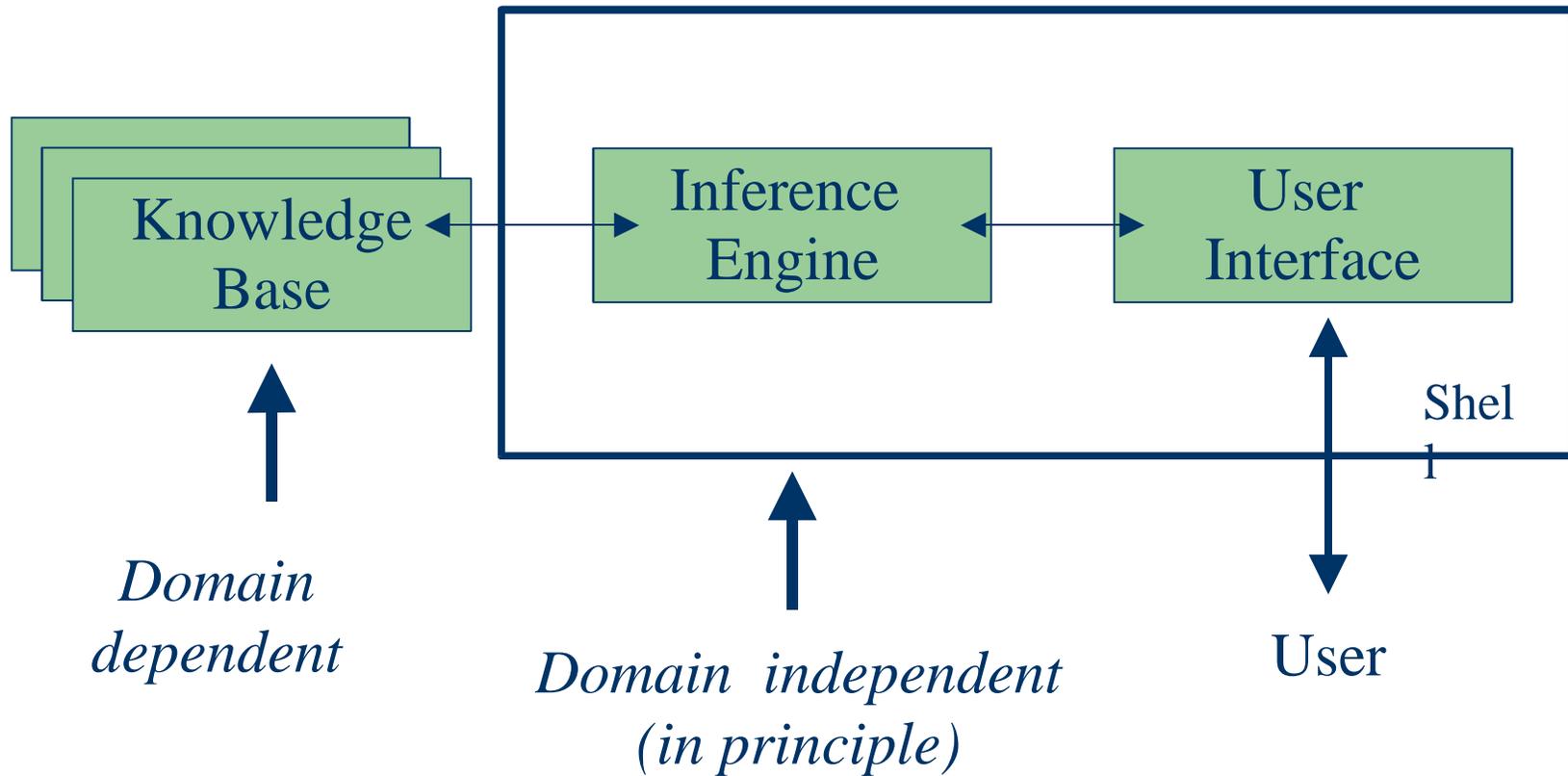
BOA structure:





# BOA Framework Architecture

*Corresponds to Basic Expert System Architecture:*





# System Features

---

- Proposed architecture allows to separate services from the implementation details.
- It makes the system very flexible and capable to install any types of projects (not only scram-managed) without any special efforts from the user. It is essential in case of external products!
- System provides convenient user interface with built-in help and standard interfaces to the underlying pluggable components.
- Allows for customized configuration on the domain level.



# Implementation and Status

---

- Experience with first prototype (written in Perl) has shown that structural programming, while providing solid base for algorithmic part of the system, does not allow required flexibility.
- The OO model is developed, base classes and their responsibilities identified.
- Python is chosen as a programming language.



## Implementation and Status (2)

---

- Base classes (Framework, Domain, Project, Version), major subclasses, user interface and persistency mechanisms are implemented.
- Currently work on the algorithmic part and implementing standard interfaces to the components with the aim to provide the base use cases functionality.



# Conclusions

---

- The base concept and architecture for the automated build system have been developed.
- Proposed architecture allows to use existing tools via standard interfaces and provides required flexibility for various types of projects and builds.
- OOP approach and python scripting language were chosen for the implementation.
- The core classes and components of the system are implemented and work. Work on implementing standard interfaces in progress.