# ADVANCED SOFTWARE DEVELOPMENT WORKSTATION PROJECT*

Daniel Lee
Inference Corp.

The Advanced Software Development Workstation Project, funded by Johnson Space Center, is investigating knowledge-based techniques for software reuse in NASA software development projects. Two prototypes have been demonstrated and a third is now in development. The approach is to build a foundation that provides passive reuse support, add a layer that uses domain-independent programming knowledge, add a layer that supports the acquisition of domain-specific programming knowledge to provide active support, and enhance maintainability and modifiability through an object-oriented approach. The development of new application software would use specification-by-reformulation, based on a cognitive theory of retrieval from very-long-term memory in humans, and using an Ada code library and an object base. Current tasks include enhancements to the knowledge representation of Ada packages and abstract data types, extensions to support Ada package instantiation knowledge acquisition, integration with Ada compilers and relational databases, enhancements to the graphical user interface, and demonstration of the system with a NASA contractor-developed trajectory simulation package. Future work will focus on investigating issues involving scale-up and integration.

* Funded under NASA Contract NAS-9-17766.

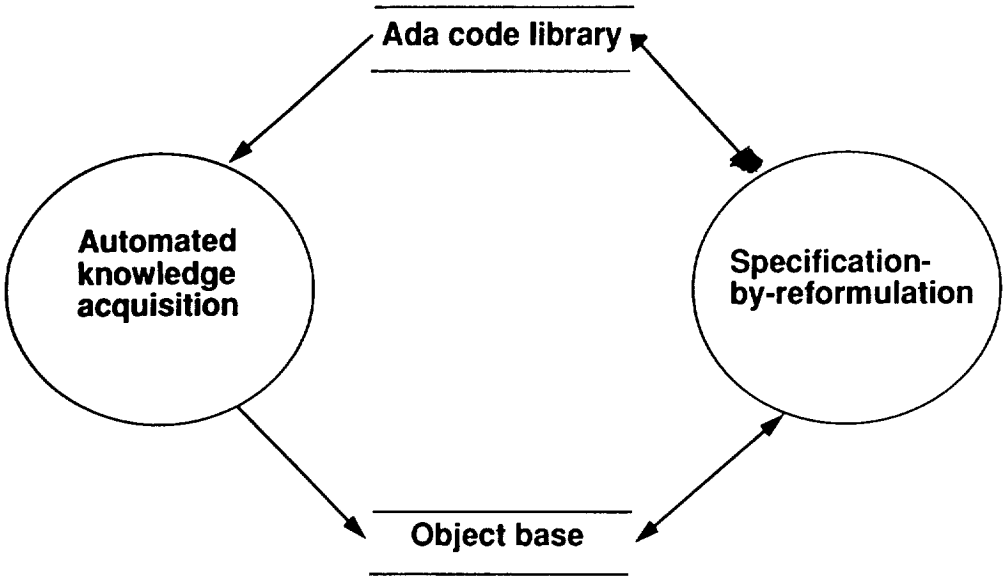
## The Software Reuse Process
(from [Prieto-Diaz and Freeman 87])

```
begin
  retrieve matching components from catalog
  if identical match
    then use matching component
    else
      begin
        select best matching component
        modify matching component
      end
end
```

## ASDW Technical Approach

- Build a foundation that provides passive reuse support (catalog retrieval and parts composition).
- Add a layer that uses domain-independent programming knowledge to provide interactive support (syntactic constraint checking and code generation).
- Add a layer that supports the acquisition of domain-specific programming knowledge to provide active support (semantic constraint checking).
- Enhance maintainability and modifiability through an object-oriented approach.

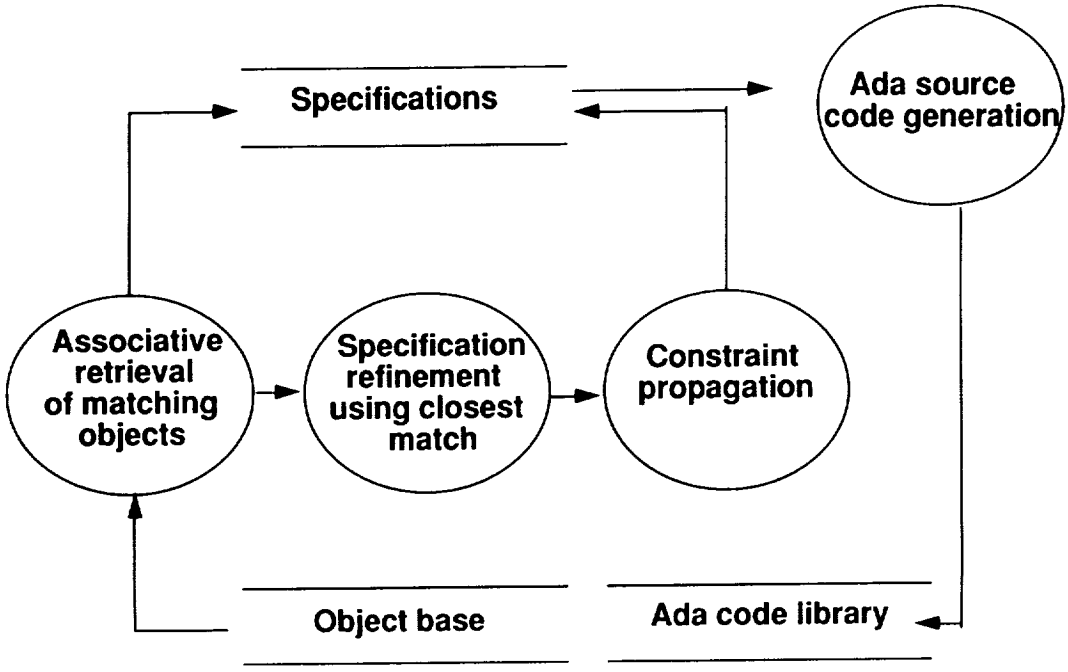
## Application Development Using the ASDW




## Specification-by-reformulation

- A generic user interface architecture for task support.
- Based on a cognitive theory of retrieval from very-long-term memory in humans.
- A specification-by-reformulation environment consists of:
  - A specification language.
  - A mechanism for providing feedback to the user about the current specification.
  - A mechanism for performing actions on specifications.
- Using specification-by-reformulation for software parts composition:
  - Domain object descriptions and library package specifications form an application-specific specification language.
  - Constraint propagation provides feedback.
  - Specialization and generalization of specifications and code generation are actions.

## The Specification-by-reformulation Process




# ASDW Project: Work in Progress

- Current tasks:
  - Enhancements to the knowledge representation of Ada packages and abstract data types.
  - Extensions to support Ada package instantiation knowledge acquisition.
  - Integration with Ada compilers and relational databases.
  - Enhancements to the graphical user interface.
  - Demonstration of the system with a NASA contractor software library (trajectory simulation package).
- Third prototype demonstration: 2/89.

# ASDW Project: Future Work

- Goal: Investigate issues involving scale-up and integration.
- Tasks:
  1. Develop and integrate associative retrieval algorithms for use with large software libraries.
  2. Develop and integrate conceptual clustering algorithms for automatic taxonomy generation.
  3. Integrate prototype with NASA software development environment.
  4. Conduct prototype evaluation in conjunction with an ongoing NASA contractor Ada development project.
- Fourth prototype demonstration: 10/89.