PROTOTYPE SOFTWARE REUSE ENVIRONMENT AT GODDARD SPACE FLIGHT CENTER

Walt Truszkowski
NASA Goddard Space Flight Center

The Goddard Space Flight Center (GSFC) work is organized into four phases and includes participation by a contractor, CTA, Inc. The first phase was an automation study, which began with a comprehensive survey of software development automation technologies. Eight technical areas were analyzed for goals, current capabilities, and obstacles. The study documented current software development practice in GSFC Mission Operations and Data Systems Directorate, and presented short- and long-term recommendations that included focus on reuse and object-oriented development. The second phase, which has been completed, developed a prototype reuse environment with tools supporting object-oriented requirements analysis and design. This phase addressed the operational concept of software reuse, i.e., it attempted to understand how software can be reused. This environment has two semantic networks: object and key words, and includes automated search, interactive browsing and a graphical display of database contents. Phase 3 was a domain analysis of Payload Operations Control Center (POCC) software. The goal in this phase was to create an initial repository of reusable components and techniques. Seven existing Operations Control Centers at GSFC were studied, but the domain analysis proved to be very slow. A lesson learned from this was that senior people who understand the environment and the functionality of the area are needed to perform successful domain analyses. Four reuse paradigms were identified which are appropriate to different parts of a POCC. Phase 4 is the development of a prototype environment for rapid synthesis of POCC software. The four paradigms (or views) of software reuse will be prototyped and combined to provide support for POCC software development. These four paradigms are a dialog-based specification of high-level architecture, a very-high-level-language specification of the operational database, interface navigation/selection of reusable components, and graphical programming. Future work includes the design of a knowledge-based reuse environment.

# Workshop on NASA Research in Software Reuse

Phase 1 (FY '86): Automation Study

- Comprehensive survey of software development automation technologies

- Analyzed 8 technical areas: goals, current capabilities, obstacles
  - Semi-formal specification, formal specification, reuse, knowledge-based systems, prototyping, software metrics, performance analysis, work management

- Documented current S/W development practice in GSFC Mission Operations & Data Systems Directorate
  - Methods, tools, perceived strengths/weaknesses

- Short- and long-term recommendations
  - Focus on reuse and Object Oriented Development (OOD)
 - Revisit CHI and AI around 1990

Phase 2 (FY '87): Prototype Reuse Environment

- Tools supporting object-oriented requirements analysis and design
  - Extended Goddard Object Oriented Design (GOOD) methodology to requirements analysis
  - Enhanced IDE Software Through Pictures environment

- Operational concept of software reuse

- Two semantic networks: objects and key words
  - Obnet: entity-relationship database of reusable components
  - Keynet: classification of reusable components

- Automated search and interactive browsing

- Graphical display of database contents

SEMANTX Architecture

The RMS KeyNet

51
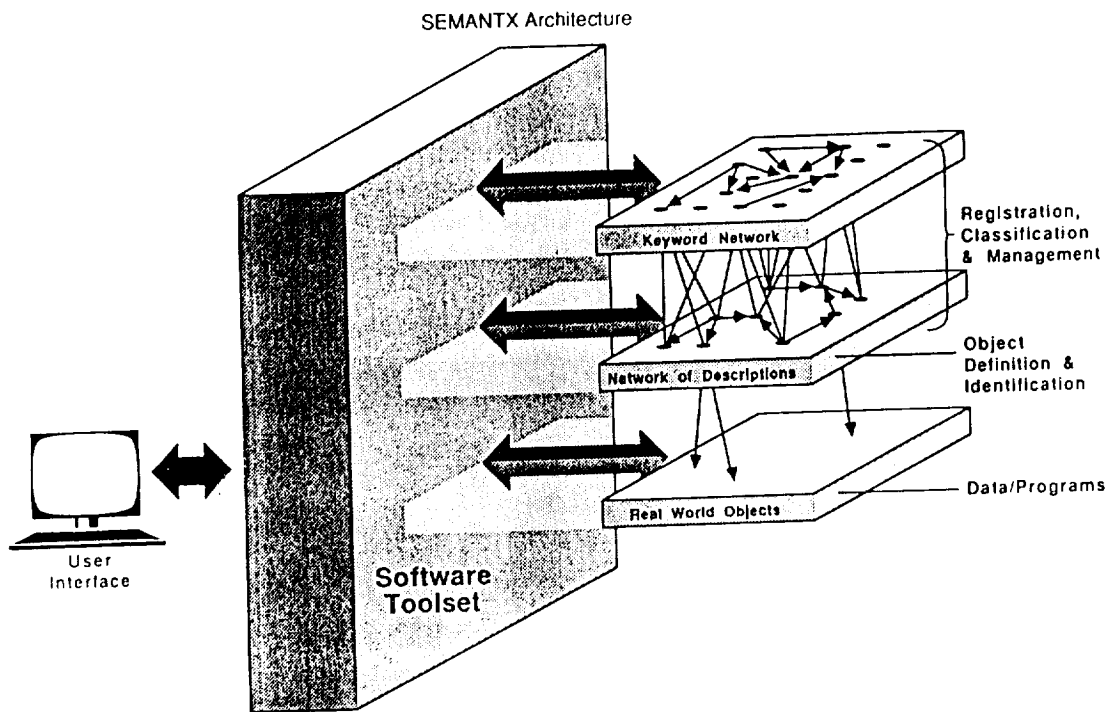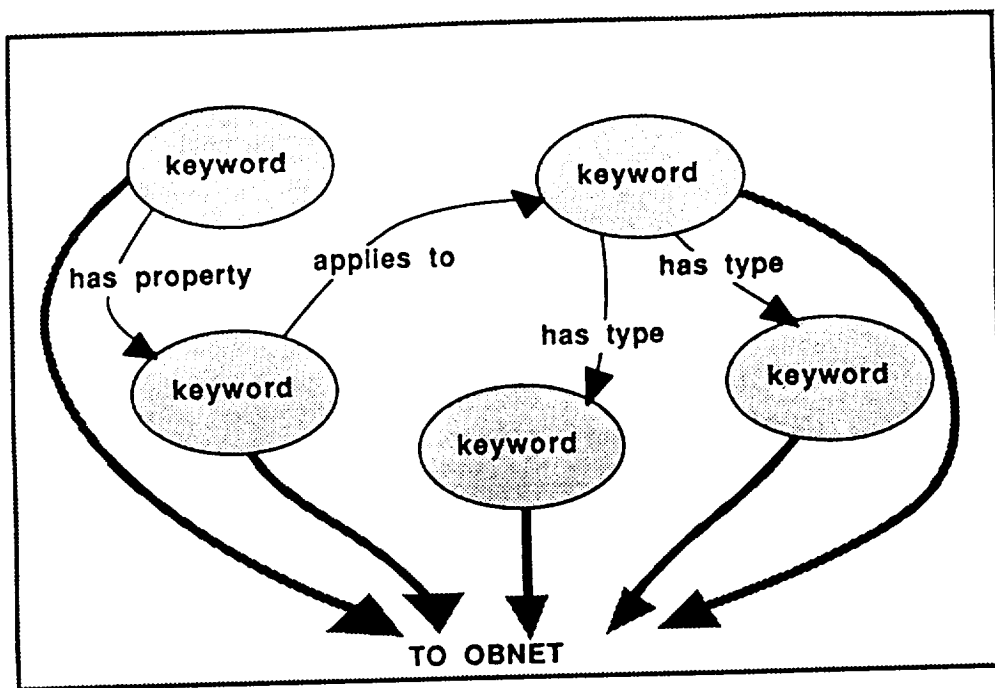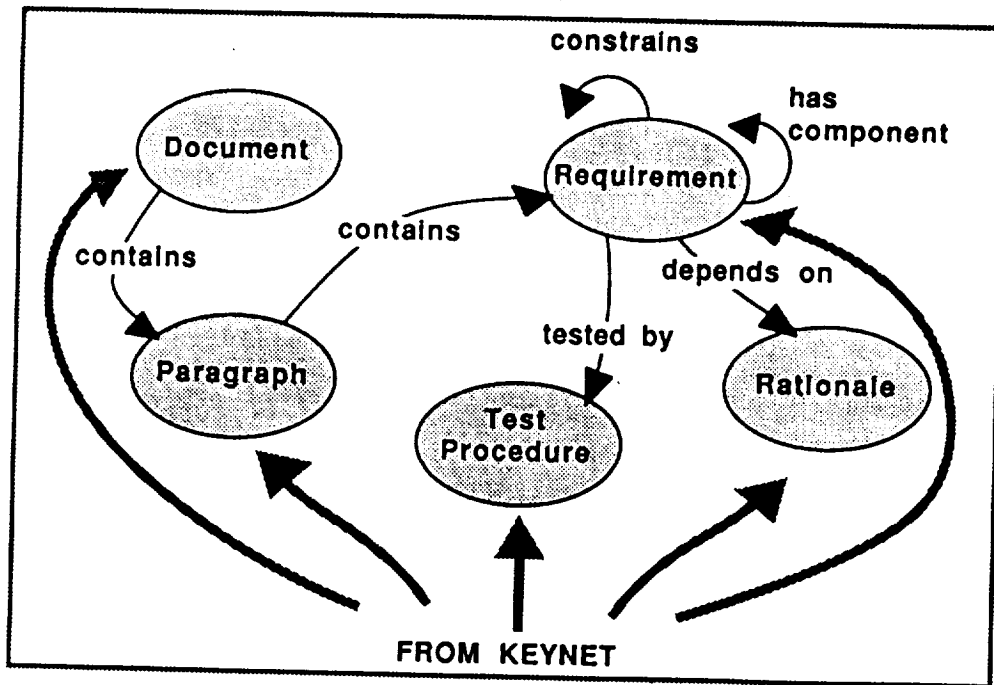
Phase 3 (FY `88): Domain Analysis of POCC Software

• **Goal: Create an initial repository of reusable components and techniques**

• **Studied seven Multi-Satellite Operations Control Center (MSOCC) systems**
  - Standard Software
  - Dynamic Explorer (DE)
  - International Sun/Earth Explorer (ISEE)
  - Earth-Radiation Budget Satellite (ERBS)
  - MSOCC Applications Executive (MAE)
  - Gamma Ray Observatory (GRO)
  - Cosmic Background Explorer (COBE)

• **Determine typical POCC architecture and components**
  - **Classified variations**
  - **Identified obstacles to reuse**

• **Identified 4 reuse paradigms appropriate to different parts of a POCC system**

Phase 4 (FY `89): Prototype Environment for Rapid Synthesis of POCC Software

- **Dialog-based specification of high-level architecture**

- **Very-high-level language specification of Operational Database**
  - **Automated generation of database interface procedures**

- **Interface navigation/selection of reusable components**

- **Graphical programming**
  - **Specify new combinations of reusable components**
  - **Automated code generation from Object and Functional Diagrams**

Four Automation Techniques Combine to Support POCC Software Development
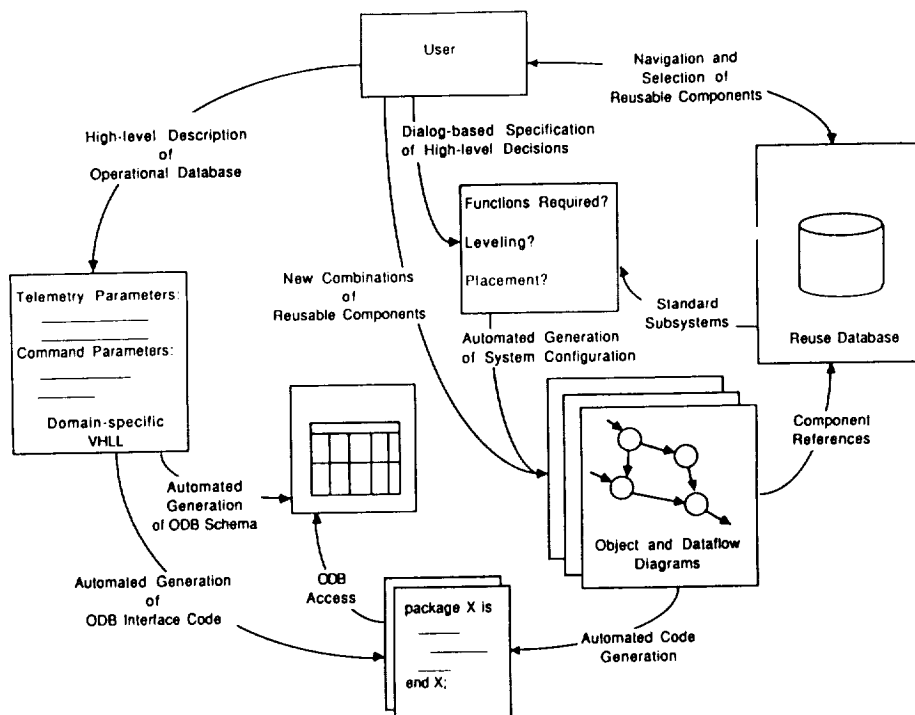
AI Revisited: Design of Knowledge-based Reuse Environment (FY `89)

- Survey recent efforts
    - Determine available technologies
    - Develop a knowledge-based reuse concept

- Focus on essential areas not yet explored
    - Capturing developer rationales
    - Learning from errors (e.g., misused components)