

Impact of Software Engineering Research on Modern Programming Languages

Report on a work in progress !

Barbara G. Ryder and Mary Lou Sofa

What is this project?

- **Goal:** Determine and document the impact that fundamental software engineering research has had on modern programming languages
- Focus on languages currently in widespread use: e.g., Java, C++, Ada, Perl
- Find origins of, and influences on, essential features of these languages

Challenges

- **PL and SE have a complex, close, synergistic relationship**
 - **Hyp 1: SE research impacted PL design**
 - **Hyp 2: PL impacted SE research and practice**
 - **SE(PL) research also influenced SE(PL)**
 - **Parnas on modularity -> OO design**
 - **Simula 67 -> C++, Java**

Challenges

- **Some researchers active in both fields (e.g., N. Wirth, C.A.R.Hoare)**
 - **Related publications in conferences and journals of both fields**
- **How to attribute these contributions?**

Sources

- **Use ACM and IEEE-CS digital libraries to find first-source articles**
- **HOPL-I (1978) and HOPL-II(1993)**
 - **Document history of language development**
 - **C, C++, Ada, Simula67**
- **Rationale for the Design of the Ada Language (1979)**
 - **PL designers tell reasons for some features**

PL Features and Constructs

- **Data abstraction, encapsulation**
- **Inheritance**
- **Visibility**
- **Types**
- **Generics**
- **Modules and scope**
- **Exceptions**
- **Concurrency**
- **Procedural abstraction**
- **PL environments and tools**

Research Process

- **Build timeline of PL historical development**
- **Compare to timeline of major research thrusts in SE**
- **Start with current PL's and identify some key features to trace back to origins (e.g., exceptions, information hiding, generics)**
- **Find SE concepts that appeared in SE research in same time frame**
- **Chart possible interactions**

Example:Modularity and Info Hiding

- **Roots of OO languages – (modules, objects, inheritance, dynamic method binding) – Simula 67**
- **Parnas (1972) was one of the first to recognize value of modularity (code + data)**
- **Parnas first to come up with notion of information hiding (1972)**
- **Refined notion of encapsulation appeared in CLU in 1977**

Example: Exceptions

- **Introduction of exceptions - ON conditions in PL/I early 1970's**
- **J. Gannon, J.J. Horning (CACM 1975) explore issues in PL design for reliability**
- **J. Goodenough (CACM 1975) defines exception conditions and proposes PL features for handling**
- **Mesa, Mitchell et al (1979) and Ada, J. Ichbiah et al. (Ada Rationale 1979)**
- **CLU, B. Liskov, A. Snyder (IEEE-TSE 1979) offers clean definition of handling**