
Le projet SPOOL:

Aperçu et perspectives

Rudolf K. Keller

<http://www.iro.umontreal.ca/~keller>

Comité de gestion du LUB, le 18 avril, 2000



SPOOL Project



Quality Engineering Group



GÉLO Group



*Consortium for Software Engineering Research
Consortium de recherche en génie logiciel*

SPOOL

Spreading Desirable Properties
into the Design of
Object-Oriented, Large-Scale
Software Systems

Laboratoire de génie logiciel (GÉLO)

Software Engineering Lab

- **Research group**
 - 3 professors (Keller, Sahraoui, Lustman)
 - 18 graduate students

- **Teaching SE at DIRO**
 - strong baseline curriculum in CS
 - *B.Sc. Option Génie Logiciel* (starting in fall 2000)



SPOOL Team

- **Bell Canada**

- Bruno Laguë (associate director, Quality Eng. & Research)
- Charles Leduc, Sébastien Lapierre

- **Université de Montréal**

- Rudolf K. Keller (PI), François Lustman
- R. Schauer, G. Knapen (research associates)
- H. Kabaili, I. Khriss, R. Schauer, N. Tagoug (Ph.D.)
- S. Bassil, B. Bouazza, A. Chaumon, P. Dufresne, F. Martel, P. Pagé, S. Robitaille, G. Saint-Denis, R. Yin (M.Sc.)

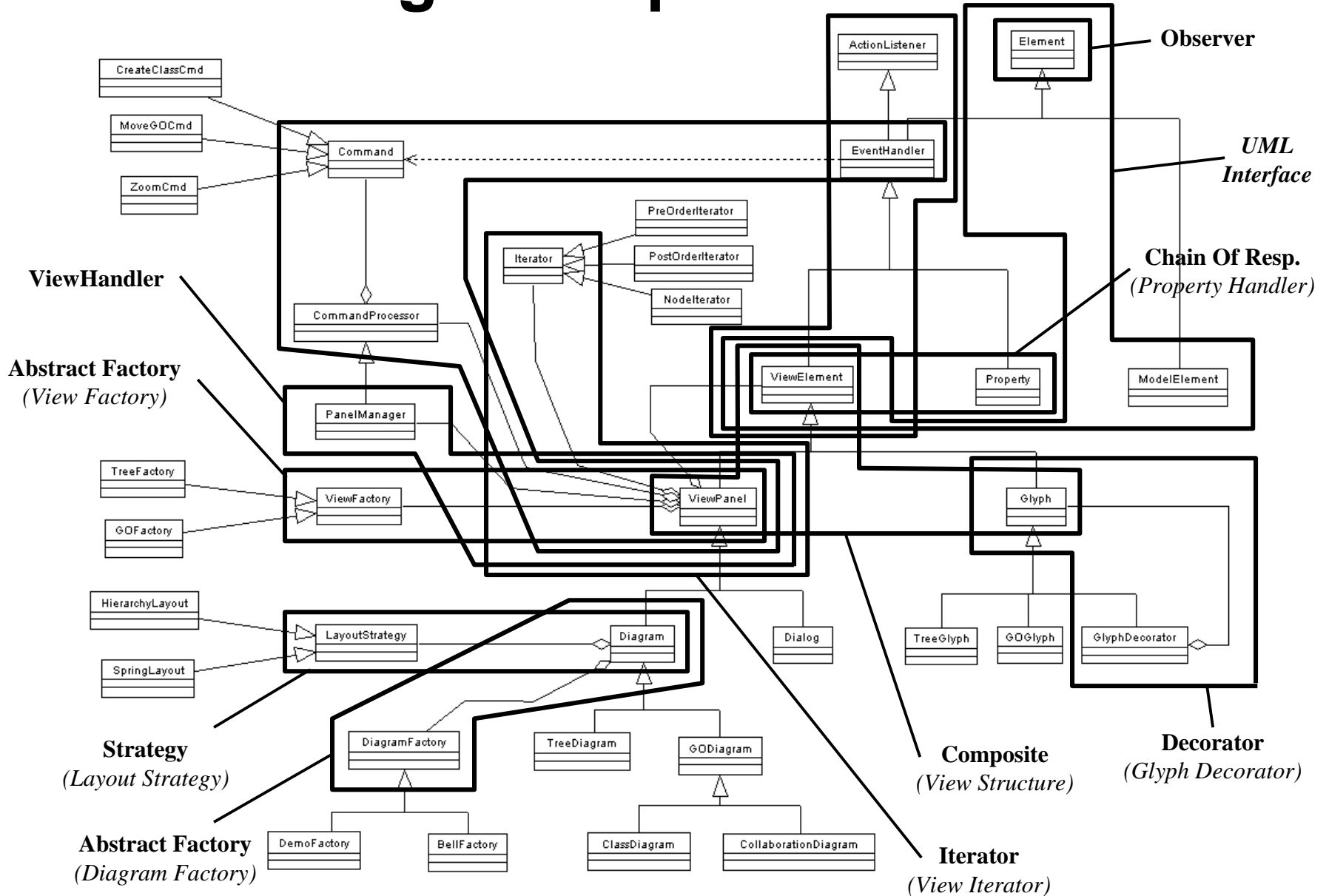
Context

- **Software acquisition in the large**
 - object-oriented/real-time systems, millions of LOC
 - conflicting goals: high reliability vs. addition of services
 - evolvability as major concern in product selection/risk assessment
- **Design is key factor of influence on evolvability of software**
- **Objectives of SPOOL project**
 - identify and quantify those design properties that are significant for evolvability
 - develop concepts for the systematic construction of high quality o-o designs

Key Questions and Subprojects

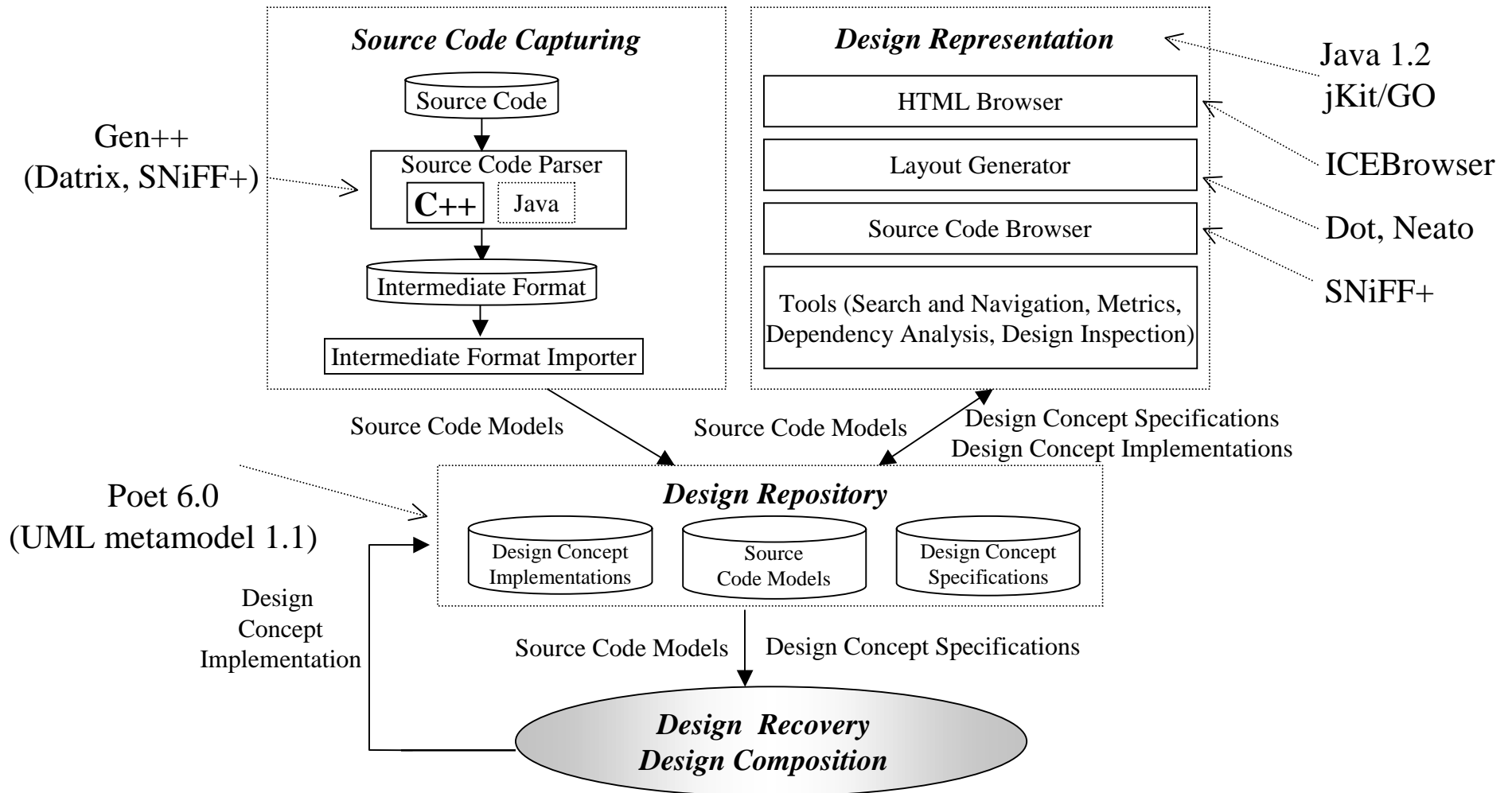
- **What makes for a good o-o design?**
 - Recovery of design components
 - Change impact analysis
- **How do you achieve a good o-o design?**
 - Design composition
 - Design transformation
- **Approach**
 - empirical: study of legacy software
 - constructive: design of new methods, tools, measures
 - experimental: evaluation of new concepts

Solution: Design Components



Recovery of Design Components

SPOOL Environment - Architecture



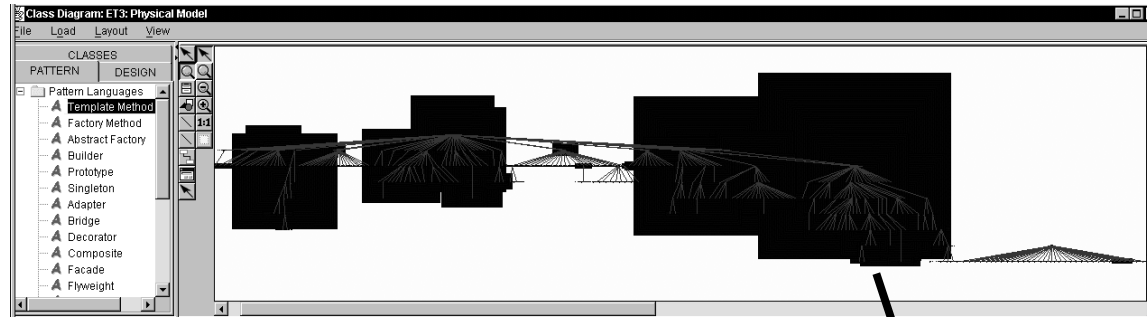
Recovery of Design Components

SPOOL Environment – User Interface

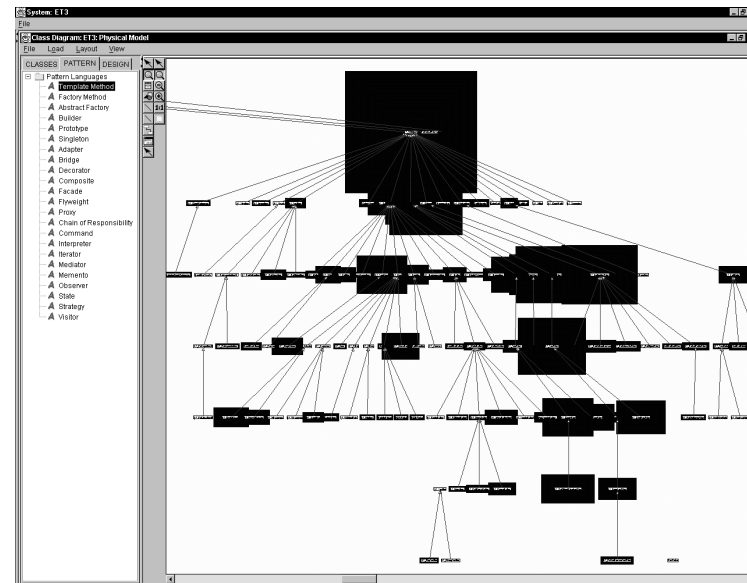
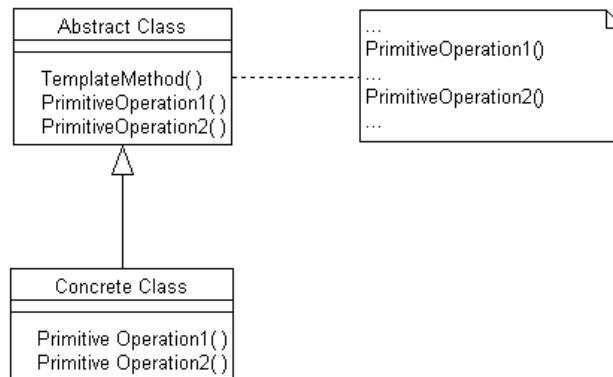


Recovery of Design Components

Detection of Design Patterns



Template Method:

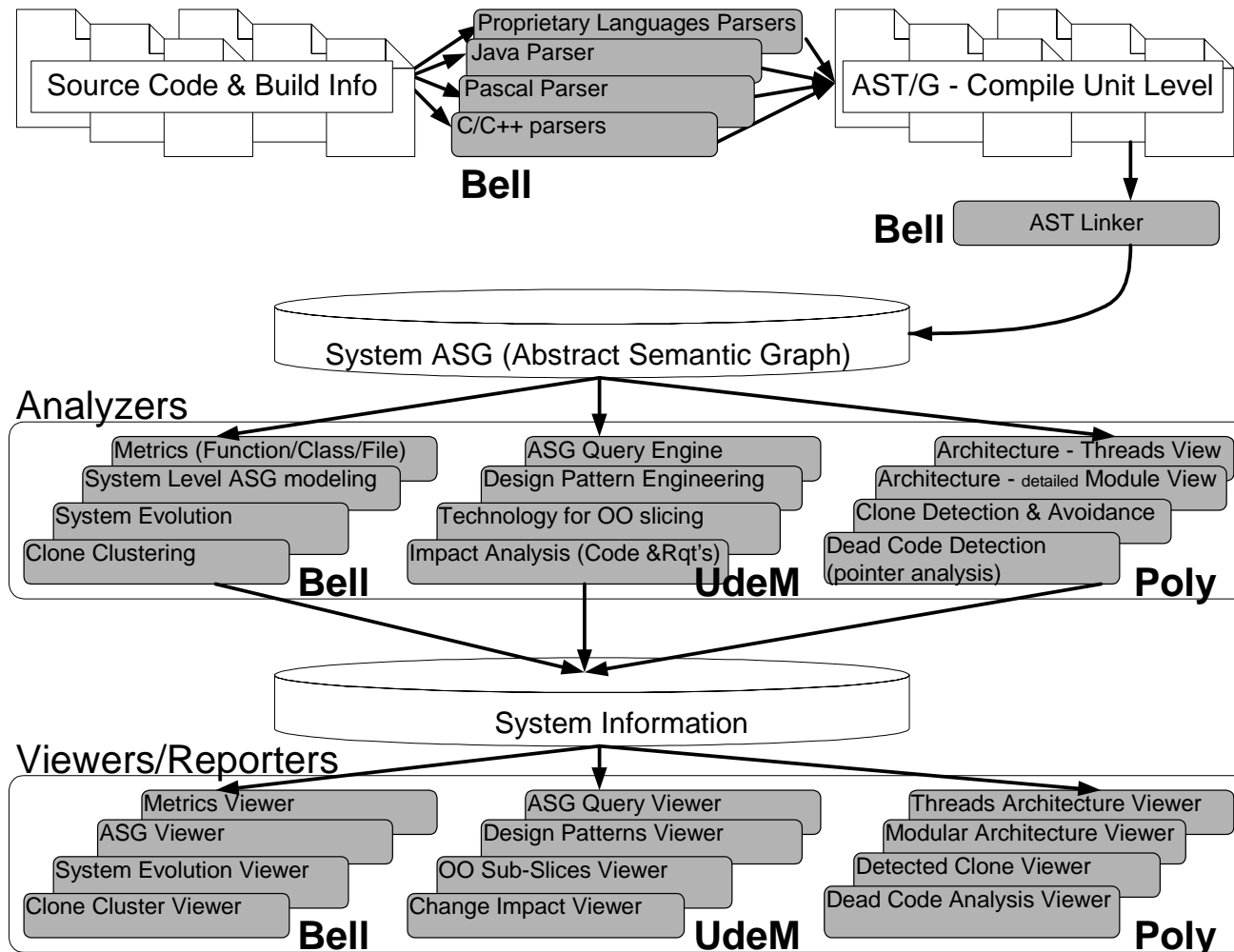


Recovery of Design Components

Achievements and Roadmap

- **Structural design patterns** (ICSE'99)
- **Hot spot recovery**
 - Best paper award (Intl. Conf. on Software Maintenance: ICSM'99)
- **Method replacement** (CSMR'2000)
- **Object-Oriented Recursion, Recursion Chains**
- **Code Smell, Antipatterns**
 - Fowler et al.: Refactoring
- **Design Pattern Catalog**
 - Gamma et al.: Design Patterns (“Pattern Bible”)

Bell Environment Architecture



Key Questions and Subprojects

- **What makes for a good o-o design?**
 - Recovery of design components
 - Change impact analysis
- **How do you achieve a good o-o design?**
 - Design composition
 - Design transformation

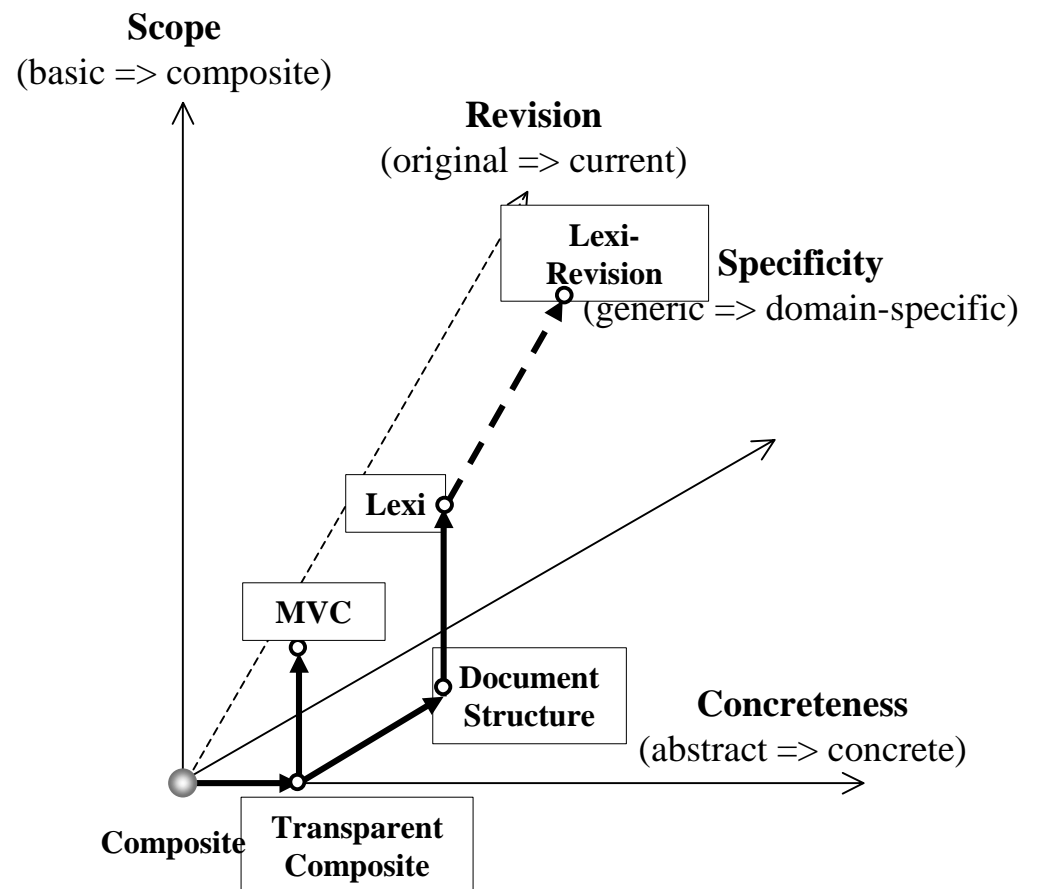
Change Impact Analysis

- **assess system quality with respect to evolution**
- **definition of change impact model**
- **implementation of model for C++ language**
- **experimental studies on industrial systems:**
 - nine design metrics, six typical changes
 - high correlation between changeability and the access to a class by other classes through method invocation or variable access
 - no measurable influence of depth of inheritance tree on changeability

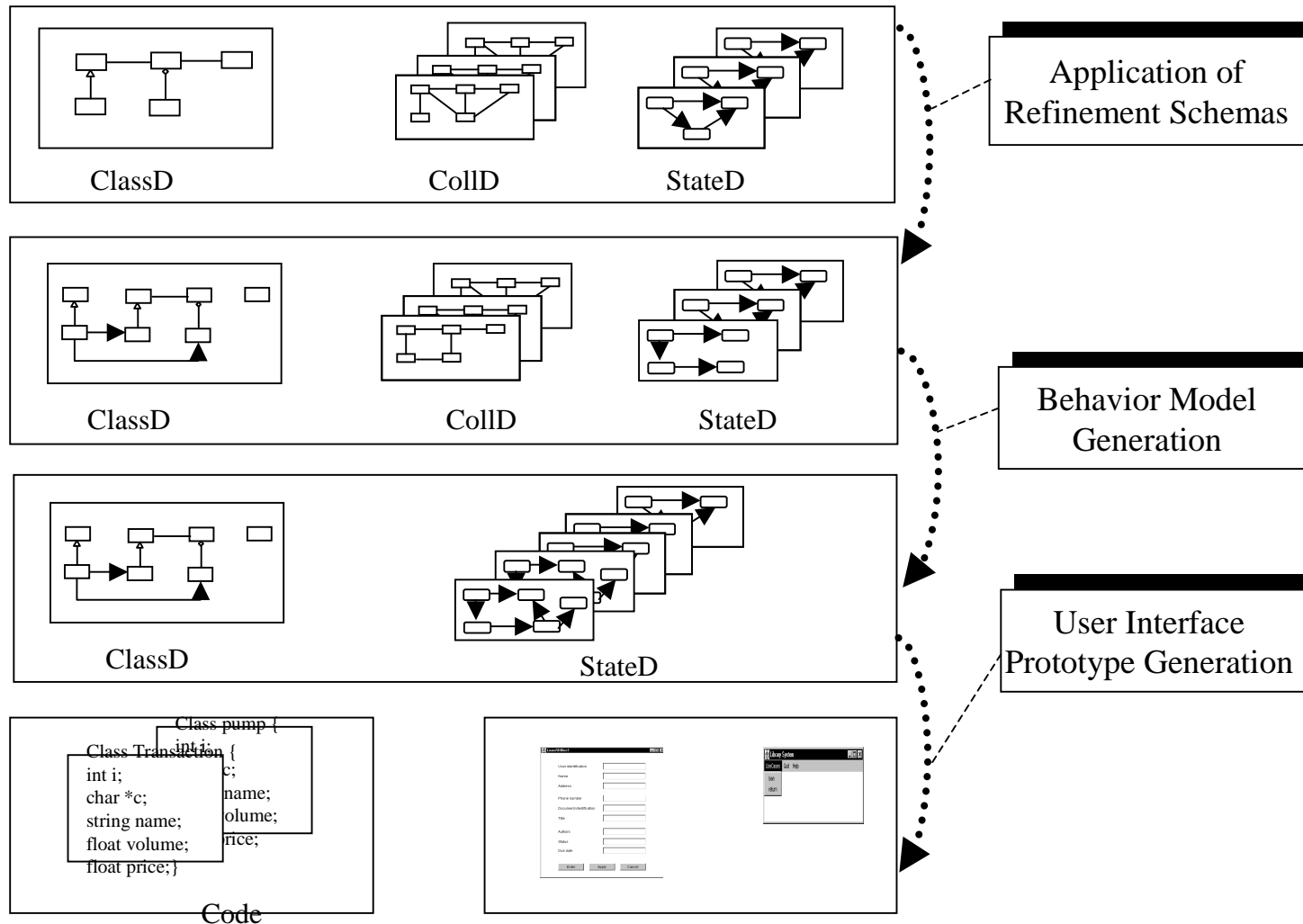
Design Composition

Design component: reified design solution (pattern) fit for software component.

- **Concreteness**
 - level of detail
- **Specificity**
 - applicability to domains
- **Scope**
 - level of composition
- **Revision**
 - change history



Design Transformation



SPOOL Project

Results of First Four Years

- **Scientific and technical highlights**
 - numerous publications, incl. 4 chapters in upcoming CSER Book
 - organization of workshops: OOPSLA'97, DEXA'98, OOPSLA'2000(?)
 - SPOOL reverse engineering environment: demos at IBM CASCON, ICSE'99 Los Angeles, etc.
- **Technology transfer Bell-UdeM:** SPOOL design browser, tools
- **Educational impact**
 - training of 3 Ph.D., 10 M.Sc. students, 2 research associates
 - 3 new/adapted software engineering courses
- **Interaction with Bell, CSER, research community**
 - Strong Canadian presence in reverse engineering community

SPOOL Project

Lessons Learned (1)

- **Industry-led research**

- industrial partner as client, increased pressure to succeed
- additional criteria of success: applicability, relevance
 - examples: SPOOL renewal proposal, Datrix integration
- research proposal as contract with deliverables from both sides
 - example: availability of data for experimentation
- interaction culture
 - examples: student presentations for Bell group at 3 months intervals, GÉLO seminar with speakers from Bell

SPOOL Project

Lessons Learned (2)

- **Reverse engineering research**
 - powerful tool infrastructure is essential
 - costly, long-term investment
 - charms and risks of high-tech tools and standards, e.g., JavaCC, UML metamodel
 - importance and risks of tool reuse, e.g., parsing and visualization tools
 - availability of industrial-strength case studies is essential

SPOOL Project

Lessons Learned (3)

- **Recruitment of quality students**
 - structural problem: limited number of available students
 - selling factors: industry-led project, practical emphasis (tools, experiments), scholarships
 - chicken and egg situation

Conclusion

- **Reverse engineering as a domain of high practical and scientific interest**
- **Lessons learned**
- **Industry-university collaboration proves highly rewarding**
- **SPOOL - Two more years ahead !**
- **More about the project:**
 - <<http://www.iro.umontreal.ca/labs/gelo/spool>>
 - <<http://www.iro.umontreal.ca/labs/gelo/datrix>>
 - <<http://www.iro.umontreal.ca/~keller/publications.html>>