

# Characterizing the Technical Potential of a Module

ICS 225 Project, Winter 2005

Sushil Bajracharya and Trung Ngo

{sbajrach,trungcn}@ics.uci.edu

## Background - Technical Potential

- Net Options Value is a mathematical model for evaluating design options based on the economic theory of real options
  - Design is a value seeking process
  - Making changes in design is an experiment with uncertain outcome
  - In a modular system, the total Value of the System is the sum of the NOV's of each module

$$\text{NOV of } i^{\text{th}} \text{ module} = \max_{k_i} \left\{ \begin{array}{l} \text{Expected value} \\ \text{from } k\text{-experiments} \end{array} - \begin{array}{l} \text{Cost incurred in} \\ k\text{-experiments} \end{array} - \text{Visibility cost to} \right. \\ \left. \text{replace } i^{\text{th}} \text{ module} \right\}$$

- $\max_{k_i}$  is the maximum value out of all experiments for the  $i_{\text{th}}$  module
- **Cost** is the redesign cost for each of the  $k$  number of experiments
- **Visibility Cost** is the cost incurred in making changes in all other modules that depend on the  $i_{\text{th}}$  module

## What is technical potential 'σ' ?

Degree of *uncertainty* inherent in a module

$$NOV_i = \max_{k_i} \left\{ \overbrace{\sigma_i n_i^{1/2} Q(k_i)}^{\text{Benefit}} - \underbrace{C_i(n_i)k_i}_{\text{Investment}} - Z_i \right\}$$

$$\text{NOV of } i^{\text{th}} \text{ module} = \max_{k_i} \left\{ \begin{array}{l} \text{Expected value} \\ \text{from } k\text{-experiments} \end{array} - \begin{array}{l} \text{Cost incurred in} \\ k\text{-experiments} \end{array} - \text{Visibility cost to} \right. \\ \left. \text{replace } i^{\text{th}} \text{ module} \right\}$$

- *Environmental Factors govern the value of  $s$*  [sul+01]
  - Users, Data, Deployed Environment

- External Parameters are one of the environmental parameters [lop+dsm:05]

## Validating the heuristics for $\sigma$

- ICS 225 Project
  - Verify intuitive understanding about  $s$  against the evolutionary data from real world example
  - Application under study
    - ArgoUML
  - External Parameters
    - Libraries that contribute to UI, UML model, data (XML) and others..
  - $s n^{1/2}$  = degree of uncertainty = deviation in risk factors
  - Open issues in the project are indicators of technical risk
  - *Hypothesis*: Increase in External Libraries should increase the technical potential and hence, open issues.

# Data Collection

- ArgoUML
  - Active, 10-year old project
  - Latest version: more than 3000 .java files
- Data Sources
  - ArgoUML's CVS repository
    - More than 44 "tagged" releases in CVS repository
  - ArgoUML's Bugs and Issues Database
    - More than 3000 bug/issue tickets
    - Tickets are associated with SUBCOMPONENT
      - Ex: Class Diagram, Model,...

# Data Collection – Logical Component

|         |              |
|---------|--------------|
| Rows    | Subcomponent |
| Columns | Version      |

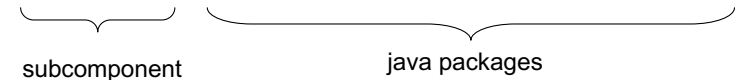
Generate report

| Subcomponent:                     | 0.10 | 0.10.1 | 0.11.1 | 0.11.2 |
|-----------------------------------|------|--------|--------|--------|
| Build scripts and tools           | 1    |        |        |        |
| ClassDiagram                      | 16   | 9      | 11     | 8      |
| Classfile module                  |      |        |        |        |
| CodeGeneration/ReverseEngineering | 5    | 4      | 1      | 1      |
| Contents                          |      |        |        |        |
| Cop module                        |      |        |        |        |
| Critics                           | 2    |        | 1      |        |
| Csharp module                     |      |        |        |        |
| DeploymentDiagram                 | 2    |        |        |        |

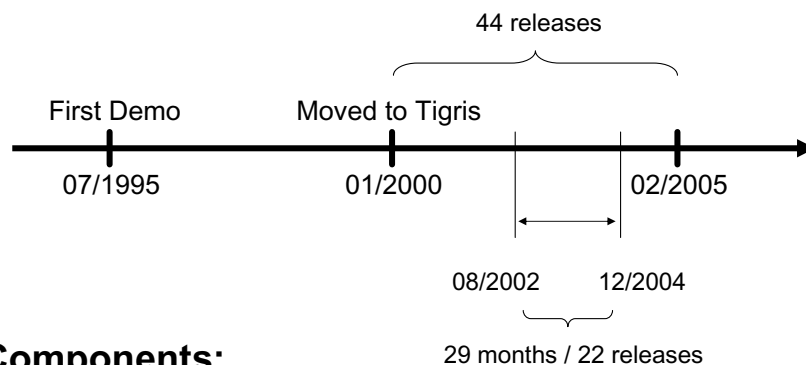
## • Logical component:

– A mapping between "subcomponents defined in issuezilla database and java source files

Example: "ClassDiagram" = org.argouml.uml.diagram.static\_structure +  
org.argouml.uml.diagram.static\_structure.layout +  
org.argouml.uml.diagram.static\_structure.ui



# Data Collection



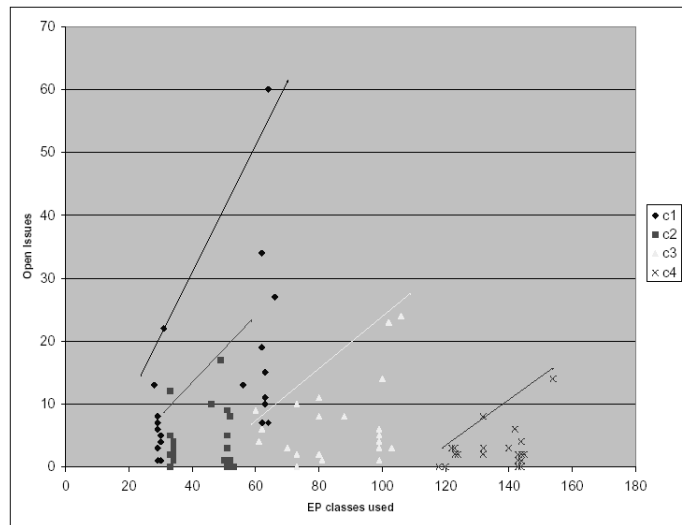
## Components:

- ❖ Class Diagram
- ❖ State/Activity Diagram
- ❖ Code generation/Reverse Engineering
- ❖ Model

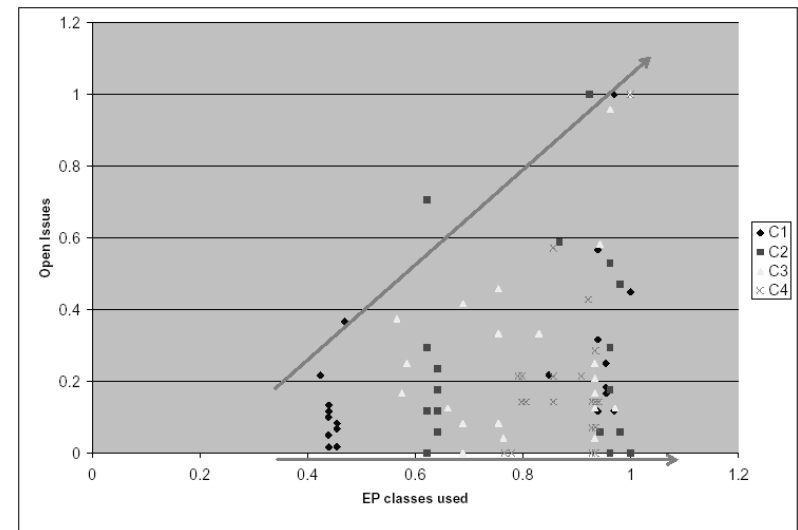
# Data Collection

- Environmental params
  - Measure dependency between ArgoUML's components and external libraries
    - Metric: Number of external classes used
  - Tools used:
    - Ant, DependencyFinder, XSLT, and Excel
- Indicators of sigma
  - Measure number of tickets in issuezilla database for each logical component
  - Tools used:
    - Ant, Issuezilla's query tools, XSLT, and Excel

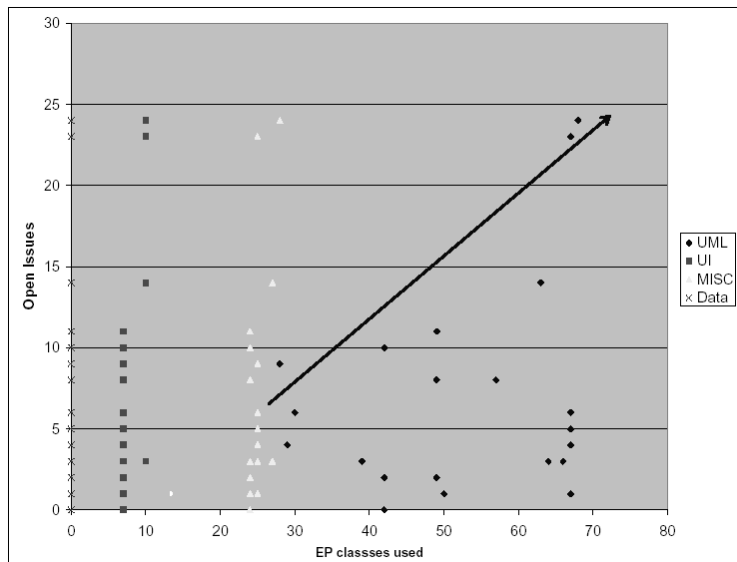
## Results – EP classes Vs Issues



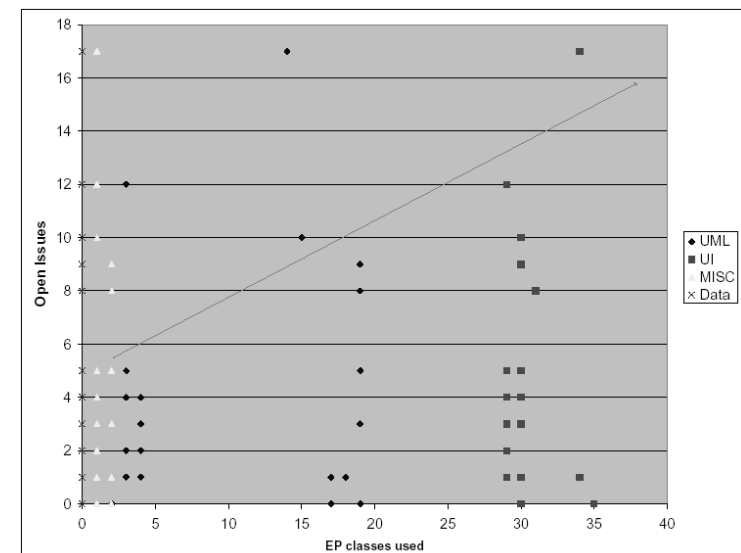
## EP classes used Vs Open Issues (Scaled)



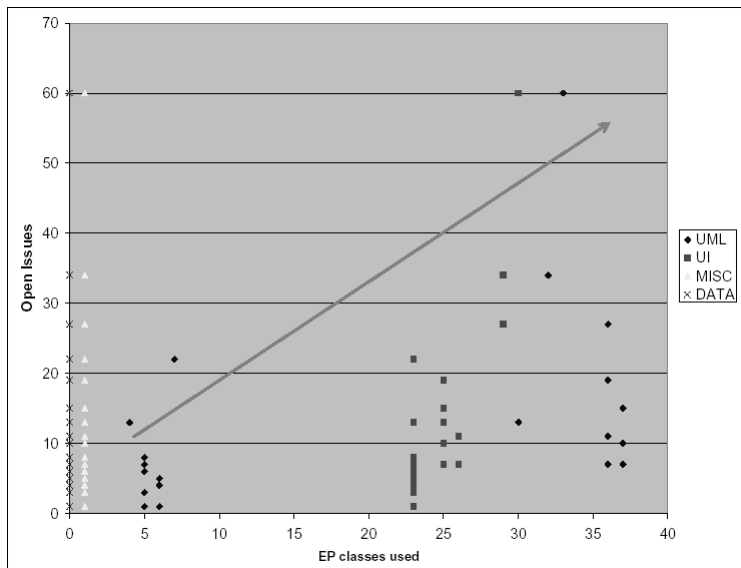
## Module - CodeGeneration



## Module – State/ActivityDiagram



# Module - ClassDiagram



## Conclusion

Suggestive hints about the hypothesis that more dependencies to external parameters increases the technical potential

For large scale study  
Need good 'examples' to study  
Issues with data available from open source systems  
Systems studied should exhibit well documented interesting design changes  
Need a disciplined approach to record pertinent data in software repositories and databases

A modified version of paper has been accepted for EDSE-7, ICSE05