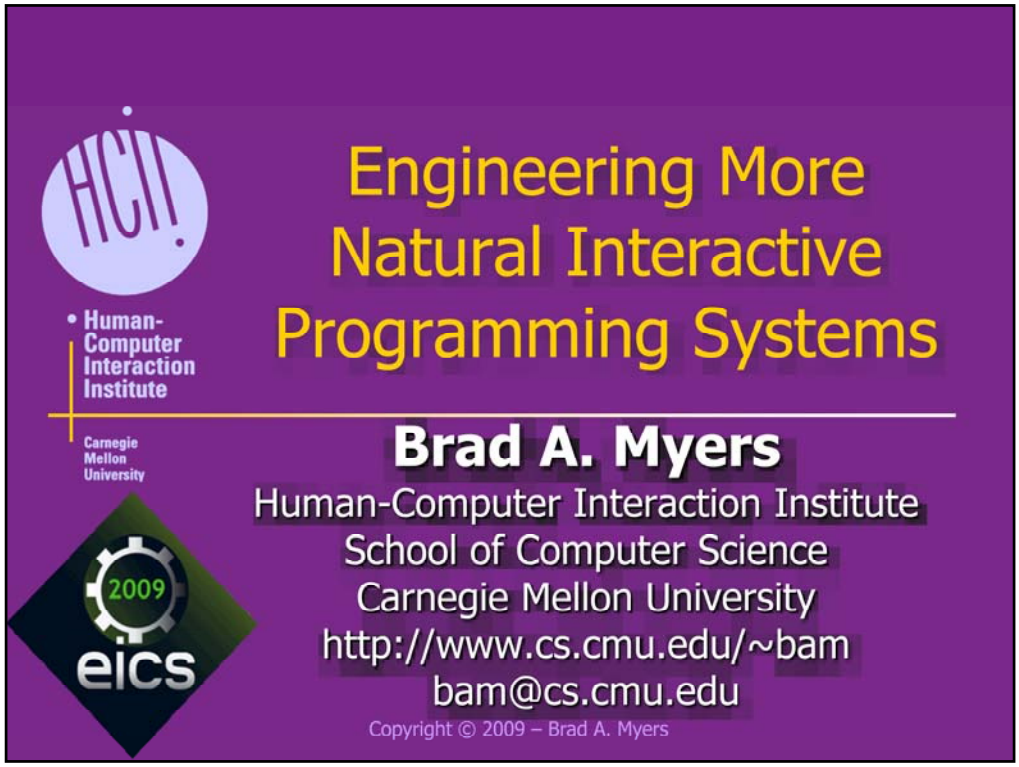


About 50 minutes

Engineering

- **Engineering** is the discipline, art and profession of acquiring and applying technical, scientific and mathematical knowledge to design and implement materials, structures, machines, devices, systems, and processes that safely realize a desired objective or inventions.

– Wikipedia

http://en.wikipedia.org/wiki/Engineering

Engineering

- **Engineering** is the discipline, art and profession of acquiring and applying technical, scientific and mathematical knowledge to design and implement materials, structures, machines, devices, systems, and processes that safely realize a desired objective or inventions.

  – Wikipedia

http://en.wikipedia.org/wiki/Engineering

Interactive Programming Systems

- Focus on systems for developers:
  - Programming languages themselves
  - Application programming interfaces (APIs) such as Toolkits, Libraries, Software Development Kits (SDKs), Frameworks, etc.
  - Integrated Development Environments (IDEs)
    - Writing code
    - Debugging code
  - Documentation tools

Why Programmers?

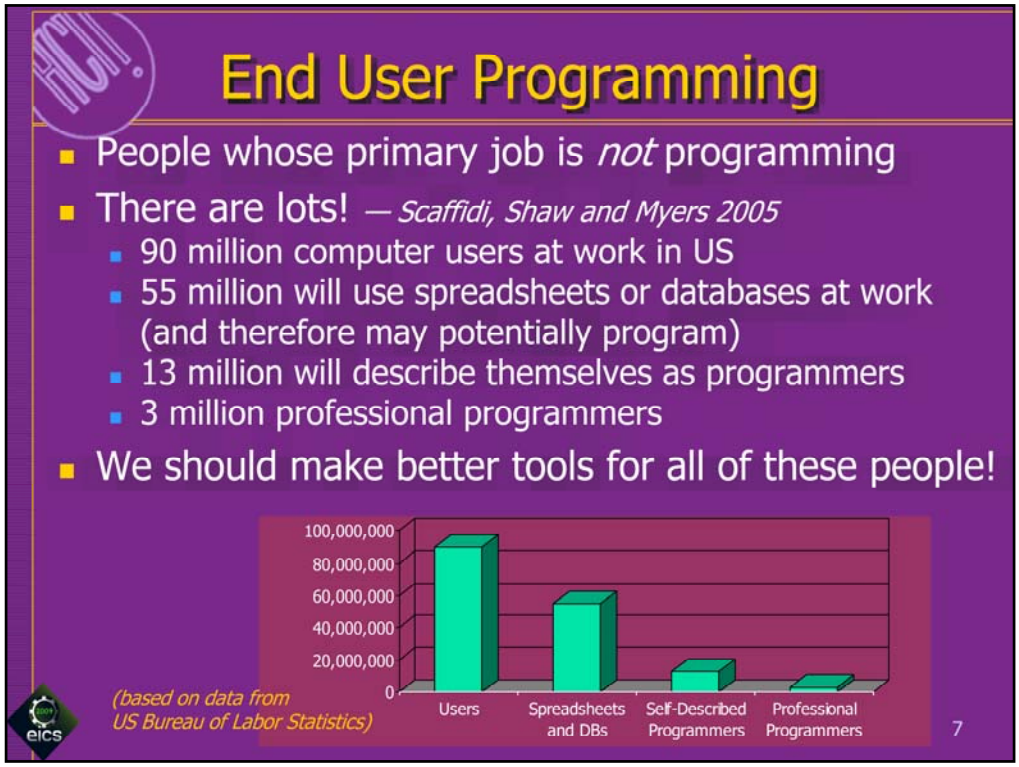
- By 2012, 30% of new jobs may require programming skills — *US Department of Labor*
- But programming and learning to program have only become more complex, with increasingly complex APIs
- Increasing demand for programmers to be efficient and create correct code

What Knowledge?

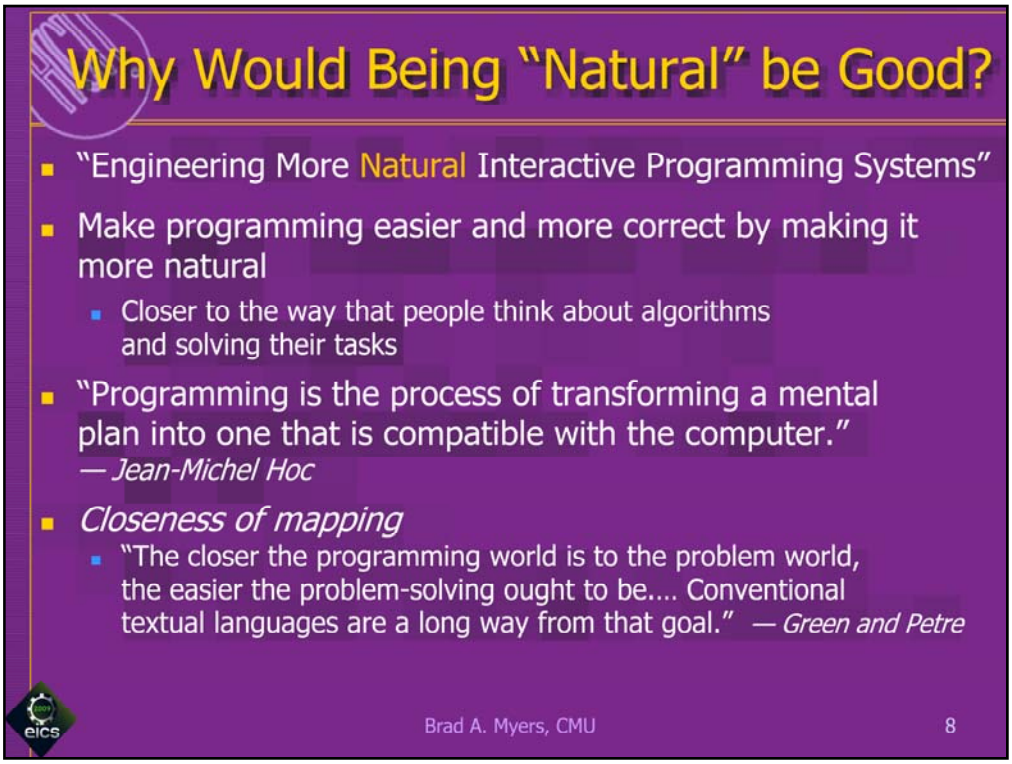
- Engineering = "acquiring and applying … knowledge to design and implement … systems"
- Radical notion:

Treating programmers as if they are people!
Gain knowledge from studying them!

- Use proven human-computer interaction (HCI) methods to study programmers
  - Professional programmers
  - Novice programmers
  - End-User Programmers (EUP)



Scaffidi, C., Shaw, M., and Myers, B. “Estimating the Numbers of End Users and End User Programmers,” in *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC'05). 2005. Dallas, Texas: pp. 207-214.*

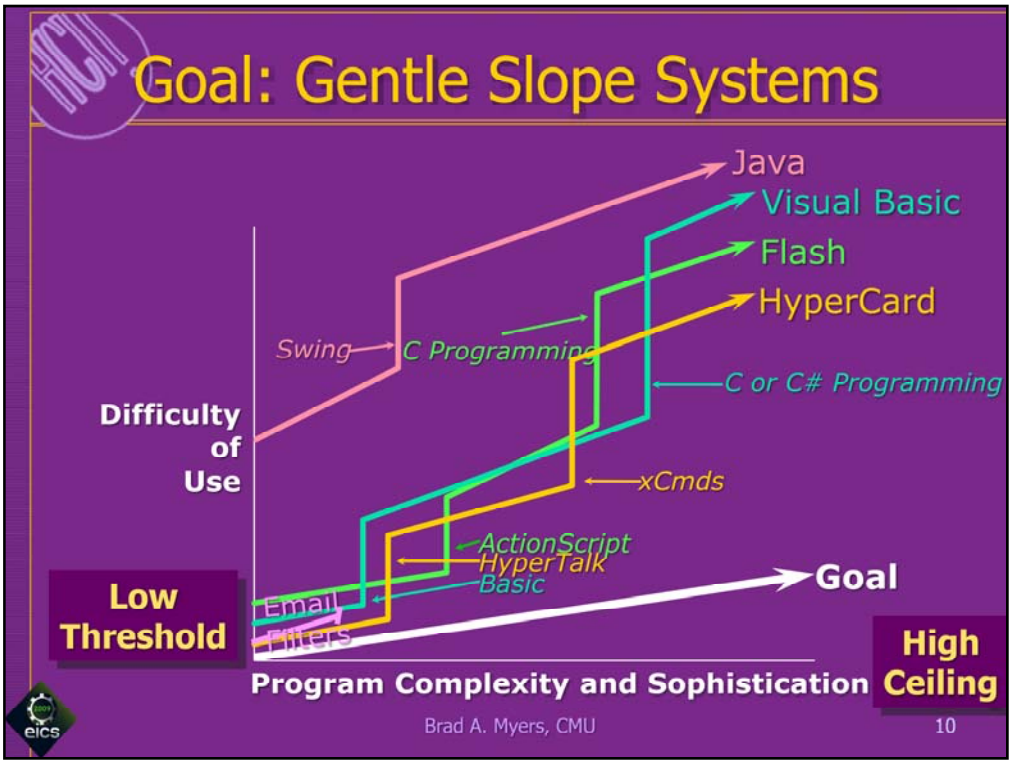


Hoc, J.-M. and Nguyen-Xuan, A. Language semantics, mental models and analogy. J.-M. Hoc et al., Eds. *Psychology of Programming. Academic* Press. London, 1990, 139–156

Green, T.R.G. and M. Petre, *Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework.* Journal of Visual Languages and Computing, 1996. **7**(2): p. 131-174.

Java

Visual Basic

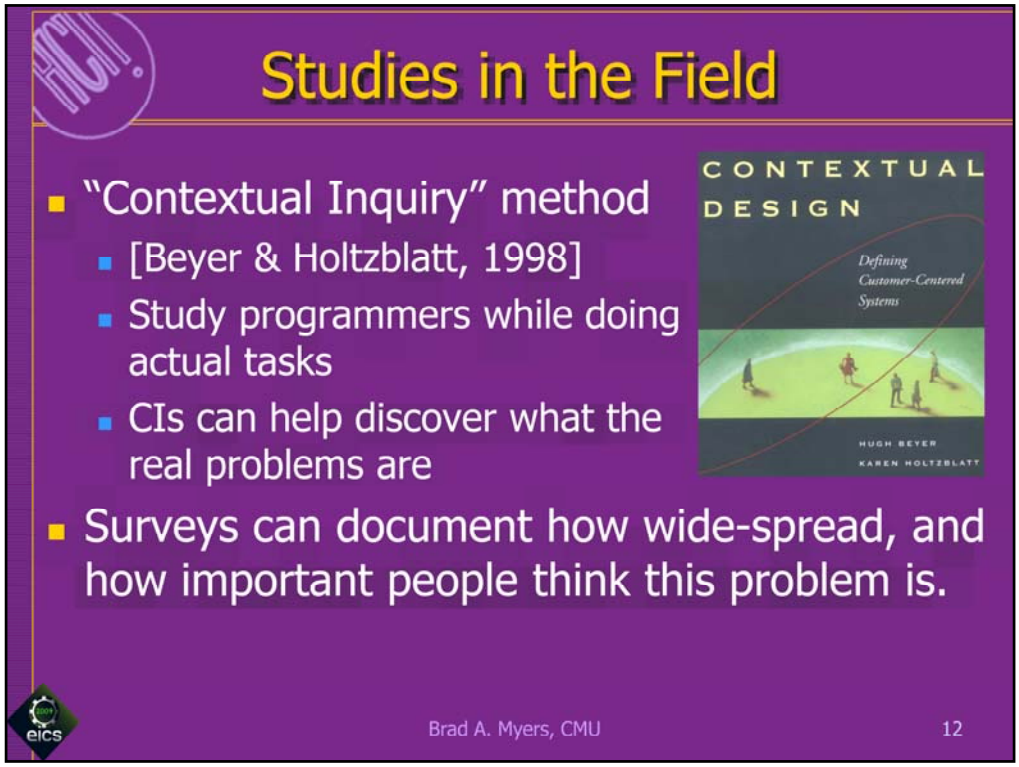


Originally appeared in:

Myers, B.A., Smith, D.C., and Horn, B. “Report of the `End-User Programming' Working Group,” in *Languages for Developing User Interfaces. 1992. Boston, MA: Jones and Bartlett. pp. 343-366.*

HCI Techniques We Have Used

- Discovering the real problems:
  - Contextual Inquiry
  - Surveys
  - Lab studies of existing systems
- Create models from results
- Prototypes of novel approaches
- Lab studies of the prototypes
- Can get *real measurements* for the effectiveness of the tools (=Engineering!)
  - Don't have to rely just on intuition

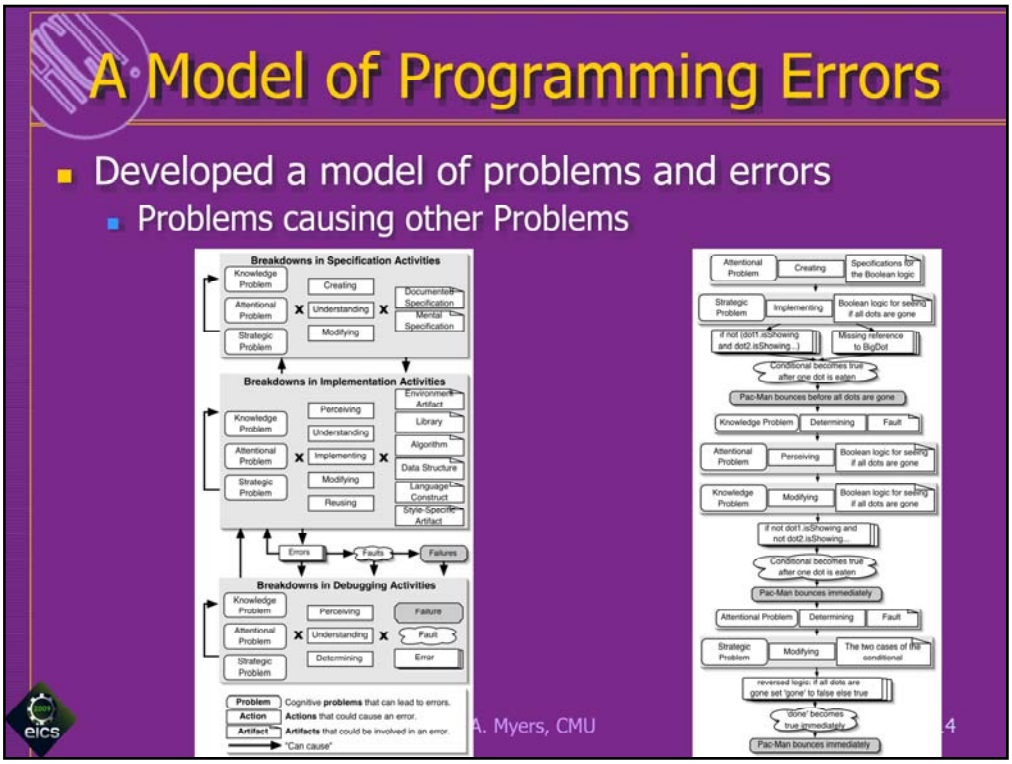


Beyer, H. and Holtzblatt, K., *Contextual Design: Defining Custom-Centered Systems. 1998, San Francisco, CA: Morgan Kaufmann Publishers, Inc.*

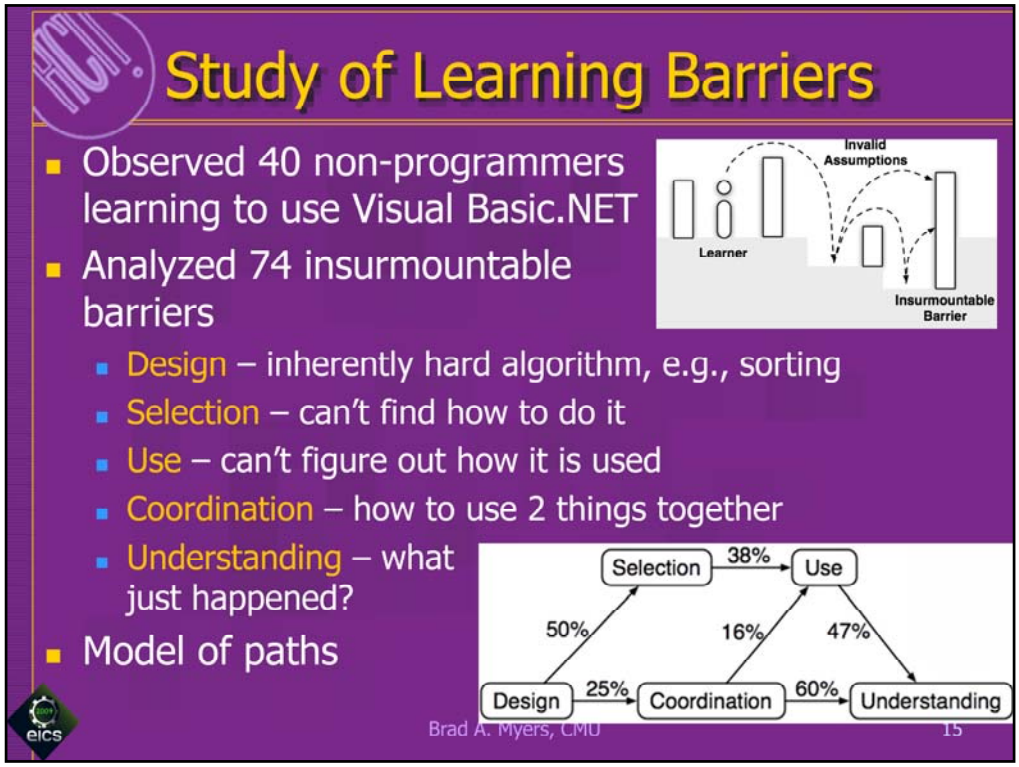
Left out: Scaffidi study of information workers

Novices Programming in Alice

- Study of novice errors in programming
  - All of the observed debugging problems could be addressed by "Why" questions
    - 32% were "why did"; 68% were "Why didn't"
- Current debugging techniques require user to guess where bug is or where to look
  - Most of initial guesses are wrong, even for experts
  - Two-thirds of novices' bugs were inserted due to incorrect guesses

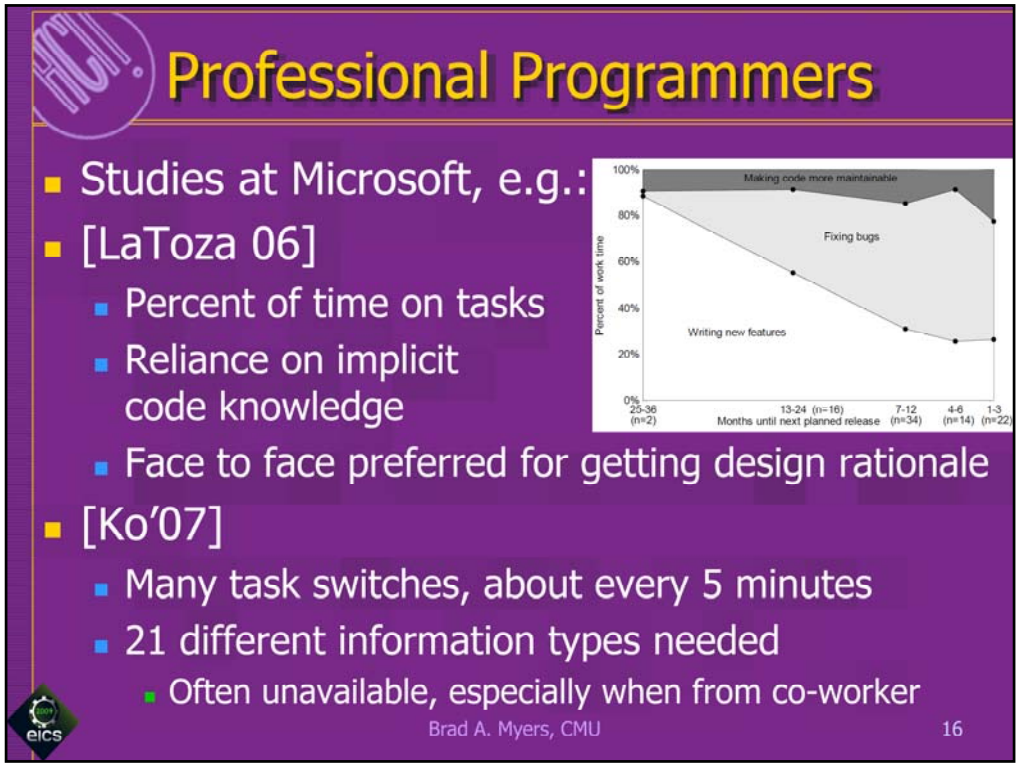


Ko, A.J. and Myers, B.A. “Development and Evaluation of a Model of Programming Errors,” in *IEEE Symposium on End-User and Domain-Specific Programming (EUP'03), part of the IEEE Symposia on Human-Centric Computing Languages and Environments. 2003. Auckland, New Zealand: pp. 7-14.*



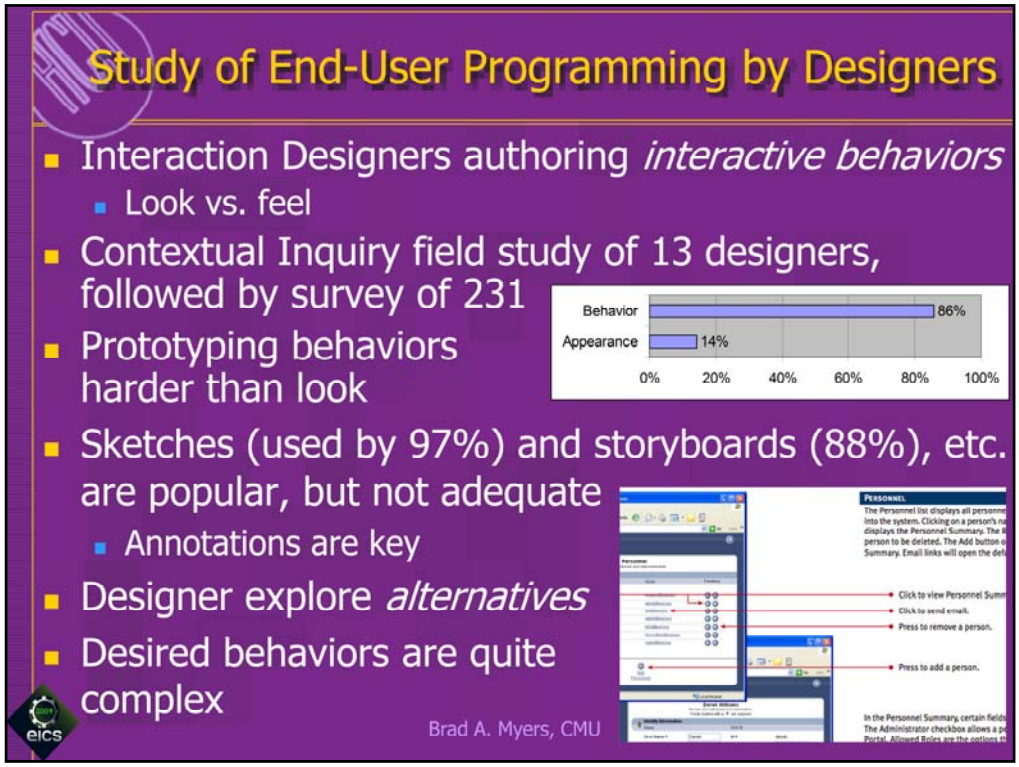
Ko, A.J., Myers, B.A., and Aung, H.H. "Six Learning Barriers in End-User Programming Systems," in *IEEE Symposium on Visual Languages and Human-Centric Computing. 2004. Rome, Italy: pp. 199-206.*

Model of invalid assumptions led to what other kinds of barriers. Don't include barriers where use stuck (insurmountable barriers)

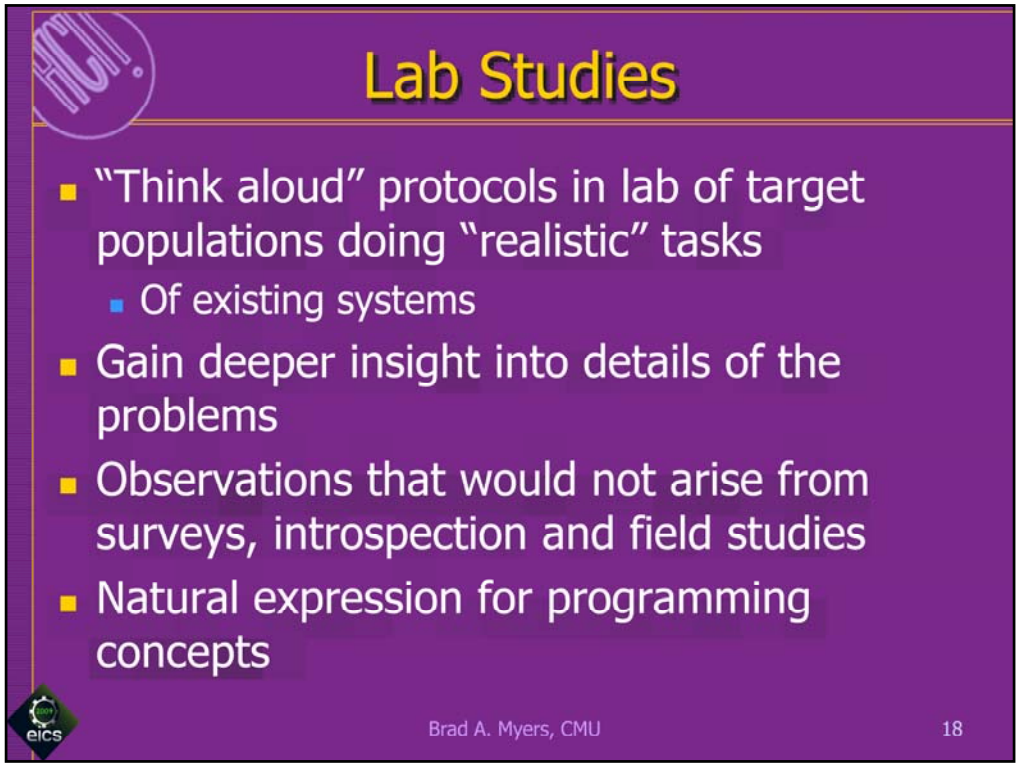


Ko, A.J., DeLine, R., and Venolia, G., "Information Needs in Collocated Software Development Teams," in *International Conference on Software Engineering (ICSE'2007), May 20-26, 2007. pp. 344-353.*

LaToza, T.D., Venolia, G., and DeLine, R. "Maintaining Mental Models: A Study of Developer Work Habits," in *Proceedings of the International Conference on Software Engineering (ICSE'2006). 2006. Shanghai, China: pp. 492 - 501.*



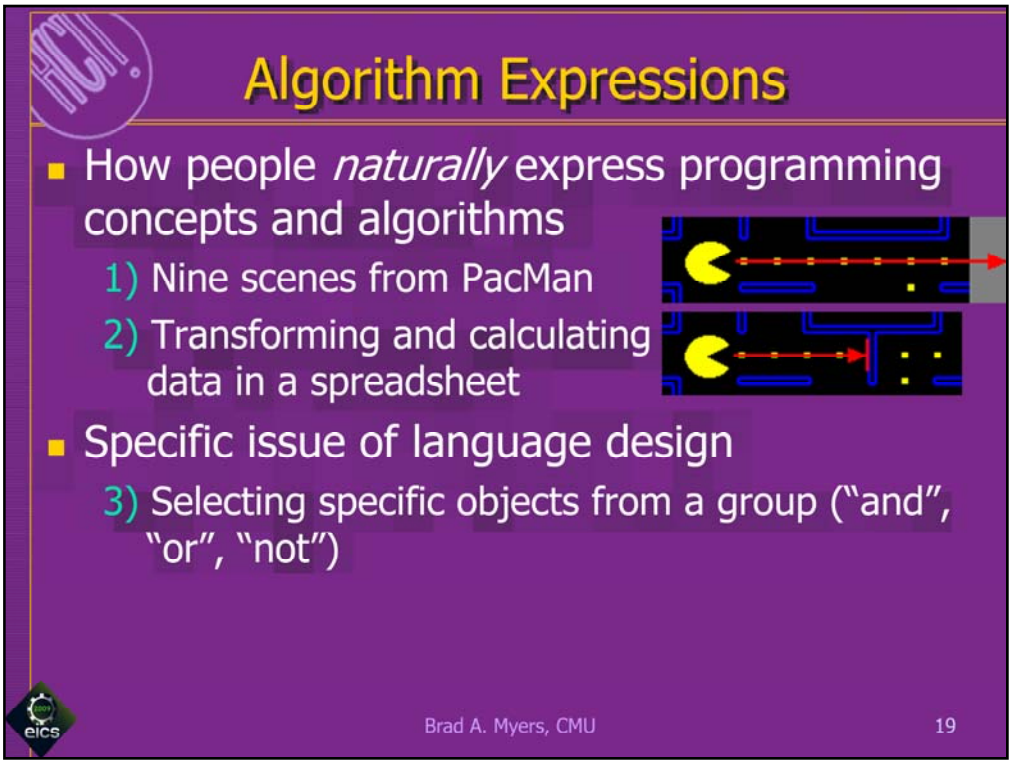
Myers, B., *et al., "How Designers Design and Program Interactive Behaviors," in 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC'08, Sept 15-18, 2008. Herrsching am Ammersee, Germany. pp. 185-188.*



Note: not of a new prototype system

Reference for Think-aloud:

Nielsen, J., *Usability Engineering. 1993, Boston: Academic Press.*



Pane, J.F., Ratanamahatana, C.A., and Myers, B.A., "Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems." *International Journal of Human-Computer Studies, 2001.* ***54(2): pp. 237-264.***

Examples of Results

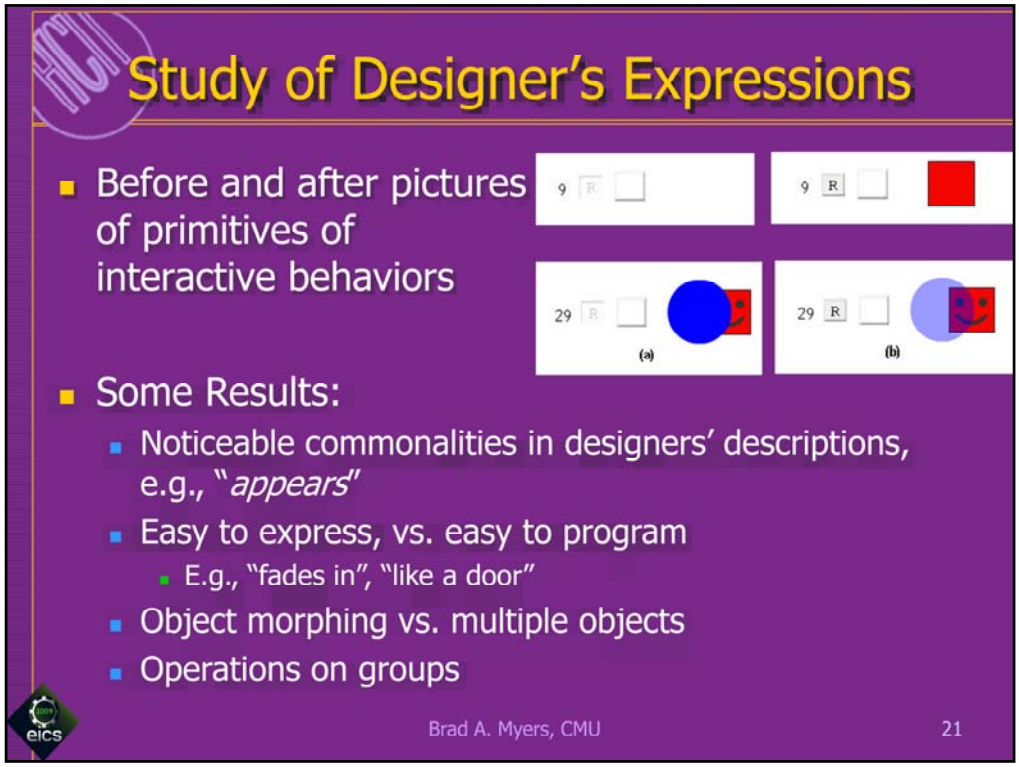
- Rule-based style

  "If PacMan loses all his lives, its game over."
- Set operations instead of iterations

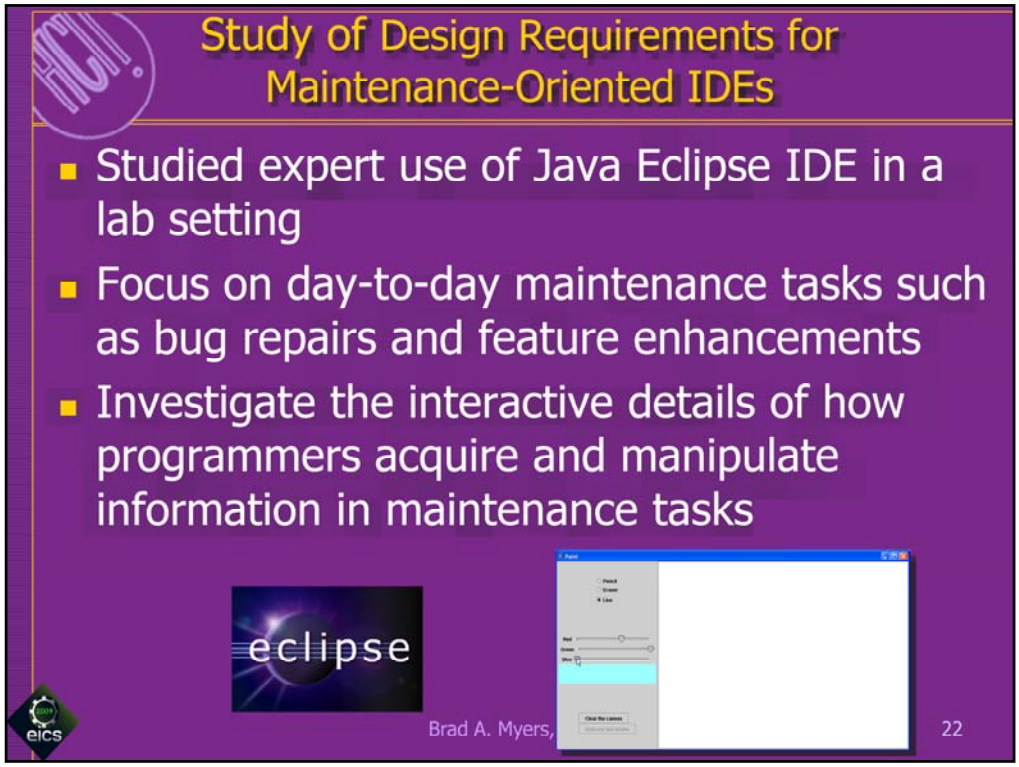
  "When PacMan eats all of the dots, he goes to the next level."
- "And", "Or", "Not" don't match computer interpretation
  - ... men *and* women, ...(*not* an apple)or pear
- Most arithmetic used natural language style

  "When PacMan eats a big dot, the score goes up 100."
- Operations suggest data as lists, not arrays
  - People don't make space before inserting
- Objects normally moving

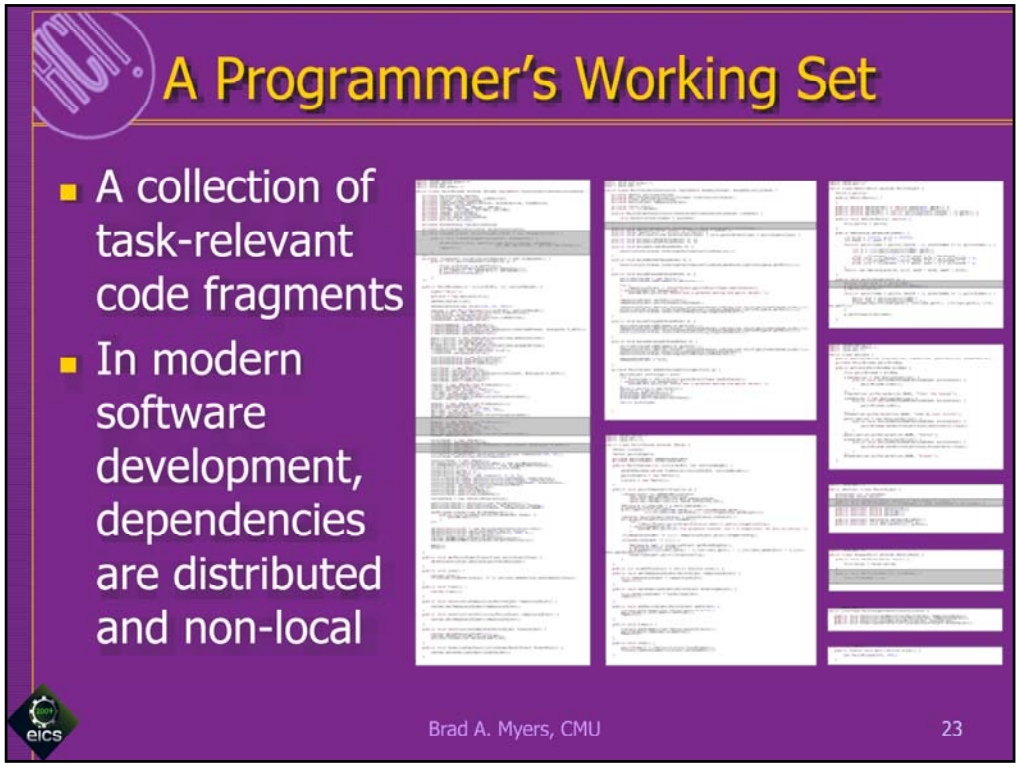
  "If PacMan hits a wall, he stops."
  - so objects remember their own state



Park, S., Myers, B., and Ko., A. "Designers' Natural Descriptions of Interactive Behaviors," in *2008 IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC'08. 2008. Herrsching am Ammersee, Germany: pp. 185-188.*



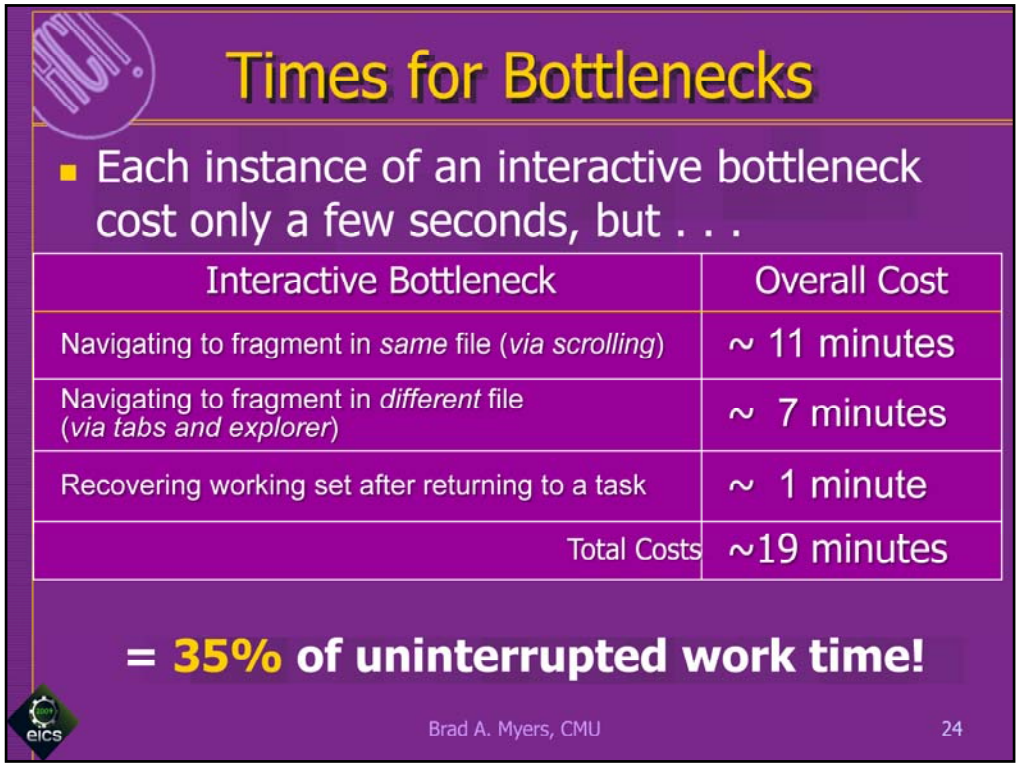
Ko, A.J., Aung, H.H., and Myers, B.A. “Eliciting Design Requirements for Maintenance-Oriented IDEs: A Detailed Study of Corrective and Perfective Maintenance Tasks,” in *ICSE'05: 27th International Conference on Software Engineering. 2005. St. Louis, MO: pp. 126-135.*



To better illustrate a working set, here is all of the code in the Paint program. The grey regions indicate the code that one programmer included in their working set as part of the stroke thickness task.

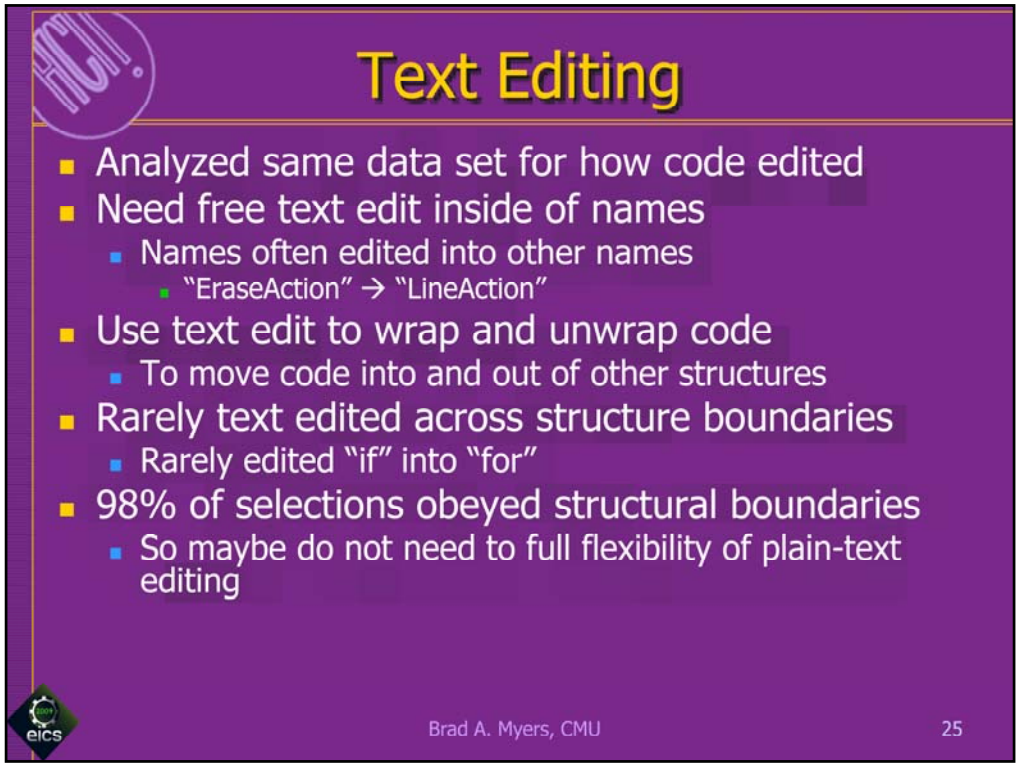
This makes sense: in modern software development, dependencies are distributed and non-local.

Given this idea of a working set, we'll now take a more detailed look at how programmers formed, represented, and navigated their working sets, and how the IDE helped or hindered them.

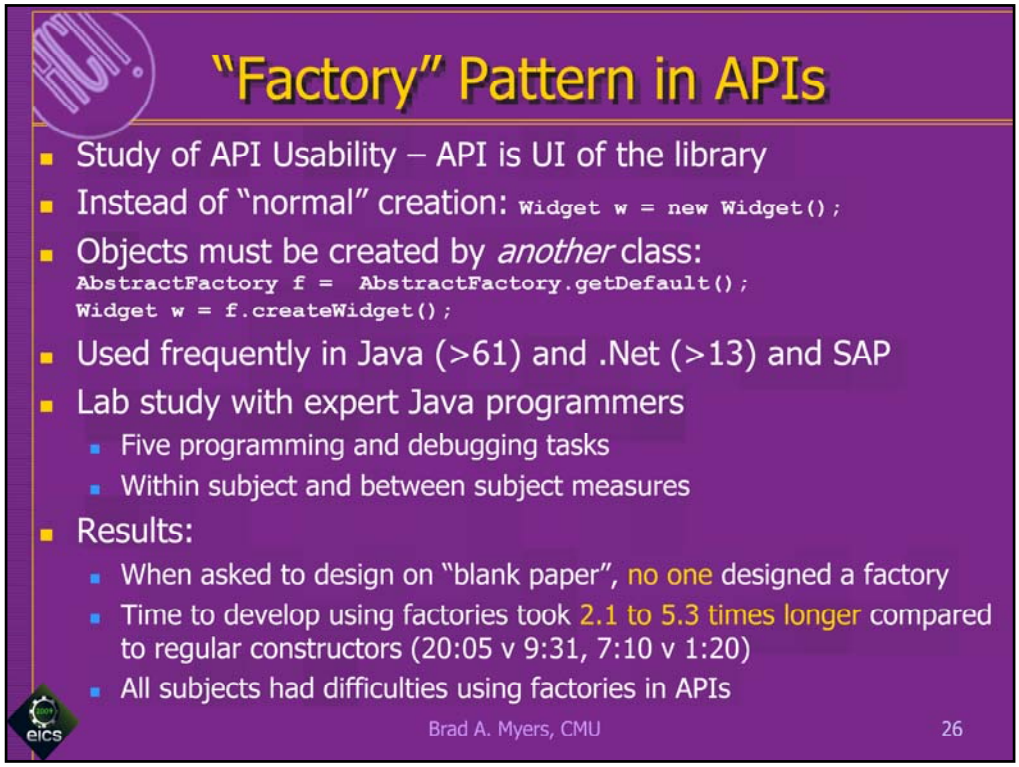


Ko, A. J., Myers, B. A., Coblenz, M., & Aung, H. H. (2006). An Exploratory Study of How Developers Seek, Relate, and Collect Relevant Information during Software Maintenance Tasks. *IEEE Transactions on Software Engineering, 33(12), 971-987.*

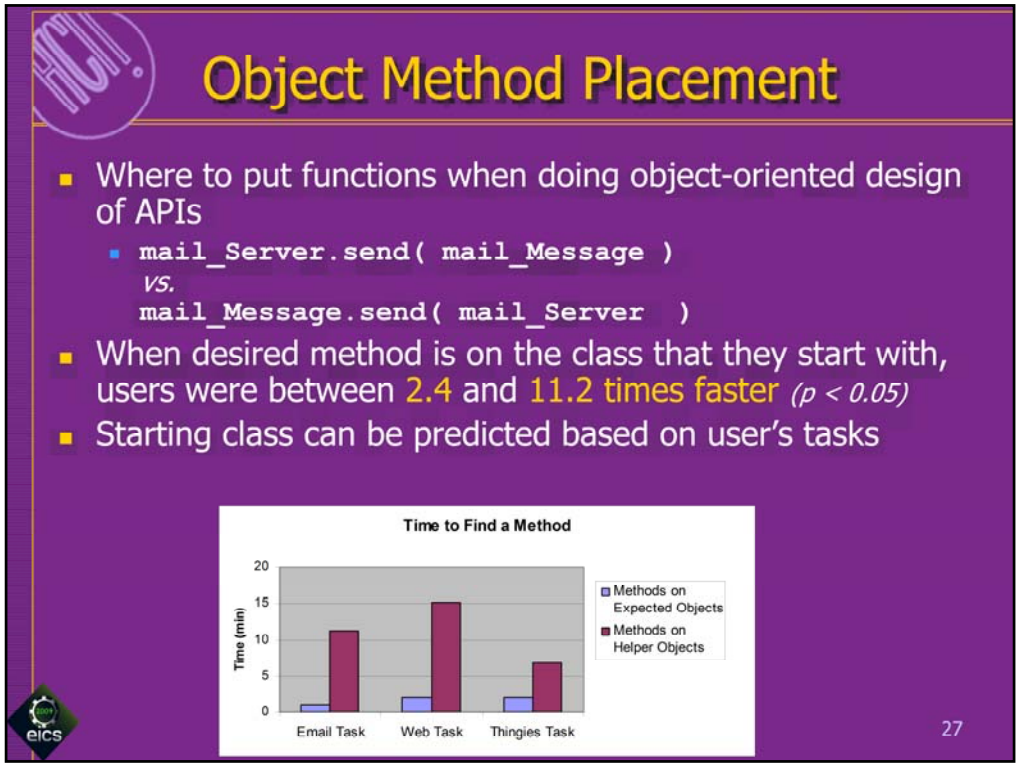
http://www.cs.cmu.edu/~ajko/papers/Ko2006SeekRelateCollect.pdf



Ko, A. J., & Myers, B. A. (2006). *Barista: An Implementation Framework for Enabling New Tools, Interaction Techniques and Views in Code Editors. Paper presented at the Proceedings CHI'2006: Human Factors in Computing Systems, Montreal, Canada, 387-396.*



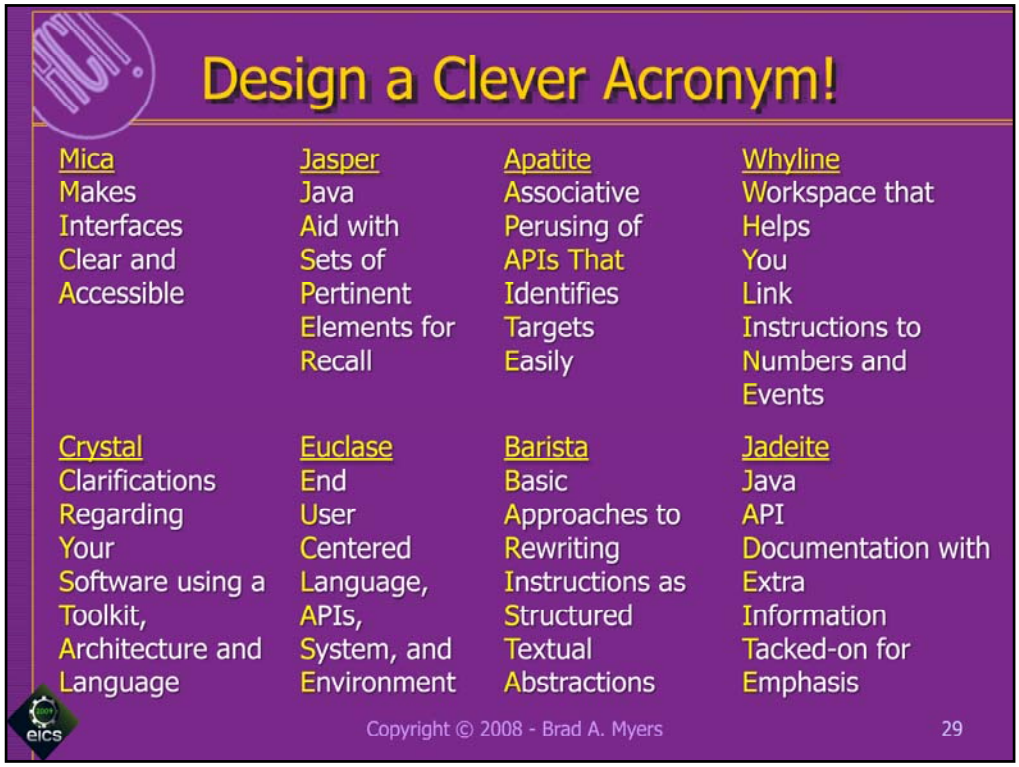
Ellis, B., Stylos, J., & Myers, B. (2007). *The Factory Pattern in API Design: A Usability Evaluation. Paper presented at the International Conference on Software Engineering (ICSE'2007), Minneapolis, MN, 302-312.*



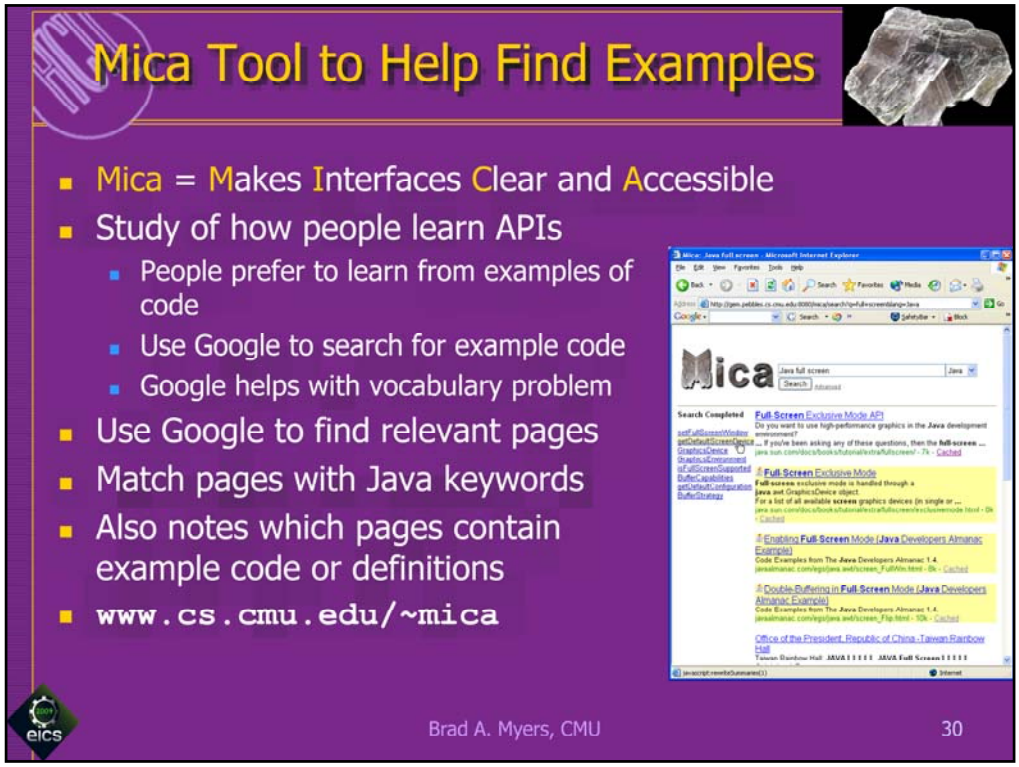
Stylos, J., & Myers., B. A. (2008). *The Implications of Method Placement on API Learnability. Paper presented at the Sixteenth ACM SIGSOFT Symposium on Foundations of Software Engineering (FSE 2008), Atlanta, GA, 105-112.*

Create & Test Prototypes

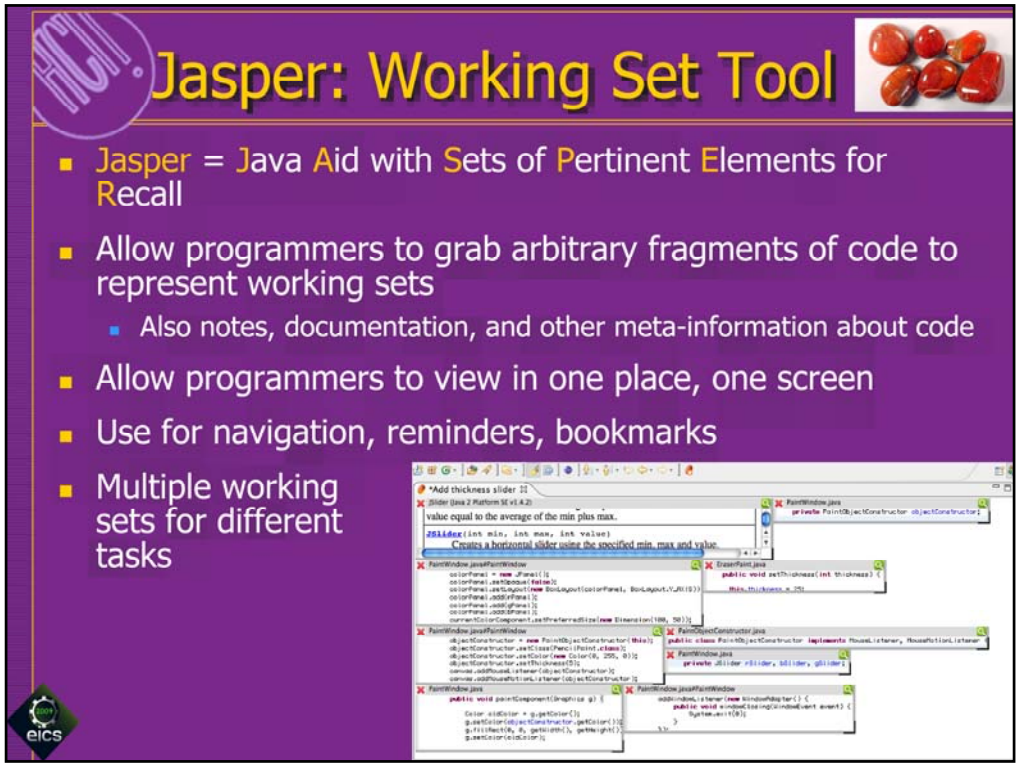
- Address the observed barriers
- More likely to solve real problems
- Test the prototypes in user studies
  - Measure impact compared to existing situation
- But most importantly first…



See http://www.cs.cmu.edu/~bam/acronyms.html



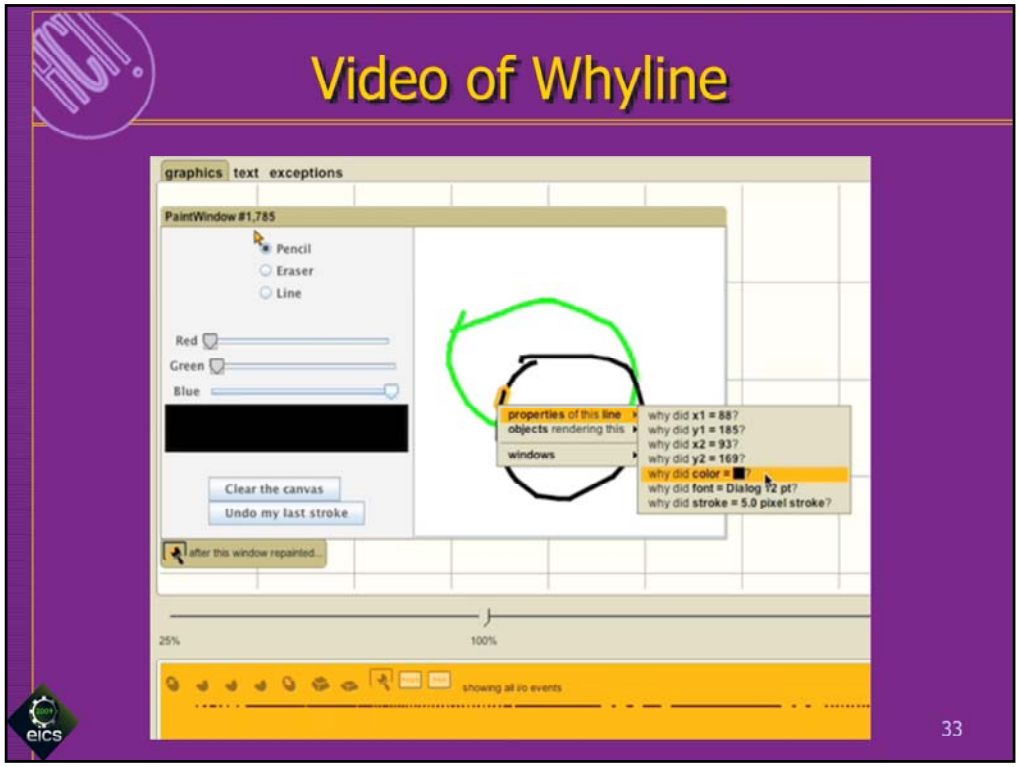
Stylos, J. and Myers, B.A. “Mica: A Programming Web-Search Aid,” in *IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC'06. 2006. Brighton, UK: pp. 195-202.*



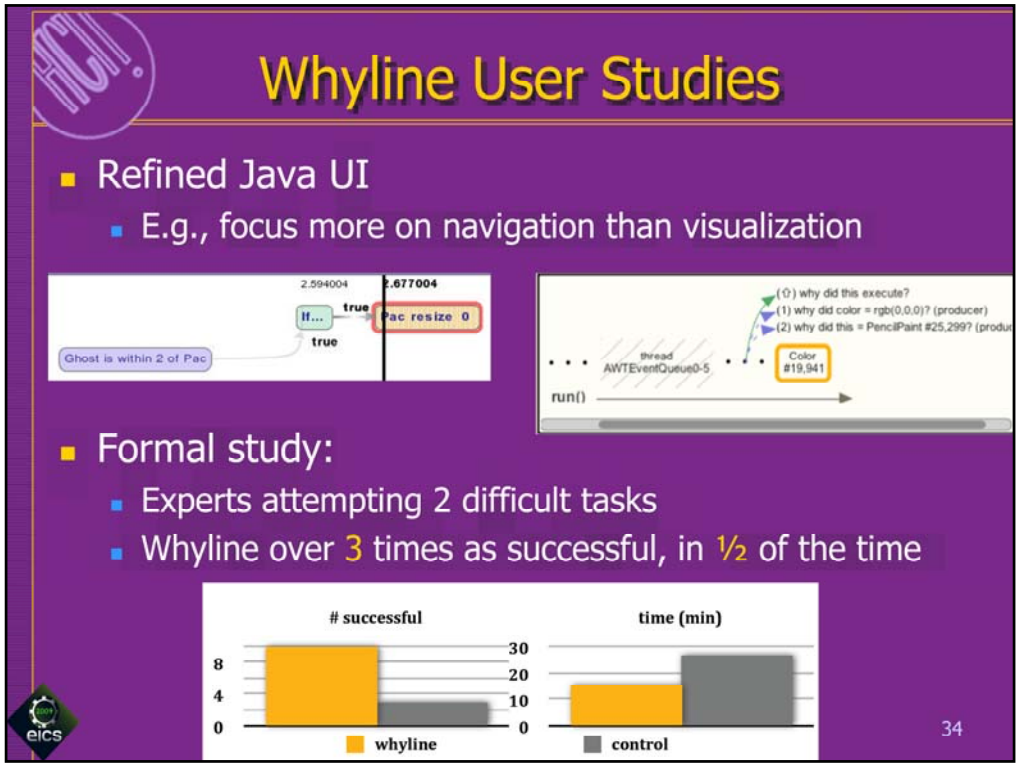
Coblenz, M.J., Ko, A.J., and Myers, B.A. “JASPER: An Eclipse Plug-In to Facilitate Software Maintenance Tasks,” in *Eclipse Technology eXchange (ETX) Workshop at OOPSLA 2006. 2006. Portland, Oregon: pp. 65-69.*

Whyline

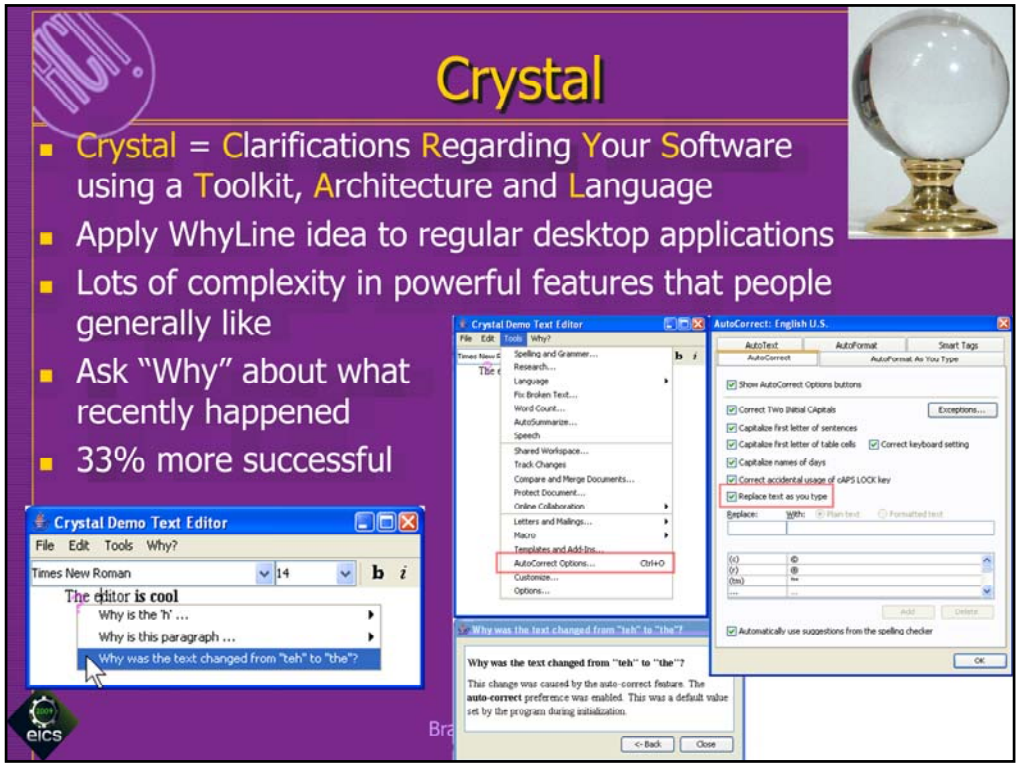
- Allow users to directly ask "Why" and "Why not"
- Collect a trace and replay
- Whyline = Workspace that Helps You Link Instructions to Numbers and Events
- Ask about specific output that is in error
  - Accurate about what is wrong, but not when guess why
- Visualization of combined control and data flow
- Highlighting of relevant parts of the code



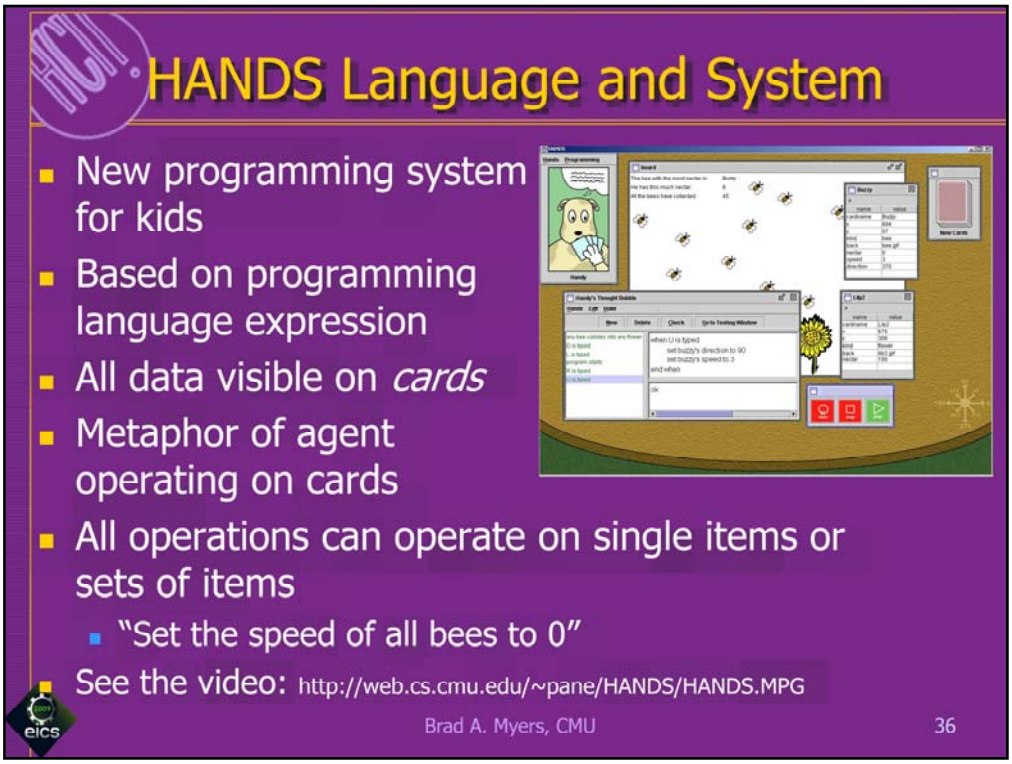
Ko, A.J. and Myers, B.A., “Debugging Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior,” in *ICSE'2008: 30th International Conference on Software EngineeringMay 10 - 18, 2008. Leipzig, Germany. pp. 301-310.* ***Winner, Distinguished Paper Award.***



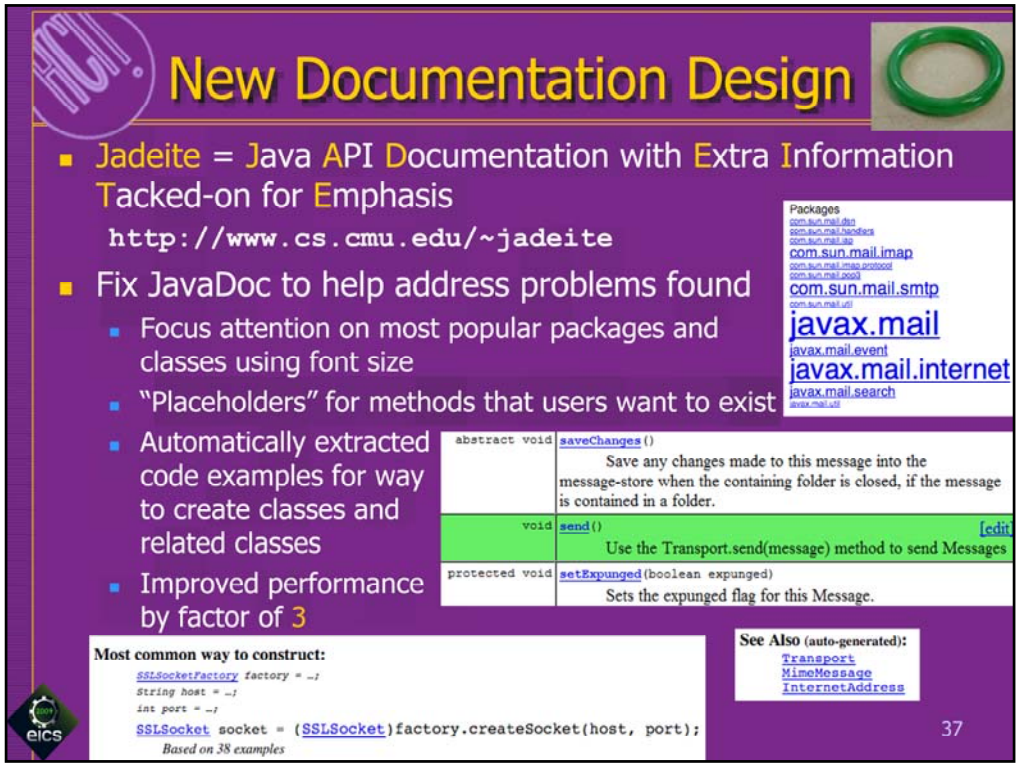
Ko, A.J. and Myers, B.A., “Debugging Reinvented: Asking and Answering Why and Why Not Questions about Program Behavior,” in *ICSE'2008: 30th International Conference on Software EngineeringMay 10 - 18, 2008. Leipzig, Germany. pp. 301-310.* ***Winner, Distinguished Paper Award.***



Myers, B., *et al. "Answering Why and Why Not Questions in User Interfaces," in Proceedings CHI'2006: Human Factors in Computing Systems. 2006. Montreal, Canada: pp. 397-406.*



Pane, J.F. and Myers, B.A. "The Impact of Human-Centered Features on the Usability of a Programming System for Children," in *Human Factors in Computing Systems. 2002. Minneapolis, MN: pp. 684-685. http://www.cs.cmu.edu/~pane/research.html. Extended Abstracts for CHI'2002.*

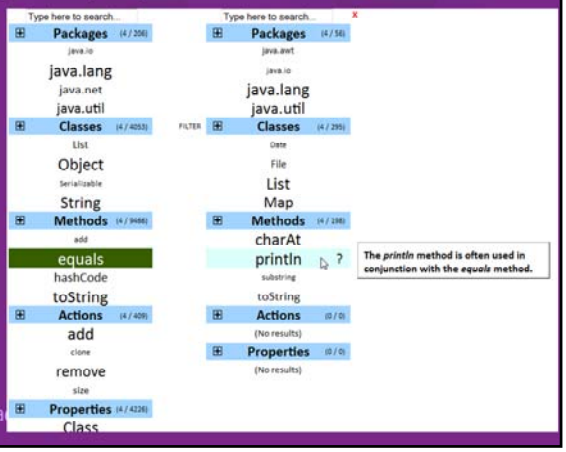


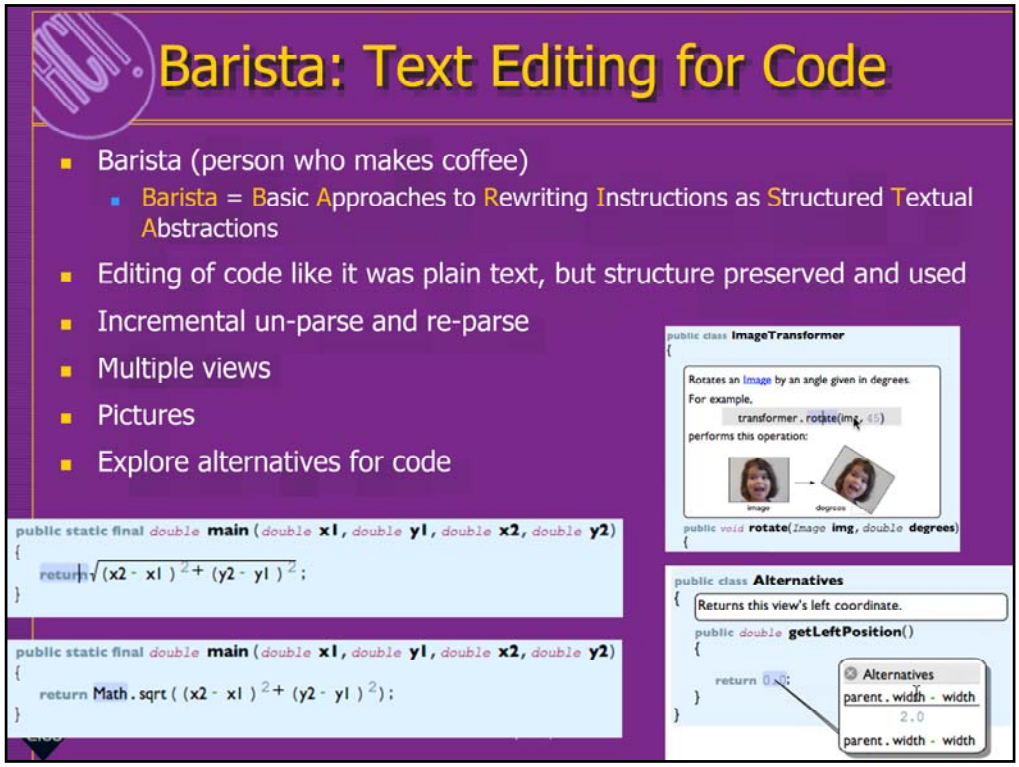
Stylos, J., Faulring, A., Yang, Z., and Myers, B.A., "Improving API Documentation Using API Usage Information," in *VL/HCC'09: IEEE Symposium on Visual Languages and Human-Centric ComputingSept. 20-24, 2009. Corvallis, Oregon. To appear.*

Apatite Documentation Tool

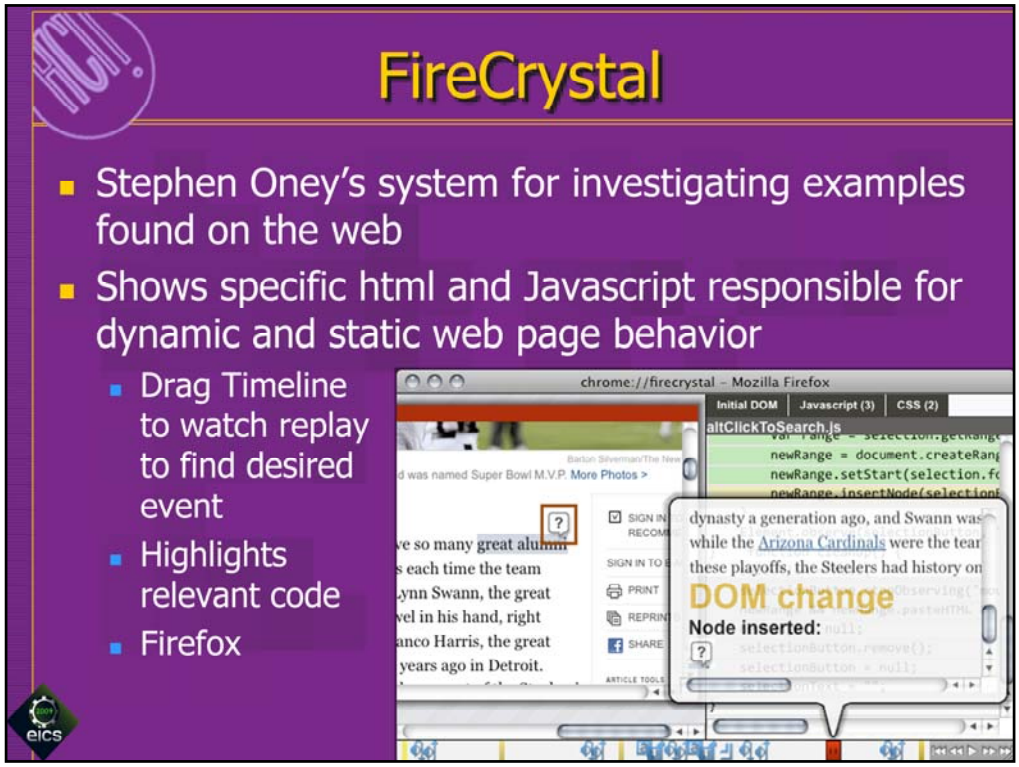
- Apatite = Associative Perusing of APIs That Identifies Targets Easily

  `http://www.cs.cmu.edu/~apatite`
- Start with verbs (actions) and properties and find what classes implement them
- Find things associated with other things
  - E.g., classes that are often used together
  - Classes that implement or are used by a method

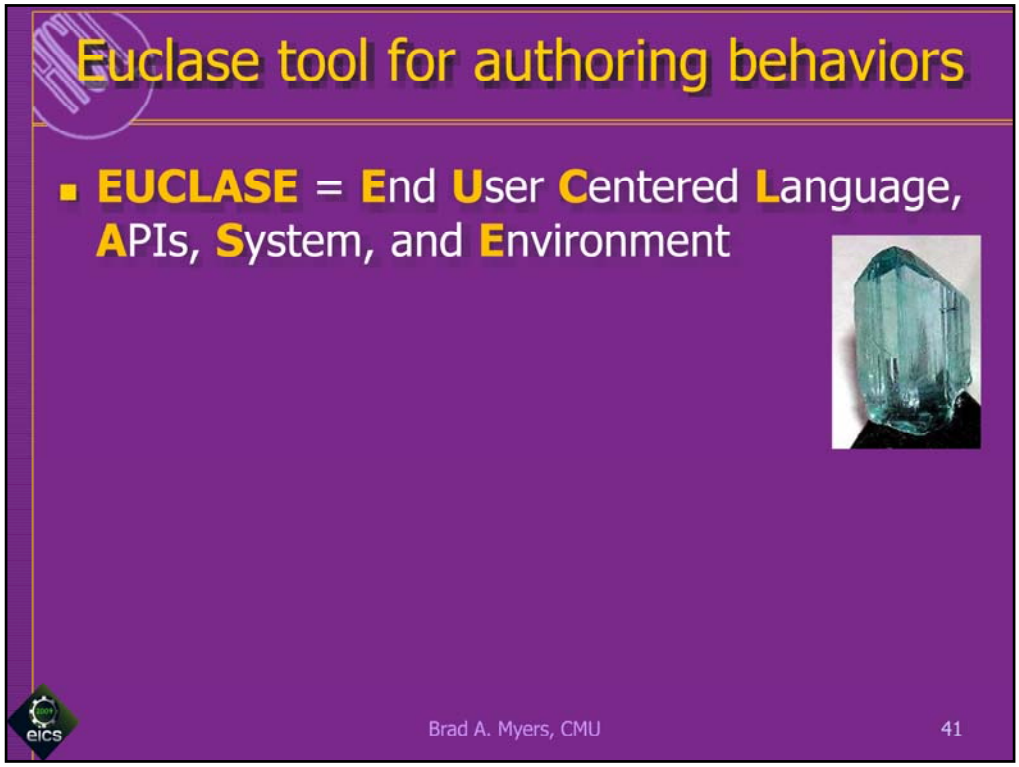




Ko, A.J. and Myers, B.A. “Barista: An Implementation Framework for Enabling New Tools, Interaction Techniques and Views in Code Editors,” in *Proceedings CHI'2006: Human Factors in Computing Systems. 2006. Montreal, Canada: pp. 387-396.*



Oney, S. and Myers, B., "FireCrystal: Understanding Interactive Behaviors in Dynamic Web Pages," in *IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC'09, Sept. 20-24, 2009. Corvallis, Oregon. To appear.*



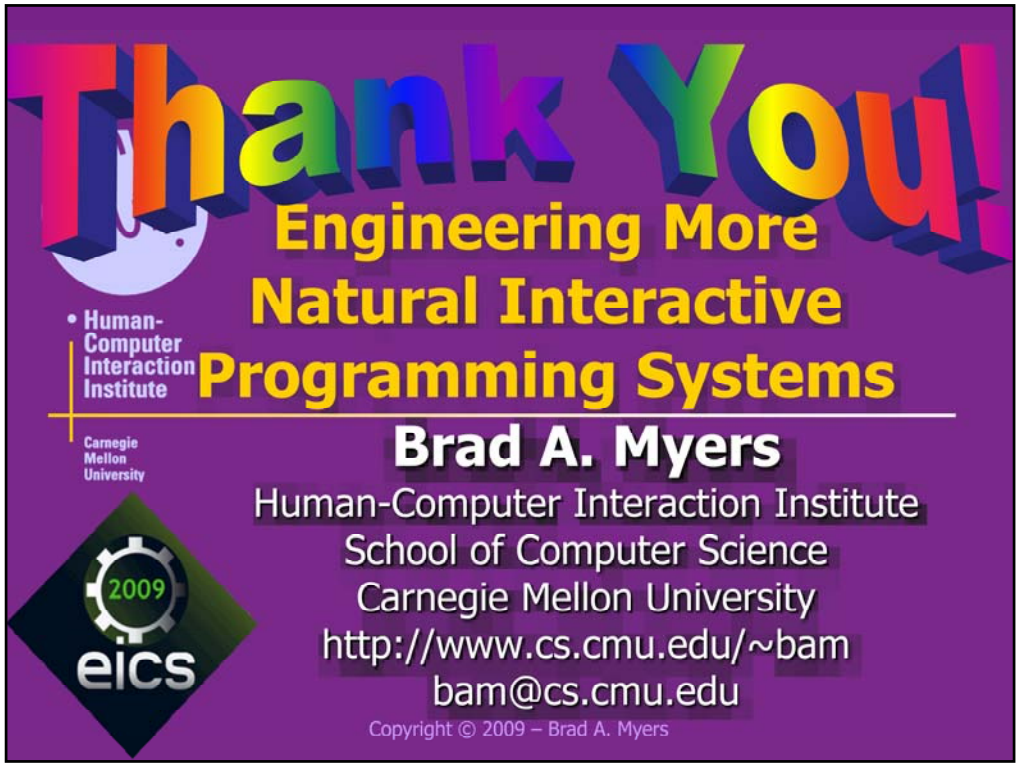
Have name, need system

Conclusions

- HCI techniques can provide significant new knowledge about developers
- Use this knowledge to guide new systems
- HCI techniques can provide evidence that systems work

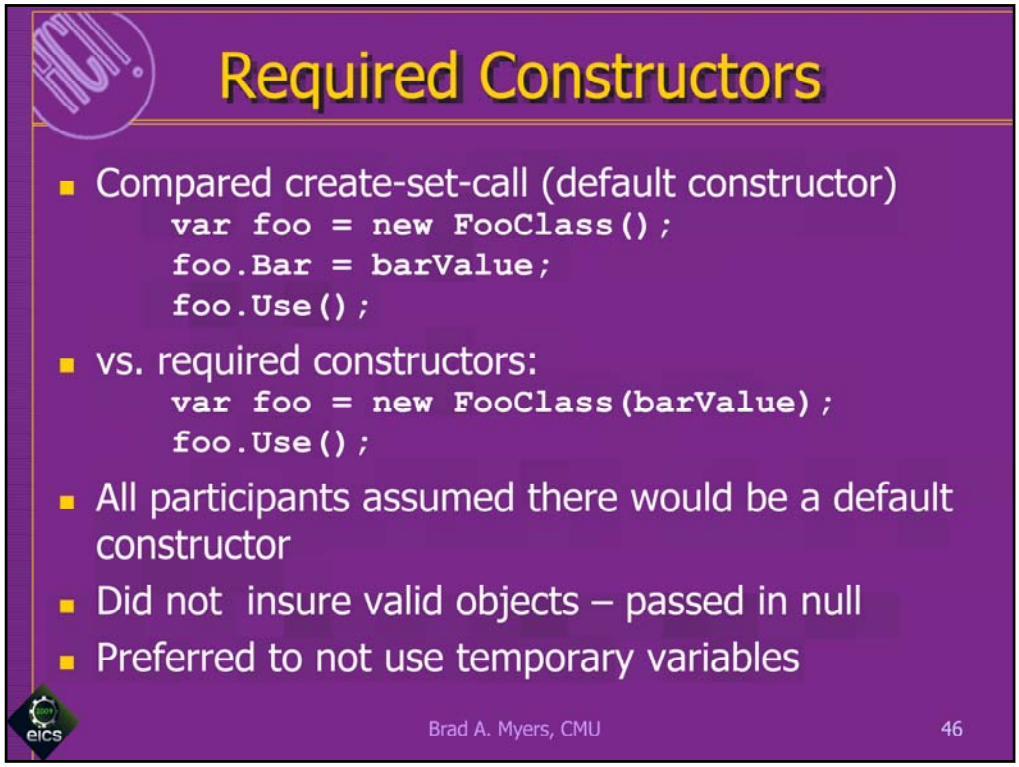
Thanks to:

- Funding:
  - NSF under IIS-0329090, CCF-0811610 and IIS-0757511 (Creative-IT)
  - NSF ITR CCR-0324770 as part of the EUSES Consortium
  - SAP
  - Adobe
  - IBM
- Many students:
  - Htet Htet Aung
  - Jack Beaton
  - Ruben Carbonell
  - John R. Chang
  - Polo Chau
  - Luis J. Cota
  - Michael Coblenz
  - Dan Eisenberg
  - Brian Ellis
  - Andrew Faulring
  - Aristiwidya B. (Ika) Hardjanto
  - Sae Young (Sophie) Jeong
  - Andy Ko
  - Thomas LaToza
  - Joonhwan Lee
  - Leah Miller
  - Mathew Mooty
  - Gregory Mueller
  - Yoko Nakano
  - Stephen Oney
  - John Pane
  - Sunyoung Park
  - Chotirat (Ann) Ratanamahatana
  - Christopher Scaffidi
  - Jeff Stylos
  - David A. Weitzman
  - Yingyu (Clare) Xie
  - Zizhuang (Zizzy) Yang

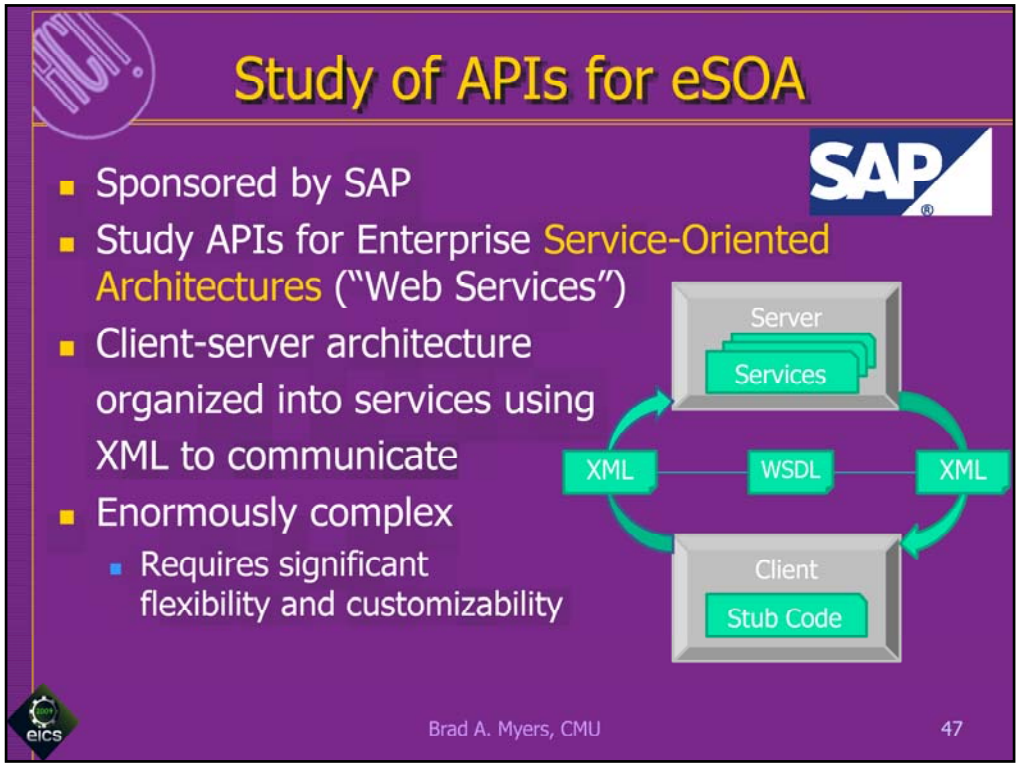


About 65 minutes

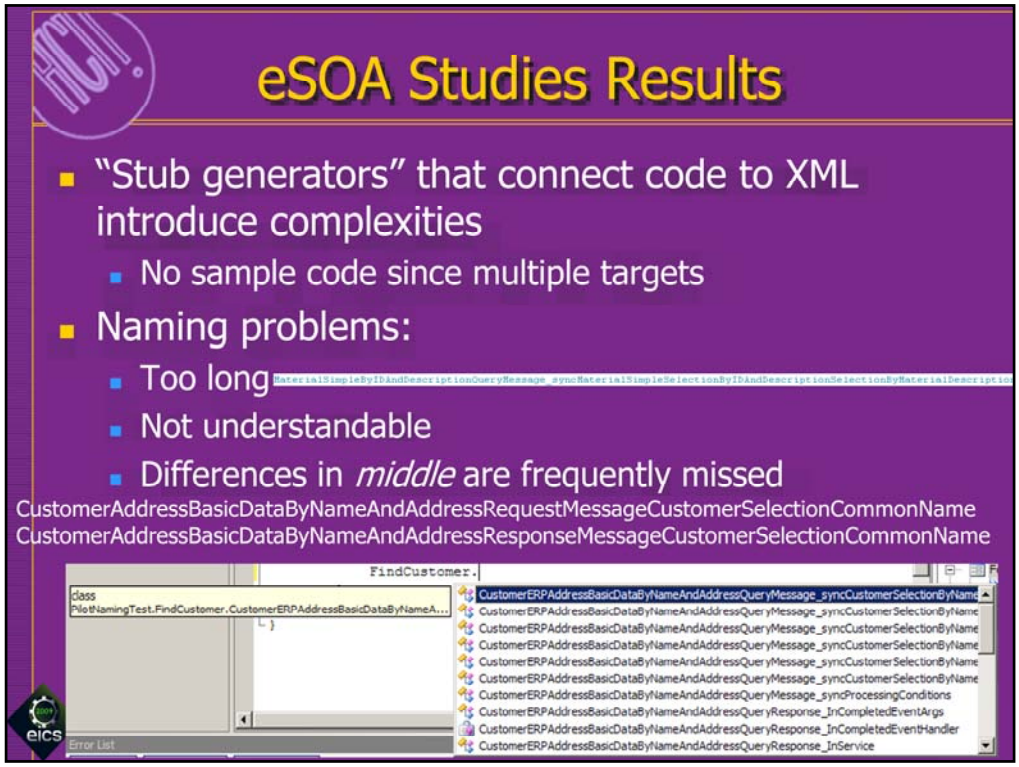
SLIDES NOT USED – NOT ENOUGH TIME



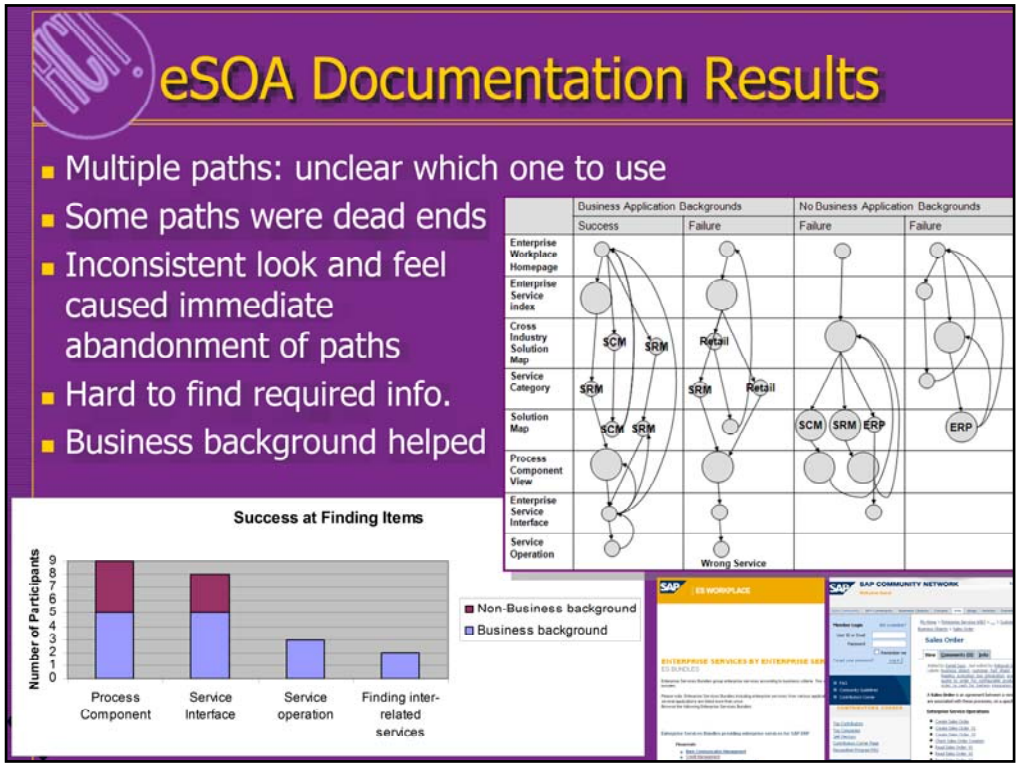
Stylos, J. and Clarke, S. "Usability Implications of Requiring Parameters in Objects' Constructors," in *International Conference on Software Engineering (ICSE'2007). 2007. Minneapolis, MN: pp. 529-539.*



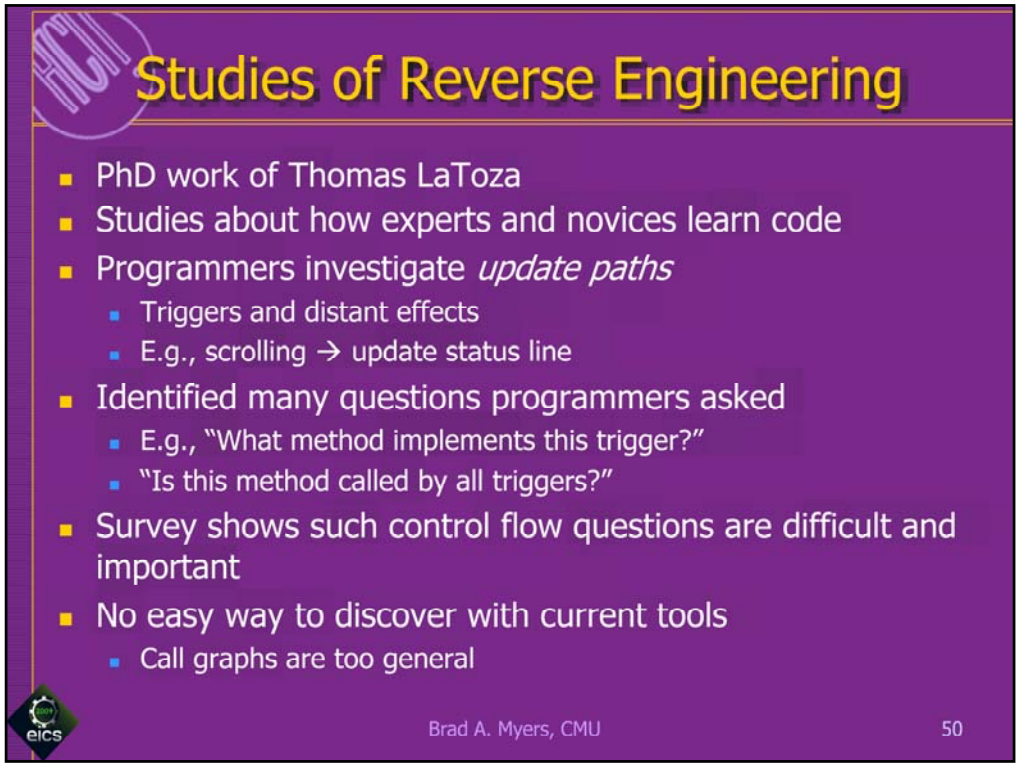
Beaton, J., Myers, B.A., Stylos, J., Jeong, S.Y., and Xie, Y. "Usability Evaluation for Enterprise SOA APIs," in *2nd International Workshop on Systems Development in SOA Environments, SDSOA 2008 (Co-located with ICSE 2008). 2008. Leipzig, Germany: pp. 29-34.*



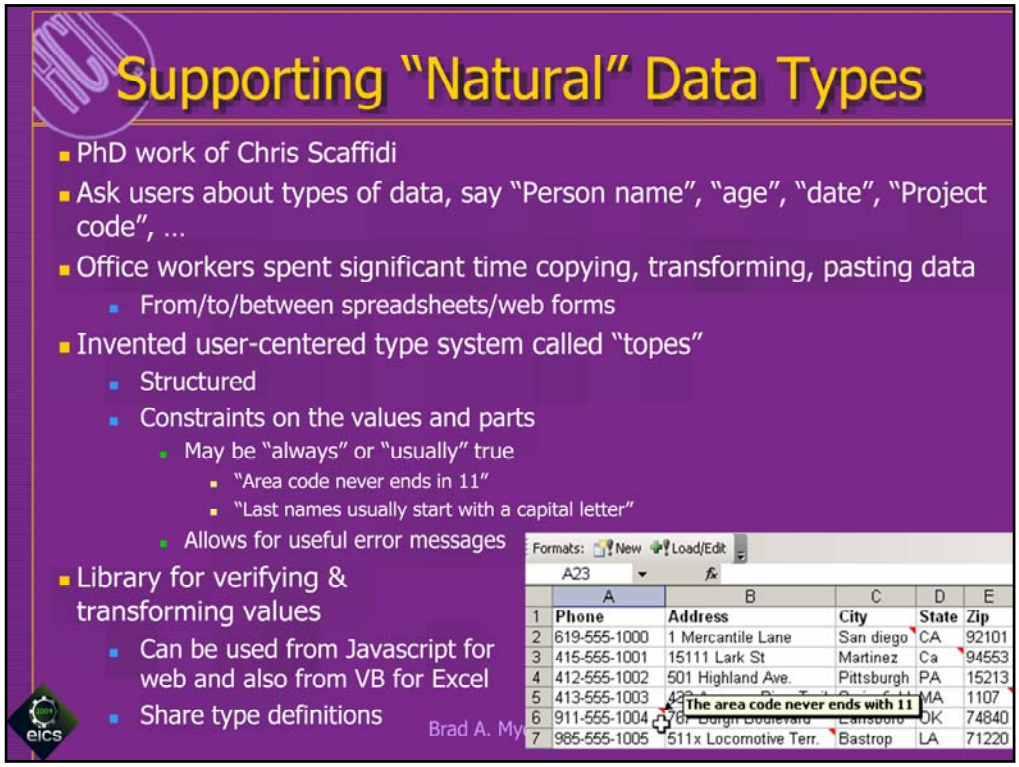
Beaton, J., Myers, B.A., Stylos, J., Jeong, S.Y., and Xie, Y. "Usability Evaluation for Enterprise SOA APIs," in *2nd International Workshop on Systems Development in SOA Environments, SDSOA 2008 (Co-located with ICSE 2008). 2008. Leipzig, Germany: pp. 29-34.*



Jeong, S.Y., Xie, Y., Beaton, J., Myers, B.A., Stylos, J., Ehret, R., Karstens, J., Efeoglu, A., and Busse, D.K. “Improving Documentation for eSOA APIs Through User Studies,” in *Second International Symposium on End User Development (IS-EUD'2009). 2009. Siegen, Germany: Springer-Verlag. LNCS 5435. pp. 86-105.*



LaToza, T.D., Garlan, D., Herbsleb, J.D., and Myers, B.A., "Program comprehension as fact finding," in *ESEC/FSE 2007: ACM SIGSOFT Symposium on the Foundations of Software EngineeringSeptember 3-7, 2007. Dubrovnik, Croatia. pp. 361-370.*



Scaffidi, C., Myers, B., and Shaw, M. "Topes: Reusable Abstractions for Validating Data," in *International Conference on Software Engineering. 2008. Leipzig, Germany: pp. 1-10.*
http://www.cs.cmu.edu/~cscaffid/papers/eu_20070914_topemod.pdf.

Topes, cont.

- Toped: Interactive editor for specifying Topes
- Can learn from examples or directly specified
- User test
  - Successful for novices
  - Significantly better at finding errors compared to regular expressions or previous research tools
  - Compared to manual checked, user generated topes were quite accurate
  - Faster to author a tope and check when >50 items to check
- Topes were able to easily express most formats found across a corpus of web forms and spreadsheets