The DIKU
Human-Centered Computing Group

PROGRAM
Introduktion af HCC-gruppen - Erik Frøkjær, 10 min.
IT og sundhed: CITH projektet - Jørgen Bansler, 10 min
Visualisering - Mikkel Rønne Jakobsen, 10 min
Experience Lab - Viktor Hansen og Jon Loldrup, 15 min
Focus of the HCC group

• Human-centered computing (HCC) is an “emerging, interdisciplinary academic field broadly concerned with computing and computational artifacts as they relate to the human condition”
• Background in human-computer interaction, user-centered design, and software development
• Core Computer Science topics (ACM curriculum: HCI, SE, Society & Prof. Practices) and new interdisciplinary educations
Current research

- Usability research (e.g., measuring usability, improving software development)
- IT in health care (e.g., electronic medical records, IT-based guidelines)
- Advanced interaction techniques (e.g., visualization, tangible user interfaces)
- Information search and natural language processing
Usability / User Experience research

• Concerns the concept of usability (or quality-in-use) and methods for improving it

• Research questions include:
  – What is usability or quality-in-use? (e.g., CHI 2000)
  – How to measure usability? (e.g., IJHCS 2006)
  – New methods, techniques and tools (e.g., MOT in TOCHI 2008, CUT at CHI 2005)
  – How to work systematically with usability in industrial software engineering? (e.g., CHI 2008)
Though usability evaluation is complex, simple things may matter a lot ... e.g. including business goals

Does people’s cultural background affect the importance they place on different aspects of usability?

Results: absolute questions

- Significant, medium-sized effect

Chinese (N = 258)
Danish (N = 154)
New Ways of Input
Robots on surfaces
Information access and use

• Increasing amounts of data; how do we support interaction with that data?

• Research questions include:
  – How to support understanding information?
  – When/how may information visualization help?
  – Improving information search using Venn diagram interfaces
Executable Object Modeling with Statecharts
David Harel & Eran Gery, Computer, JULY 1997, 30 no. 7, 31-42

Statecharts, popular for modeling system behavior in the structural analysis paradigm, are part of a fully executable language set for modeling object-oriented systems. The languages form the core of the emerging Unified Modeling Language.

Models for the development of object-oriented systems should be behaviorally expressive and rigorous as well as intuitive and well structured. Thus, any modeling approach must be detailed and precise enough to produce fully executable models and permit the automatic synthesis of efficient code in languages such as C++. Most OO modeling methodologies specify a model through graphical notation. Entity-relationship-like diagrams typically specify object classes and their interactions; however, it is necessary to describe how objects are and how they interact. Most methodologies also adopt a state-based formalism to specify behavior using statecharts for some sublanguages only.

However, many methodologies fail to rigorously define the semantics of the languages. Without a rigorous semantic definition, precise model behavior, over time is not well defined and full executability and automatic code synthesis is impossible. A richly expressive behavioral language like statecharts makes modeling easier but requires great care in defining the way it integrates with the other parts of the model. Statecharts must capture not only the state of the object as a precondition to service requests, but also the dynamics of the object’s internal behavior in responding to those requests and in maintaining relationships with other objects.

These issues are complicated and go beyond recommending a modeling approach or methodology—they are language design concerns, requiring rigorous mathematical underpinnings. Both syntax and semantics must be fully worked out, any possible combination of constructs must be clearly characterized as syntactically legal or illegal, and each legal combination must be given a unique and formal meaning.

To address these needs, we embarked on an effort to develop an integrated set of diagrammatical languages for object modeling, built around statecharts, and to construct a supporting tool that produces a fully executable model and allows automatic code synthesis. The language set includes two constructive modeling languages (languages containing the information needed to execute the model or translate it into executable code):

- Object-model diagrams specify system structure by identifying object classes and their multiplicities, object relationships and roles, and subclassing relationships.
- Statecharts describe system behavior. A statechart attached to a class specifies all behavioral aspects of the objects in that class.
Strengths of the HCC group

• Strong points of the HCC group:
  – Empirical, use of multiple methods
  – Interdisciplinary (both research and education)
  – Building/understanding balance

• High-class research
  – Top-outlets (e.g., ACM CHI conference, ACM TOCHI)
  – Acclaim (e.g., invited to associate editorships at IJHCS/IwC/JUS and to key program committees)
Summary

• HCC is about computing as it relate to humans in particular with respect to constructing usable and useful IT

• Key research areas: usability, health care IT, advanced interaction styles

• Strengths: empirical/interdisciplinary, strong publications

• Challenges: small group, needs to increase funding, enjoy collaboration with highly ambitious and demanding professionals