Going To See The Wizard...

A Look At IBM’S Personal Wizard Project

Robert Palmer
ID: 105279781

CSE 352
IBM Personal Wizards Project

• Started in 2003 by Vittorio Castelli
• The Personal Wizards **project** has created and developed the field of collaborative programming-by-demonstration
• Collaborative programming-by-demonstration (PBD) is a way of developing rich procedure models by recording how one or more experts interact with an application while performing a specific task
• Recordings are then combined into an executable, distributable model via an appropriate learning algorithm.
IBM Personal Wizards Project - Problem statement:

• Computer systems provide limited support for performing common procedures such as configuring software components or setting up an e-meeting.

• Traditional knowledge capture techniques – scripts, wizards, documentation – are difficult to author and difficult to maintain.

• Traditional tutorial/evaluation tools fail to provide “hands on” immediacy.
IBM Personal Wizards Project - Problem Solution

• Capture procedural knowledge by observing experts performing a task:
  – PW is as easy to use as a macro recorder
  – Multiple experts contribute to a procedure that can be as powerful as a traditional script
  – A learning algorithm incrementally combines the recordings to produce a script that models the task
  – The learning algorithm allows an author to manually edit the script, to simplify learning and maintenance of the model

• The script can be used as
  - an automation tool,
  - smart documentation,
  - to create guided walkthroughs that provide benefits of both traditional automation and documentation
IBM Personal Wizards Project: Applications

Learning Procedural Knowledge by Demonstration

- Intelligent Documentation Authoring
- Know-how Dissemination
- Procedural Groupware Development
- Automated Testing
- Tutorial Creation
- Walkthrough Generation
- End-user Programming
- End-user Programming
Programming By Demonstration

• What do we mean by programming by demonstration?
• Appeared in software development research as early as the 1980s
• Teaching a computer or a robot new behaviors by demonstrating the task to transfer directly instead of programming it through machine commands.
• User performs a sequence of actions that the computer must repeat, generalizing it to be used in different data sets.
IBM Personal Wizards Project: Programming By Demonstration

• Collaborative PBD therefore extends traditional PBD by learning from multiple demonstration

• Combines the collective knowledge of multiple experts.

• This approach models how to perform the same general task under different environmental conditions
Key Technological Components

Authoring

• Cooperation between machine learning algorithms and human author
  – Produce transparent procedure model
  – Need fewer examples than an entirely automated approach
  – Allows the user to direct the learning algorithm, and impose semantic structure

• Incremental procedure model construction
  – Capture “well-worn paths” first
  – Augment procedure model when new paths are observed
  – Combine the collective knowledge of multiple experts

Playback

• In-context guidance
  – Visual cues displayed on the actual application GUI

• Automation
  – Automation at multiple granularities: from single-step to full procedure

• Mixed Initiative and Off-Track Tracking
  – User can take initiative at any point, and deviate from script
  – System tracks user and tries to realign actions to script
  – User can turn on recording to capture new procedure paths
Key Technological Component: Combining manual procedure editing with learning

Goal
• to allow interleaved automated learning and manual procedure editing
  – provides robustness against inaccurate user actions
  – overcomes limitations of automated induction.

Constraint
• Manual editing must have priority over automated learning.

Solution
• Build a learning algorithm that produce only hypotheses that are consistent with the results of editing.
  – Prevents the learning algorithm from undoing manual edits when provided with new observations
Key Technological Component::
Mixed initiative interaction

- Allow either system or user to perform actions
- User is not constrained to follow the procedure
- System tries to match user actions with the procedure by simultaneously considering multiple hypotheses
Key Technological Component: Documentation Generation and Playback

- Procedure model is represented in a human-readable format
- Information on how to identify widgets is captured automatically
- Step documentation
  - Screenshots can be automatically obtained for each user action
  - Screenshots are automatically linked to corresponding steps
  - Additional text can be added to the screenshots
- In-context guidance
  - Automatic widget highlighting
  - Text boxes can be associated with highlighted widget
  - GUI widget annotation can be used to attract user attention to specific parts of the GUI
Personal Wizards Project

- **Instrumentation**
  - See what user sees
  - See what user does

- **Abstraction**
  - Decide what User means

- **Execution**
  - Do what user wants to do

- **Learning**
  - Learn what user wants to do
Personal Wizards Architecture

- **Expert**
  - Recording & abstraction
  - Mouse clicks
  - Keystrokes
  - UI events

- **Recording & abstraction**
  - Traces of high-level user actions

- **Learning algorithm**
  - Procedure model (script)

- **Playback**
  - End user

- **Authoring**
  - Author
  - List of actions
  - Grouping, annotations, explicit programming

- **Incrementally Refine**
Personal Wizards Architecture

Watch
- Expert
- Expert
- Recording: Actions performed on application GUI

Learn
- Learning Algorithm (automatic procedure model Induction)
- Procedure model
- Grouping, annotations, editing
- Authoring
- Author

Share
- Playback
- End user

Refine
IBM Personal Wizards Project: Learning Algorithms

- **SimIOHMM**, the first algorithm to combine multiple recordings to produce a probabilistic model of the procedure;
- **Augmentation-Based learning (ABL)**, the first incremental learning algorithm that allows seamless interweaving of demonstrations and manual edits of the model;
- **Distributed ABL**, that extends ABL by supporting a new model of learning by reconciling separately modified procedure models through a version-control-like environment.
IBM Personal Wizard Project: SimIOHMM

- Similar Input-Output Hidden Markov Model
- Extends the Input-out Hidden Markov Model by biasing the learner with a similarity function defined on the states
  - States – representations of the GUI content prior to each action
- Automatically identifies equivalent steps in different traces or recordings of a procedure
  - Used to explain the differences between traces in terms of probabilistic decision points
IBM Personal Wizard Project: SimIOHMM

• Two problems with SimIOHMM
  – Produces a vague model that is very difficult to present to the end user in a readable form
  – It’s a batch algorithm – the user does not see the real-time effects of her actions on the model

• Thus, a second algorithm is used

• Augmentation – Based Learning (ABL)
IBM Personal Wizard Project: Augmentation-Based Learning

• Augmentation Based Learning builds procedure models in real-time from observations of the interactions between a user and an application
  – Each observation is summarized by a state-action pair (SAP)
  – A SAP consists of a representation of the content of the UI prior to the action and a description of the user action at an appropriate level of abstraction

• For example, a typical action description has the form “Uncheck ‘Use default compiler options’”

• ABL produces a program specified by a grammar selected to address the needs of the intended consumers
IBM Personal Wizards Project:
Incremental learning
- Start from existing procedure model, observe user actions, augment model

Procedure Model
Observed Actions

Align Actions To Model

Augment Model To Account For Unexplained Actions

Unexplained Action
IBM Personal Wizard Project: Distributed Augmentation-Based Learning

• Distributed Augmentation-Based Learning
  – An extension to ABL
  – Builds procedure models from the user demonstrations, from manual edits, and by combining procedure models separately modified by different authors

• Uses version control (something like CVS)

• DABL’s version control algorithm is responsible for merging independently updated versions of the model
IBM Personal Wizard Project: Distributed Augmentation-Based Learning

• The combination of independently modified versions of a procedure model must be a learning operation
  – The merge process must be able to infer new information that can be obtained only by considering together both versions being merged
• The merged procedure model must be compatible with the learning algorithm
• The learning algorithm must be able to modify a merged model using new traces
IBM Personal Wizards Project: Learning From Multiple Demonstrations

• Align similar steps in procedure:

  expert #1: [Diagram showing steps]

  expert #2: [Diagram showing steps]

Build procedure model:

- Repository available
- Project not checked out
- Project checked out

Repository Not available
Bibliography


• “Distributed Augmentation-Based Learning: A Learning Algorithm for Distributed Collaborative Programming-by-Demonstration”; Castelli, Vittorio, 2007; http://delivery.acm.org/10.1145/1220000/1216327/p160-castelli.pdf?key1=121632...