The Smart Personal Assistant: An Overview

Wayne Wobcke
Anh Nguyen, Van Ho, Alfred Krzywicki, Anna Wong

School of Computer Science and Engineering
University of New South Wales
Outline

- History: BT Intelligent Assistant
- Smart Internet Technology CRC
- E-Mail Management Assistant (EMMA)
  - Ripple Down Rules for "user controlled" personalization
- Smart Personal Assistant (SPA)
  - Agent-based dialogue management
  - Adaptive dialogue agents
  - Usability evaluation
- Calendar Assistant
  - Knowledge Acquisition/Data Mining for user modelling
- Conclusion
History: BT Intelligent Assistant

- Integrated system of personal assistants
  - Time management: Diary, Coordinator
  - Information management: Web, Yellow Pages
  - Communication management: E-Mail, Telephone
- Each assistant has own
  - User interface (all accessible via toolbar)
  - User model (some share common profile)
  - Learning mechanism (some use common mechanism)
- Communication between assistants using Zeus
- Coordination of assistants through plans
- Inspired by human-centred design
Innovations in IA

• Integration of paradigms
  • Classical AI + Fuzzy Logic (Diary, Coordinator, Web)
  • Bayesian Networks + Fuzzy Logic (Telephone, E-mail)
  • Agents + scheduling (Coordinator)

• Integration of technologies
  • Speech recognition (Telephone, E-mail)
  • Natural Language Processing (Yellow Pages)
  • Information Retrieval (Web, Yellow Pages)
  • Scheduling (Diary, Coordinator)
Basic Problem: Usability

• E-Mail
  • How long does the system take to learn?
  • What guarantees are there concerning accuracy?

• Diary
  • Is it truthful or does it represent the user?
  • How does the user specify preferences?

• Coordinator
  • Who will define the coordinator’s plans?
  • Will the user adopt a standard ontology?
Smart Internet Technology CRC

- Government supported university–industry collaboration
- 11 university, 1 government, 8 industry, 7 SME partners
- Adaptive Interfaces/Personal Assistants programme
  - Multi-modal user interfaces, Conversational agents, Personalization, Knowledge Acquisition, Machine Learning
  - 7 Research Assistants, 7 PhD students over 5 years
- Smart Personal Assistant project
  - Dialogue management for mobile device applications
  - 1.5 Research Assistants, 1 PhD student over 5 years
SPA Research Themes

- **Adaptivity**
  - Personalized services and interaction
  - Accommodate user’s changing preferences
  - Balance user control and system autonomy

- **Mobility**
  - Platforms such as wireless PDAs and mobile phones
  - Use of information about context
  - Architectures that support modular development

- **Usability**
  - Natural interfaces supporting multi-modal interaction
  - User-oriented design methodology
SPA Research Objectives

• Architectures
  • Platform to support device-independent interaction
  • Agent architectures for coordination of services
  • Thanks to Agent Oriented Software for JACK

• Dialogue Management
  • Agent-based dialogue model
  • Adaptive dialogue agents

• Personalization
  • Knowledge Acquisition techniques
  • Machine Learning/Data Mining algorithms
Outline

- History: BT Intelligent Assistant
- Smart Internet Technology CRC
- E-Mail Management Assistant (EMMA)
  - Ripple Down Rules for “user controlled” personalization
- Smart Personal Assistant (SPA)
  - Agent-based dialogue management
  - Adaptive dialogue agents
  - Usability evaluation
- Calendar Assistant
  - Knowledge Acquisition/Data Mining for user modelling
- Conclusion
EMMA

- Objective
  - E-mail management assistant with high accuracy

- Novel technique
  - Combines Ripple Down Rules and Machine Learning

- Result
  - Shows applicability of Ripple Down Rules to domain
EMMA Approach

- Address whole e-mail management process
  - Sorting, prioritizing, replying, archiving, deleting
- Use Ripple Down Rules (RDR)
  - Easy to maintain rule sets
  - More accurate than Machine Learning methods
- Combine RDR with Machine Learning
  - Make suggestions to user to help define rules
Ripple Down Rules

- Hierarchical system of if-then rules
- Allows multiple conclusions
- Allows incremental knowledge acquisition
- Support for maintaining consistency of rule base
- All conclusions validated by prior rules
- Easy to create and maintain 20000+ rules
Ripple Down Rules: Classification

Rule 0
Root

Rule 1
(a, b) -> CSE

Rule 2
h -> CS9999

Rule 3
(a, c) -> Friends

Rule 4
(b, e) -> CS8888

Rule 5
x -> Friends

Rule 6
d -> John

Rule 7
(e, f) -> CRC

Rule 8
a -> Assignment

Rule 9
(g, h) -> SPAT
Ripple Down Rules: Refinement

Rule 0
Root

Rule 1
(a, b) -> CSE

Rule 2
h -> CS9999

Rule 3
(a, c) -> Friends

Rule 4
(b, e) -> CS8888

Rule 5
x -> Friends

Rule 6
d -> John

Rule 7
(e, f) -> CRC

Rule 8
a -> Assignment

Rule 9
(g, h) -> SPAT

Rule 10
(y, z) -> ARC

Rule 11
(y, z) -> ARC
Ripple Down Rules in EMMA

- Rule conditions can refer to... 
  - Sender of message 
  - Recipient(s) of message 
  - Key phrases in message subject, body

- Rule conclusions can define... 
  - Virtual display folder for sorting 
  - Message priority (high, normal, low) 
  - Action (Read/Reply with template + Delete/Archive)
Subject: CRC Workshop

Date: Mon, 16 Sep 2002 10:33:00 +1000

From: Eric McCreadh <eric.mccreadh@cs.anu.edu.au>

To: Wayne Webster <wobxko@cs.unsw.edu.au>

CC: John Lloyd <jwl@discus.anu.edu.au>, Judy Kay <judy@usyd.edu.au>

Dear Wayne,

Attached is a slide of some recent results we have had relating to some of our research within the first SFAT project. Although they are only very preliminary they are very encouraging in showing the utility of combining KA and ML. It may be a bit late to include this slide in your presentation, however, I thought it would be useful in giving you a bit of a background.

Note that, for only a small cost in terms of precision you get much better recall by adding ML to a KA approach.

- Precision of KA: 98.7%
- Precision of ML: 86.6%
- Precision of KA + ML: 96.2%
Subject: CRC Workshop
Date: Mon, 16 Sep 2002 10:39:03 +1000
From: Eric McCreath <Eric.McCreath@cs.anu.edu.au>
To: Wayne Webcke <webcke@coe.unsw.edu.au>
CC: John LLOYD <jwl@discus.anu.edu.au>, Judy Kay <judy@cs.usyd.edu.au>

Dear Wayne,

Attached is a slide of some recent results we have had relating to some of our research within the first SPAT project. Although they are only very preliminary, they are very encouraging in showing the utility of combining KA and ML. It may be a bit late to include this slide in your presentation, however, I thought it would be useful in giving you a bit of a background.

Note that this research is only in the form of a prototype.

Best,

Eric
Subject: CRC Workshop
Date: Mon, 16 Sep 2002 10:39:03 -1000
From: Eric McCreath <Eric.McCreath@cs.anu.edu.au>
To: Wayne Webster <wobcke@csse.unsw.edu.au>
CC: John Lloyd <jml@discus.anu.edu.au>, Judy Kay <judy@cs.usyd.edu.au>

Dear Wayne,

Attached is a slide of some recent results we have had relating to some of our research within the first SPAT project. Although they are only very preliminary they are very encouraging in showing the utility of combining KA and ML. It may be a bit late to include this slide in your presentation, however, I thought it would be useful in giving you a bit of a background.

Note that, for only a small cost in terms of precision you get much better recall by adding ML to a KA approach.

Precision of KA: 96.7%
Precision of ML: 86.6%
Precision of KA + ML: 96.2%

Subject contains 'CRC'
Hi,

I've finished the first two questions of AI assignment 2, but the 3rd one is so difficult, especially the findall predicate. Could you please give me some hints? Thanks in advance!

The sentences below look good, but it doesn't work. There is something wrong with the first sentence.

\[
\text{findall}(X, \text{country}(X))
\]

Also about the findall statement.

I don't really get hints for yes/no query
\[
\text{Form} = \text{A}(\text{Query} \rightarrow \text{A}(\text{yes} = \text{no})).
\]

I've been trying to do the assignment but I'm still having trouble with the logical expression.

I've got the grammar for G2 as:

\[
\text{sentence}(	ext{Form}) \rightarrow \\
\text{verb}(\text{be}), \\
\text{noun}_{\text{phrase}}(\text{Number}, \text{XNPForm}), \\
\text{noun}_{\text{phrase}}(\text{out}, \text{in}, \text{XNPForm}2), \\
\text{Form} = \text{A}(\text{Query} \rightarrow \text{A}(\text{yes} = \text{no})).
\]
RDR and Machine Learning

• Help user select key words to classify single messages
  • Suggest key word if $P(\text{folder}|\text{word}) > P(\text{folder})$
• Suggest classification based on message content
  • Suggest folder that maximizes $P(\text{folder}|\text{words})$
• Help user maintain topic profiles for (some) folders
  • List of words ranked according to $P(\text{folder}|\text{word})$
• Using Naïve Bayes classification
# User Evaluation: Accuracy

<table>
<thead>
<tr>
<th>User</th>
<th># Rules</th>
<th># messages</th>
<th># classified</th>
<th># misclassified</th>
<th>% accuracy</th>
<th>% coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>36</td>
<td>156</td>
<td>134</td>
<td>4</td>
<td>97.01</td>
<td>85.90</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>259</td>
<td>225</td>
<td>9</td>
<td>96.00</td>
<td>86.87</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>236</td>
<td>165</td>
<td>7</td>
<td>95.76</td>
<td>69.92</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>620</td>
<td>460</td>
<td>3</td>
<td>99.35</td>
<td>74.19</td>
</tr>
<tr>
<td>5</td>
<td>24</td>
<td>367</td>
<td>312</td>
<td>15</td>
<td>95.19</td>
<td>85.01</td>
</tr>
<tr>
<td>6</td>
<td>131</td>
<td>1766</td>
<td>1696</td>
<td>82</td>
<td>95.17</td>
<td>96.04</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>25</td>
<td>22</td>
<td>1</td>
<td>95.45</td>
<td>88.00</td>
</tr>
<tr>
<td>8</td>
<td>55</td>
<td>276</td>
<td>258</td>
<td>7</td>
<td>97.29</td>
<td>93.48</td>
</tr>
</tbody>
</table>

Table 1. EMMA User Trial Results
User Evaluation: Usability

- Display of sorting folders in Inbox
  - All users *strongly agreed* that the display is useful
- Rule building
  - All users commented that the interface for defining rules is *very easy* or *easy* to use
- Limitations
  - Conditions cannot be removed from rules
  - More expressive rule language (boolean operations)
Outline

- History: BT Intelligent Assistant
- Smart Internet Technology CRC
- E-Mail Management Assistant (EMMA)
  - Ripple Down Rules for “user controlled” personalization
- Smart Personal Assistant (SPA)
  - Agent-based dialogue management
  - Adaptive dialogue agents
  - Usability evaluation
- Calendar Assistant
  - Knowledge Acquisition/Data Mining for user modelling
- Conclusion
SPA

• Objective
  • Unified speech/graphical interface to a coordinated set of personal assistants (e-mail and calendar)

• Novel technique
  • BDI architecture for agent-based dialogue management

• Result
  • Shows applicability of agent-based dialogue model
System Description

- Integrated collection of personal (task) assistants
- Each assistant specializes in a task domain
  - Currently e-mail and calendar management
- Users interact through a range of devices
  - Currently PDAs, desktops
- Focus on usability
  - Multi-modal natural language dialogue
  - Adapt to user’s device, context, preferences
System Requirements

• Coordination: Provide a single point of contact
  • Coherent dialogue with all task assistants
  • Easy to switch context between task assistants
  • Possible to use different devices

• Dialogue modelling: Flexible and adaptive interaction
  • Need to understand user’s intentions
  • Need to maintain conversational context
  • Need to control conversation flow
  • Need to exploit back-end information
Dialogue Manager Requirements

- Flexible
  - Handle mixed (user, system) initiative
- Extensible
  - Easy to maintain dialogue model (dialogue acts)
- Scalable
  - Easy to add new assistants (tasks, vocabularies)
- Adaptive
  - Adapt to user’s device, context, preferences
Dialogue Characteristics

- Dialogue model
  - User-independent for deployment with different users

- Initiative
  - Mainly user-driven (reactivity)
  - System initiative is essential (pro-activeness)
    - Clarification requests
    - Notifications of important events

- Dialogue manager functions
  - Maintain coherent interaction with user
  - Coordinate actions of personal assistants
System Architecture

- Graphical Interface
  - Text
  - Speech Recognition
  - Text-to-Speech Engine
- Coordinator
  - Partial Parser
- User Device, e.g., PDA
  - Calendar Agent
  - E-Mail Agent
  - Calendar Server
  - E-Mail Server
Current Platforms

• Speech engines
  • IBM ViaVoice on Linux RedHat 8.0 (dictation mode)
  • Dragon NaturallySpeaking on Windows XP (dictation mode)

• Front-end devices
  • PDAs: Sharp Zaurus SL-5600, HP iPaq hx4700
  • Internal/headset microphone

• Users
  • Native/non-native English speakers
  • Australian/South-East Asian voice profile
SPA Demonstration

http://www.cse.unsw.edu.au/~wobcke/spa.mov

(12 MB)
Speech Recognition Performance

- I need to see him at 5pm this Friday about workshop slides
  - I need to see him at 5pm this Friday about workshop slides
  - I need to see in at 5pm this Friday about workshop slides
  - I need to see him at 5pm this Friday about workshop’s lives
  - I need to see him at 5pm this Friday about workshop slight
  - I need to see him at 5pm this Friday about workshop’s lines

- Do I have any e-mail from my boss?
  - Do I have any mail from my bus?

- Do I have any e-mail from Beyong?
  - Do I have any mail from beyond?

- Do I have any e-mail from Anh?
  - Do I have any mail from an?
Partial Parsing

- Full parsing is inappropriate
  - Limited quality of existing speech software
  - Regular use of short-form expressions
  - Unconstrained language vocabulary
    - e.g. “Are there any new messages from . . . ”
- Shallow syntactic frame

<table>
<thead>
<tr>
<th>clause</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>connective</td>
<td>Expresses the relation of the clauses</td>
</tr>
<tr>
<td>type</td>
<td>Question, declaration, imperative, ...</td>
</tr>
<tr>
<td>subject</td>
<td>Syntactic subject</td>
</tr>
<tr>
<td>predicate</td>
<td>Main verb</td>
</tr>
<tr>
<td>direct object</td>
<td>Main object of the predicate</td>
</tr>
<tr>
<td>indirect object</td>
<td>Possible second object</td>
</tr>
<tr>
<td>complement phrase</td>
<td>Other information e.g. time, location</td>
</tr>
</tbody>
</table>
Agent-Based Dialogue Management

- Reactivity
  - Responses to user requests
- Pro-activeness
  - Clarification requests
  - Notifications to user
- BDI agent approach provides these features
- Key idea: Treat dialogue as goal-directed rational action
BDI Agent Architectures

- Beliefs, desires, intentions explicit
  - Pre-defined plans for achieving goals
- Interpreter cycle – PRS (Procedural Reasoning System)
  - Event-driven selection and execution of plans
Dialogue Management Beliefs

- Dialogue model
  - Discourse history (stack of conversational acts)
  - Salient list (ranked list of recently mentioned objects)
- Domain knowledge
  - Supported tasks (for each task assistant)
  - Domain-specific vocabularies for task interpretation
- User model
  - User context information (device, modalities, . . .)
Dialogue Management Plans

**INPUT**
- Graphical Actions
- Text
- Speech

**Partial Parser**
- Speech Recognizer

**OUTPUT**
- Graphical Actions
- Speech
- Text

**COORDINATOR**
- Semantic Analysis
- Pragmatic Analysis
- Response Generation

**E-Mail Task Processing**
- E-MAIL AGENT

**Calendar Task Processing**
- CALENDAR AGENT
Example: Folder Determination

PRECONDITION:
- Task domain is E-Mail Management
- Task type is Search, Archive, Delete, Notify

TRIGGER:
- Folder Interpretation event

CONTEXT:
- Task requires some folder as one of the task objects

BODY:
- Recognize folder-related phrases
- Resolve references
- Determine folder attributes

FAILURE:
- Generate RequestClarification event
Return Message Summary Plan

PRECONDITION:
- Task domain is E-mail Management
- Task result is available
- Result contains only one message

TRIGGER:
- ResponseGeneration event

CONTEXT:
- User is on PDA
- Message content is long ("long" can be learned)

BODY:
- Summarize message content
- Send summary to user interface on PDA

FAILURE:
- Send whole content to user interface on PDA
Reusable Discourse-Level Plans

- Semantic analysis
  - Domain Classification, Semantic Analysis
- Pragmatic analysis
  - Act Type Determination, Intention Identification, Act Handling plans, Reference Resolution, Task Type Determination, People Determination, Clarification Generation, Graphical Action Handling
- Response generation
  - Response Generation meta-plan
- Plans use declarative specification of domain knowledge
E-Mail Domain-Level Plans

- Pragmatic analysis
  - Message Determination, Folder Determination
- Task processing
  - E-Mail Task Processing
- Response generation
  - Task Response Handling, Task Response Generation plans
Extension to Calendar Domain

- Semantic analysis
  - Specify domain actions, domain-specific vocabulary
- Pragmatic analysis
  - Appointment Determination, To-Do Determination
- Task processing
  - Calendar Task Processing
- Response generation
  - Task Response Handling, Task Response Generation plans
Why Agent-Based Approach?

- Robustness
  - Agent can respond if task processing fails
- Abstraction
  - Discourse-level domain independent plans are reusable
- Modularity
  - Plan level of abstraction facilitates addition of new plans
- Scalability
  - Plan-level modularity facilitates integration of new assistants
- Adaptivity
  - Meta-reasoning strategies for learning plan selection
- Dialogue modelling and coordination are rational action
Dialogue Adaptation

• Why dialogue adaptation?
  • Content adaptation for mobile devices
  • Learning “dialogue strategies” (e.g. when to confirm)
  • Appropriate dialogue manager actions (when to interrupt)

• Input parameters for content adaptation
  • User device: desktop PC/PDA/phone
  • User physical context: quiet meeting/noisy airport
  • User preferences: likes short summaries of messages, etc.
Adaptive Plan Selection

- Meta-reasoning for learning plan selection
Adaptive Dialogue Agent

- SPA Coordinator
  - Implemented using JACK BDI interpreter
  - Meta-level reasoning supported using PlanChoice event
- Alkemy learner
  - Decision-tree learner
  - Typed, higher-order logic representation of learning cases
    - Supports representation of data with complex structure
  - Expressive predicate rewrite system
    - For constraining hypothesis space
- Integration of Alkemy into Coordinator
  - Learn plan selection strategies
Adaptive Response Generation

- Different plans for generating responses
  - Display message content or summary
  - Display subset of message headers
  - Display headers sorted by sender, priority or folder
- Return Response meta-plan
  - Intermediate step in the dialogue manager's plan selection
  - Query the learner to predict one or more possible options
  - Request user to choose the most appropriate option
  - Generate learning case to update the learner
  - Select the chosen plan for execution
Return Response Meta-Plan

- User Intention Identification
- Preference Processing
- Task Processing
- Request User Clarify Preferences
- Return Message Content
- Return Message Summary
- Return Message Sub-list
- Return Message List
- Return Messages Sorted by Sender
- Return Messages Sorted by Folder
- Alkemy Learner

(update)
Alkemy Problem Specification

• Learning individual
  \[ \text{Individual} = \text{Device} \times \text{Task} \times \text{Mode} \times (\text{Set Email}) \times \text{PlanName} \]

• Learning class
  \[ \text{Class: true/false} \]

• Function to be learned
  \[ \text{Individual} \rightarrow \text{Class} \]

• Data constructors
  \[ \text{Email} = \text{Sender} \times \text{Length} \times \text{Folder} \times \text{Priority} \]
  \[ \ldots \]
  \[ \text{Device: PDA, Desktop;} \]
  \[ \text{Mode: Speech, Text;} \]
  \[ \text{Task: Search, Read, Show, Notify;} \]
Alkemy Problem Specification

- Transformations
  - Transform data into appropriate forms
  - Transformed data to be used in learning process
- Extract the length of an e-mail
  
  \[
  \text{projLength}: \text{Email} \rightarrow \text{Int};
  \]
  
  \[
  \text{projLength}: \text{project(1)};
  \]
- If at least one e-mail in a set satisfies some condition
  
  \[
  \text{setMsgExists}: (\text{Email} \rightarrow \text{Bool}) \rightarrow (\text{Set Email}) \rightarrow \text{Bool};
  \]
  
  \[
  \text{setMsgExists}: \text{setexists(1)};
  \]
- Set contains only one message whose length is less than 30 lines
  
  \[
  \text{and} (\text{projMsgs} \circ \text{numOfMsgs(true)} \circ \text{eq1})
  \]
  
  \[
  (\text{projMsgs} \circ \text{setMsgExists}(\text{projLength} \circ \text{lt30}));
  \]
Alkemy Problem Specification

• Predicate rewrite system
  • Constrains the hypothesis space
  • Necessary because of limited availability of training data

• Example

```prolog
top --> projDevice \ top;
top --> projMode \ top;
top --> and (projMsgs \ numOfMsgs(true) \ eq0) (top);
top --> and (projMsgs \ numOfMsgs(true) \ eq1)
     (projMsgs \ setMsgExists(projPriority \ top));

top --> eqDevicePDA;
top --> eqModeSpeech;
top --> eqPriorityHIGH;
top --> lt30;
```
Sample Dialogue

User  Is there any new mail from Wayne?

SPA   You have one new message from Wayne Wobcke.

       The message is more than thirty lines, should I just show you the summary?

User  Yes please.

SPA   <Displays summary of the message from Wayne Wobcke>

SPA learns to display only the summary if the message length is more than thirty lines.
Sample Dialogue

**User**  Find all messages about meeting in the Inbox.

**SPA**  There are twenty messages about meeting in your Inbox.
I'm displaying the first ten messages.

<Displays the first ten message headers>

SPA has learned to show only the first ten message headers if there are fifteen or more messages in the result.
Sample Dialogue

User  Show me the one from John.

SPA    Here is the summary of the message from John Lloyd.

<Displays summary of the message from John Lloyd>

SPA has learned the user’s preferences: display the message summary if the message is not of high priority and its length is more than thirty lines.
Learned User Preferences

- If `numOfMsgs = 0` then:
  - `plan = ReturnResponse` (yes -> True)

- If `numOfMsgs = 1` then:
  - If `msgLength > 30` and `priority = HIGH` then:
    - `plan = ReturnContent` (yes -> True)
    - `plan = ReturnSummary` (no -> True)
  - If `numOfMsgs < 15` then:
    - `plan = SortedBySender` (yes -> False)
    - `plan = ReturnSubList` (no -> True)
Usability Evaluation

• Methodology
  • 10 users: 5 female/male, 5 IT/non-IT, aged 18–45
  • All native Australian English speakers
  • Training: Voice model + training tasks
  • Testing: Training tasks + test tasks
  • Usability lab setting (quiet!)

• Objective and subjective evaluation
  • Evaluate both dialogue management and usability
  • Adopt Stibler & Denny’s three-tiered framework
User Testing

• 12 tasks, mainly simple tasks
  • Task 6: Check that you have an appointment on Friday at 11am. Reschedule it to Monday next week at 2pm.
  • Task 10: You have received messages about the war with Israel. Please find and then delete all of them.
  • Task 12: Find your e-mails for today. Read the message from Kate and complete any requests that the sender has asked of you.

• Complete tasks with speech only (no stylus)
  • Concept-word recognition: 82–91%
  • Utterances with no concept-word errors: 56–82%
Concept-Word Recognition

![Graph showing the percentage of correctly recognized concept-words per user. The graph plots the accuracy for each user, with a general trend of high accuracy, except for user 8 which shows a significant drop. The average accuracy is also indicated.]
Average Dialogue Length

![Graph showing average dialogue length across different tasks. The x-axis represents task numbers, and the y-axis represents the number of utterances. The graph includes bars for average, max, and min number of utterances, as well as overall average.]
Task Completion

- Scoring scheme (1 correct, 0.5 with help on wording)
  - Average score 10.1 (out of 12)
- High overall task completion rate (88%)
  - Though not the full story
- 14 failures in 120 tasks (4 gave up, 10 incorrect)
  - Speech recognition (8 of 14), e.g. Kate, Lloyd, budget
  - Dialogue (5 of 14), e.g. unclear confirmations
  - User (3 of 14), e.g. failure to change meeting time
  - Hard to recover from compounded errors
Sample Error Dialogue

**User**  Find e-mails about Israel.

*Fine* e-mails about Israel. [e-mail/Israel]

**SPA**  You have 4 messages about Israel.

**User**  Delete all these e-mails.

*Delete all these e-mails. [ALL e-mail]*

**SPA**  Are you sure you want to delete those messages?

**User**  Yes.

*Yes.*

**SPA**  Messages have been deleted.

*Need better confirmations, e.g. delete those 15 messages*
Utterance-Level Evaluation

- Unexpected response (202 of 569, 122 due to speech)
  - Wrong response from user’s point of view
- Inappropriate response (87 of 569)
  - Wrong response assuming correct speech recognition
- Attribute error to first erroneous component/aspect
- Dialogue management errors
  - Object identification errors (require preposition)
  - Required use of references (delete it)
  - Speech errors causing lost information (Monday)
  - Task identification errors (rename, find, check)
  - Context switching (users don’t track changes)
Unexpected Responses

**Chart Description:**
- **Y-axis:** Number of Unexpected Responses
- **X-axis:** User Number
- Bars represent different causes:
  - Caused by Parser
  - Caused by Dialogue
  - Caused by User
  - Caused by Speech

The chart illustrates the distribution of unexpected responses across user numbers, categorized by the cause.
Inappropriate Responses

![Graph showing the number of inappropriate responses for different user numbers, categorized by cause (Back-end, Parser, Dialogue)].
User Satisfaction

- Feedback from the SPA is clear and easy to understand: 4.3
- The SPA understood what I asked it to do: 4.0/6
- It was easy to make requests the SPA could understand: 3.7
- The SPA gave reasonable responses to my requests: 4.1
- Using the SPA is frustrating: 2.9
- The SPA responded in a timely manner: 4.1
- I was happy about the overall performance of the SPA: 4.1
- I would use a system like the SPA in future: 3.8

- Dialogue Manager: 482/569 (85%) processed correctly
- “Frustrating, but fun!”
Improvements

• Proper names
  • As expected, poor recognition performance
  • Solutions: Phonetic dictionary, multi-modal input from GUI, match names to address book, dynamically update vocabulary

• Context tracking
  • Users do not notice changes made by SPA
  • Solution: More explicit flagging of context changes

• Interaction styles
  • Variety of styles: “polite” (regarding), “precise” (the Friday 2pm)
  • Solution: Handle a wider variety of expressions
Outline

- History: BT Intelligent Assistant
- Smart Internet Technology CRC
- E-Mail Management Assistant (EMMA)
  - Ripple Down Rules for “user controlled” personalization
- Smart Personal Assistant (SPA)
  - Agent-based dialogue management
  - Adaptive dialogue agents
  - Usability evaluation
- Calendar Assistant
  - Knowledge Acquisition/Data Mining for user modelling
- Conclusion
Calendar Assistant

- Objective
  - Personalized meeting scheduling
- Novel technique
  - Application of Ripple Down Rules and Data Mining for suggesting attributes of structured objects
- Result
  - Shows suitability of Cascaded Ripple Down Rules
System Description

• User model based on Cascaded Ripple Down Rules
  • Multiple passes through rule base, each generating attributes
  • No pre-determined order of attribute generation
  • Rules represent user’s personal preferences
• Implemented on PDA with Generalized RDR engine
• Suggest suitable attributes for user’s appointments
  • Location, attendees, day, time, duration
• Potential for Data Mining to improve suggestions
Calendar Scenario

Rules

crc ⇒ 401k & Wed & 10:30
& anna, anh, wayne, alfred
401k ⇒ 90min
Calendar Scenario

Rules

crc ⇒ 401k & Wed & 10:30
& anna, anh, wayne, alfred
401k ⇒ 90min

New crc project meeting
Rules

crc → 401k & Wed & 10:30
  & anna, anh, wayne, alfred

401k → 90min

New crc project meeting

Refinement of crc rule

crc & semester → 401k & Tue & 10:30
  & anna, anh, wayne, alfred
Rules

$crc \Rightarrow 401k \& \text{Wed} \& 10:30$

& $anna, anh, wayne, alfred$

$401k \Rightarrow 90\text{min}$

New $crc$ project meeting

Refinement of $crc$ rule

$crc \& \text{semester} \Rightarrow 401k \& \text{Tue} \& 10:30$

& $anna, anh, wayne, alfred$

Suggest attributes for $crc \& \text{semester}$
Calendar Scenario

Rules
- \( \text{crc} \Rightarrow 401k \& \text{Wed} \& 10:30 \)
  - & anna, anh, alfred, wayne
- \( 401k \Rightarrow 90\text{min} \)

New crc project meeting

Refinement of crc rule
- \( \text{crc} \& \text{semester} \Rightarrow 401k \& \text{Tue} \& 10:30 \)
  - & anna, anh, wayne, alfred

Suggest attributes for crc & semester

New (conflicting) rule
- \( \text{crc} \& \text{semester} \Rightarrow 401k \& \text{Tue} \& 10:30 \)
  - & anh, wayne, alfred
Rules

crc ⇒ 401k & Wed & 10:30
   & anna, anh, alfred, wayne
401k ⇒ 90min

New crc project meeting

Refinement of crc rule

crc & semester ⇒ 401k & Tue & 10:30
   & anna, anh, wayne, alfred

Suggest attributes for crc & semester

New (conflicting) rule

crc & semester ⇒ 401k & Tue & 10:30
   & anh, wayne, alfred

Suggest attributes for crc & semester
Calendar Scenario

Rules
- crc ⇒ 401k & Wed & 10:30
  & anna, anh, alfred, wayne
- 401k ⇒ 90min

New crc project meeting

Refinement of crc rule
- crc & semester ⇒ 401k & Tue & 10:30
  & anna, anh, wayne, alfred

Suggest attributes for crc & semester

New (conflicting) rule
- crc & semester ⇒ 401k & Tue & 10:30
  & anh, wayne, alfred

Suggest attributes for crc & semester

User selects desired attributes
Calendar Design Features

- **Generality**
  - Built using Generalized RDR engine
  - Shows applicability of Cascaded RDR to generating attributes of structured objects in arbitrary order

- **Usability**
  - Easy to create appointments using suggestions for attributes
  - Potential for Data Mining to be used for suggesting rules and ranking suggestions

- **Techniques applicable in other domains**
Conclusion

• Architectures
  • JACK supports device-independent interaction
  • BDI agent approach supports service coordination

• Dialogue Management
  • Networked speech engine using Dragon NaturallySpeaking
  • Agent-based model provides modularity, extensibility, reuse
  • Adaptivity through integrated Alkemy with BDI cycle
  • Positive user evaluation though issues with speech recognition

• Personalization
  • Shown value of RDR in e-mail classification
  • Work on RDR/DM in calendar domain in progress
Further Work

- Smart Internet CRC ⇒ Smart Services CRC
  - 5 university, 12 (different) industry partners
  - Development using service-oriented architectures
  - Applications in finance, media and government
  - Mobile speech (?) services in these domains?
- Dialogue
  - How to manage “long-term” interaction?
- Teamwork
  - How to provide support for workplace teams?
  - How to support team-oriented dialogue?