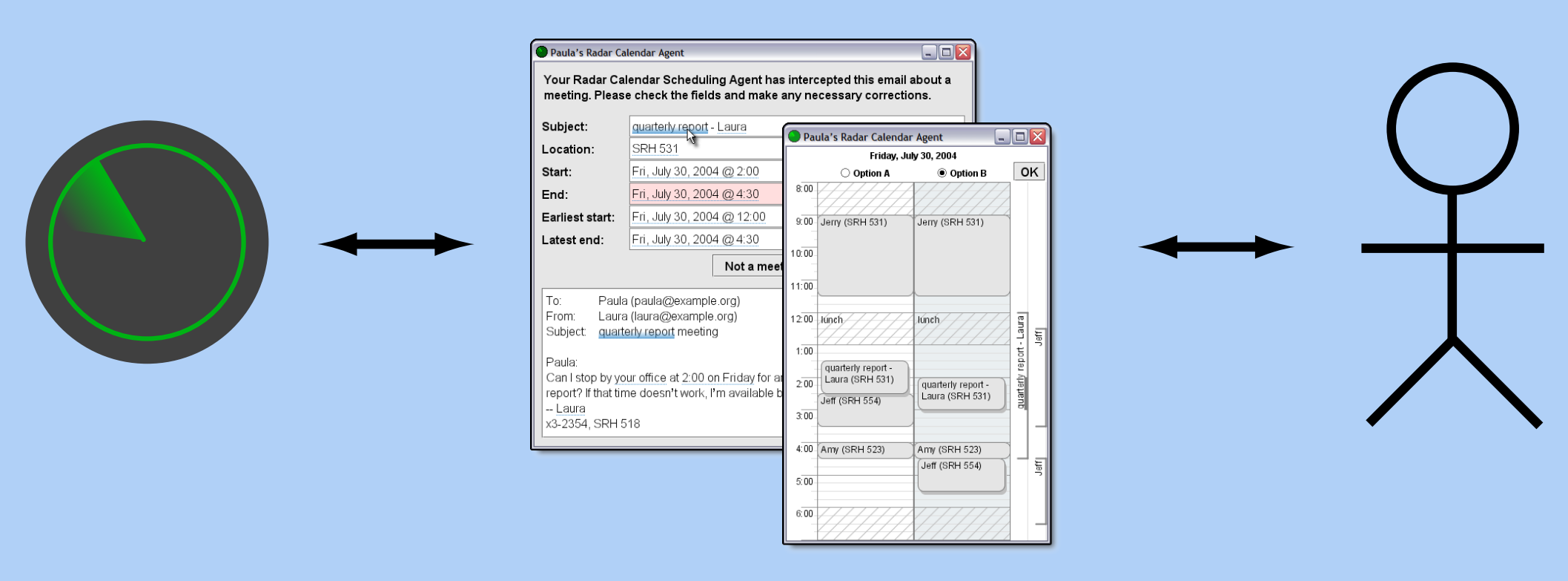
Enabling Rich Human-Agent Interaction for a Calendar Scheduling Agent

Andrew Faulring and Brad A. Myers
{faulring, bam}@cs.cm.edu
School of Computer Science, Carnegie Mellon University
http://www.cs.cmu.edu/~faulring/rhai.html

Rich
Human-
Agent
Interaction for
CALendaring

Problem

- Just like a human assistant, an agent needs to consult its supervisor when asked to perform a task that is under-specified, has ambiguous instructions, deviates from normal, or has changed.
- What is a good user interface for interaction between a user and an agent? How should an agent ask a user to check its understanding of natural language? How should an agent ask a user to approve, reject, or modify actions that it proposes?
- We are using calendar scheduling to answer these questions.

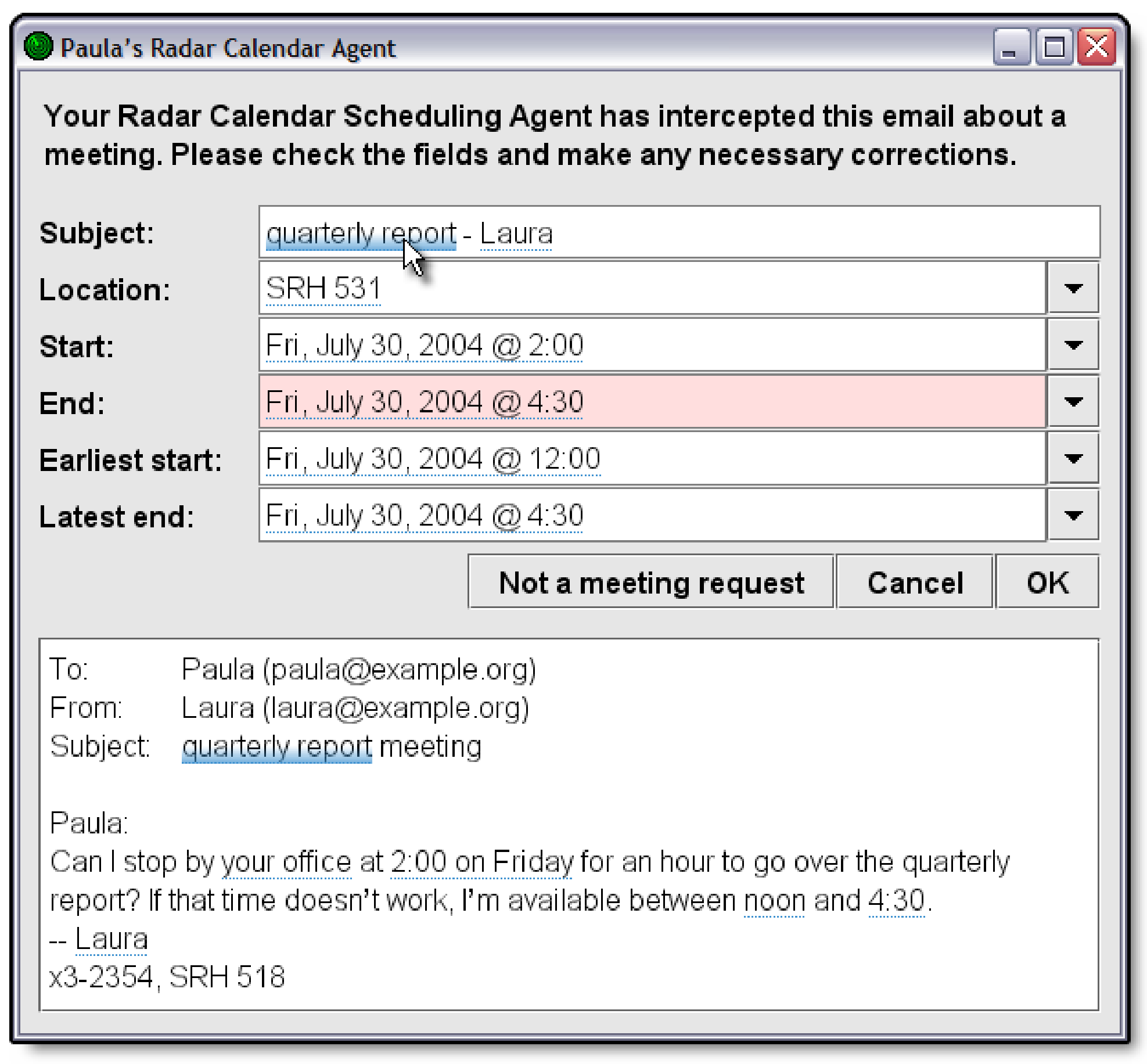


Solution

- The RhaiCAL system provides novel visualizations and interaction techniques for interacting with an intelligent agent, with an emphasis on calendar scheduling.
- After an agent interprets natural language containing meeting information, a user can easily correct mistakes using RhaiCAL's clarification dialogs, which provide the agent with feedback to improve its performance.
- When an agent proposes to take actions on the user's behalf, it can ask the user to review them. RhaiCAL uses novel visualizations to present proposals to the user and to allow the user to modify or reject them. The agent is informed of the user's actions in a manner that supports long-term learning of the user's preferences.
- We have designed a high-level XML-based language that allows an agent to express its questions and proposed actions without mentioning user interface details, and that enables RhaiCAL to generate high-quality user interfaces.

Clarifying Natural Language

- A calendar scheduling agent intercepts and interprets emails. A RhaiCAL clarification dialog box (shown on the right) lets a user to check and correct the agent's interpretation of natural language.
- The agent's instructions appear at the top, the properties of the request in the middle, and the text of the original email at the bottom.
- *Anchors* associate text in the source email with text in the request's properties. Anchors inform the user from where the agent found a property's value. Inactive anchors ("Laura") are drawn with a blue dotted underline; active anchors ("quarterly report") are drawn with a thick blue underline.
- When the user copy-and-pastes or drag-and-drops text from the email into a property, RhaiCAL creates an anchor to preserve the association. RhaiCAL returns these anchors to the agent as feedback to improve its performance through learning.



Paula's Radar Calendar Agent

Your Radar Calendar Scheduling Agent has intercepted this email about a meeting. Please check the fields and make any necessary corrections.

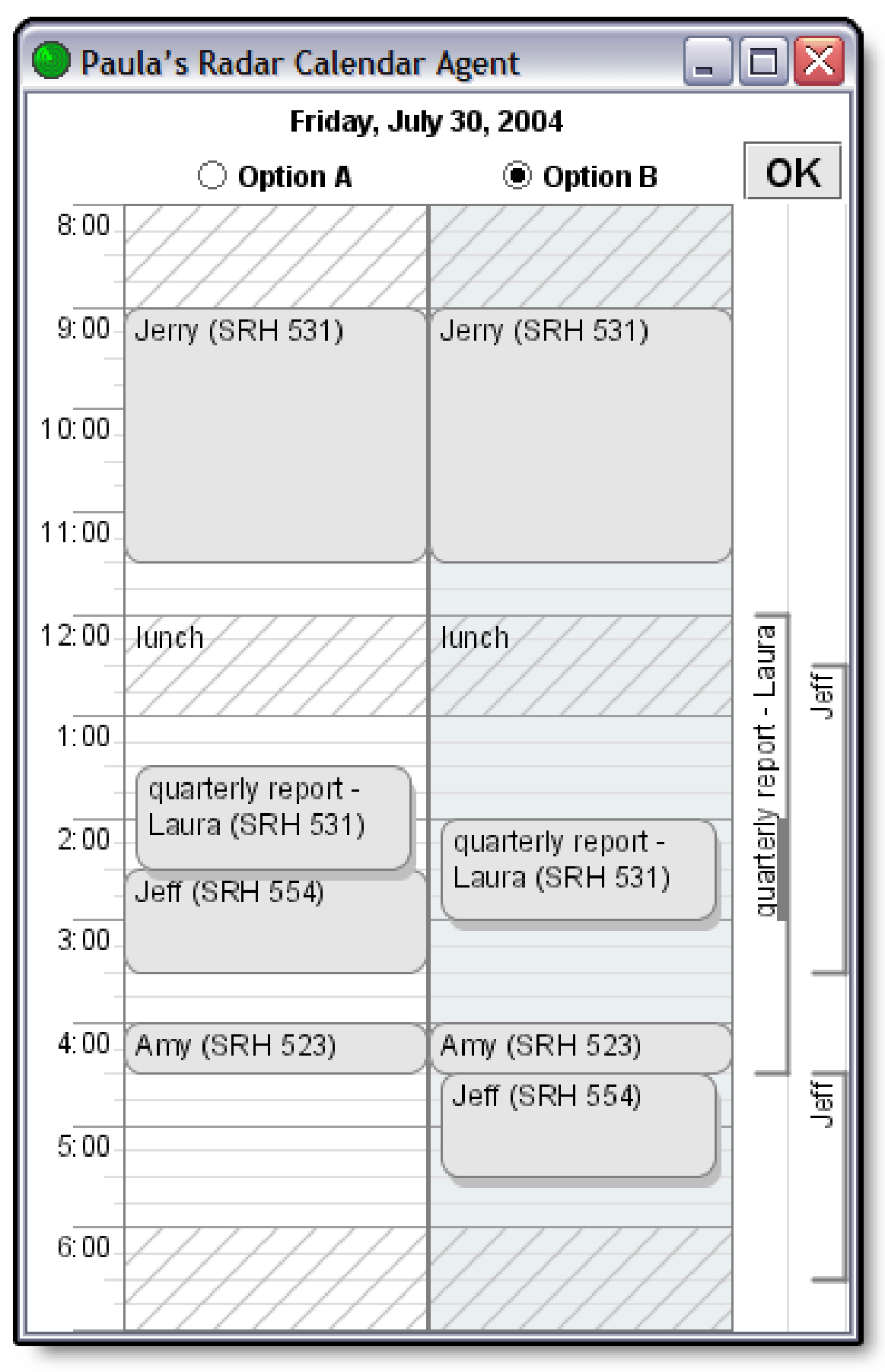
Subject:	quarterly report - Laura
Location:	SRH 531
Start:	Fri, July 30, 2004 @ 2:00
End:	Fri, July 30, 2004 @ 4:30
Earliest start:	Fri, July 30, 2004 @ 12:00
Latest end:	Fri, July 30, 2004 @ 4:30

To: Paula (paula@example.org)
From: Laura (laura@example.org)
Subject: quarterly report meeting

Paula:
Can I stop by your office at 2:00 on Friday for an hour to go over the quarterly report? If that time doesn't work, I'm available between noon and 4:30.
-- Laura
x3-2354, SRH 518

Proposing Schedules

- After the agent clarifies its understanding of Laura's email, the agent searches for schedules that satisfy Laura's request and Paula's preferences. RhaiCAL visualizes the agent's suggestions within the context of Paula's calendar to allow her to see how the proposal relates to calendar entries.
- RhaiCAL allows the agent to propose multiple schedules, which are shown side-by-side.
- RhaiCAL uses three visual layers to group the meetings:
  - The backmost layer contains items that represent the user's preferences such as lunch time. Items in this layer are filled with a light gray diagonal pattern.
  - The next layer contains existing meetings in Paula's calendar, such as those with Jerry and Amy.
  - The foremost layer contains meetings that the agent is proposing to add or change. These meetings are drawn with shadows.
- The vertical bars on the right, labeled with subject for the associated meeting, show the scheduling constraints and preferences of other meeting participants. The bar shows when the other participant is available, and the thicker bar shows the times that they prefer.
- The user edits a meeting by dragging it around the calendar or by double clicking on it to bring up a property sheet dialog. When done responding to the proposal, the user presses the "OK" button.



Communicating between Agent and UI

- RhaiCAL's XML-based language supports communication between an agent and the RhaiCAL user interface runtime. The example to the right shows an excerpt of the XML that an agent would send to RhaiCAL to propose adding the "quarter report - Laura" meeting.
- Primitive types: boolean, integer, fixed-point, floating-point, time, date, and string.
- Aggregate types: object (collection of properties), or list.
- An `<action>` block proposes a change to some data. In this example, the agent is proposing to `add` a `meeting` object to the user's calendar. Actions can be grouped together, such as "Option B" in the above example: add the meeting with Laura at 2:00 and bump Jeff's meeting to 4:30.
- Each `<option>` block contains an optional `origin` attribute, which links to the anchor from which the option's value was derived.
- A `<constraint>` block describes constraints on property values. In this example, the constraint specifies another person's available and preferred times for the meeting.
- Other constraints group options of different properties, and specify that an option should only be used once in a set of properties.

```
<action type="add">
  <parameter name="object" type="meeting">
    <object id="mtg-F408" type="meeting" option="a">
      <property name="start">
        <option>2004-07-30 13:30:00 -0400</option>
      </property>
      <property name="end">
        <option>2004-07-30 14:30:00 -0400</option>
      </property>
      <property name="summary">
        <option confidence="0.8">
          <span origin="#B7C3">quarterly report</span> -
          <span origin="#A18D">Laura</span>
        </option>
      </property>
      <property name="location">
        <option origin="#CC89">SRH 531</option>
      </property>
    </object>
    <constraint properties="start, end">
      <inclusive-range>
        <min origin="#1288">2004-07-30 12:00:00 -0400</min>
        <max origin="#D8AB">2004-07-30 16:30:00 -0400</max>
      </inclusive-range>
      <preference>
        <inclusive-range>
          <min origin="#CC89">2004-07-30 14:00:00 -0400</min>
          <max>2004-07-30 15:00:00 -0400</max>
        </inclusive-range>
      </preference>
    </constraint>
  </parameter>
</action>
```

Acknowledgements
The authors thank Ellen Ayoob, Andrew Ko, Jay Modi, Jeffrey Nichols, Jean Oh, Jeff Pierce, Bill Scherlis, Desney Tan, Aaron Spaulding, Anthony Tomasic, and John Zimmerman. Andrew Faulring is supported by a NSF Graduate Research Fellowship. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the DARPA or the Department of Interior-National Business Center (DOI-NBC).