Independent LifeStyle Assistant™ (I.L.S.A.):
AI Lessons Learned
A NIST ATP Program

Karen Zita Haigh
Karen.haigh@honeywell.com
In a Nutshell

Program Objective

Develop an intelligent home automation system with situation awareness and decision-making capability based on integration of diverse sensors, devices, and appliances to support caregivers and enable elderly users to live independently at home.

Expected Benefits:

- Support elder independent living
- Provide peace of mind to caregivers
- Support efficient quality care for caregiving organizations
- Cost savings for government and industry
- Market growth for in-home product producers
The Vision

Lois is doing fine. I’ll check on her again this afternoon.

Lois ate breakfast at 8:20.

It’s time to take your medicine!

Lois is in the living room.

Mom’s having a good day!

Lois is fine.

10:00 A.M.
Time for medicine

Honeywell Laboratories
IAAI 2004, July 25-29 2004, San Jose CA
Factors Precipitating Institutionalization

Literature reviews, interviews with adult children caregivers, and discussions with geriatric experts identified the most significant factors that pose a threat to the independence of elders.

- Mobility
- Medication Management
- Eating
- Toileting
- Isolation
- Medical Monitoring
- Cognitive Decline
- Safety
- Caregiver Burnout

Existing monitoring systems often focus on a single function – little or no integration.
Feature Set

Monitoring Functions
- Mobility (general activity level)
- Medication caddy monitoring

Response Functions
- Alerts
- Notifications
- Activity Reports

Service Features
- Reminders
- Internet & phone access

Usability Features
- Password-free elder interactions
- Operational modes (on/off)

User Interfaces
- Elder: Phone, Webpad™
- Caregiver: Web, phone

Design Philosophies:

Passive
- Allow elders to follow regular routines without imposing new ones
- No worn devices

Minimal intrusions
- Only reminders and alerts
- No requirement to use web interface for proper system behaviour
I.L.S.A. Field Study System

Wireless Sensors
monitor general or specific activities

Client Interface
Honeywell Webpad™ anywhere in client’s home

Caregiver Browser
From any internet connection

Hidden control and communication components

I.L.S.A. Server
Modular agent-based System

Honeywell Laboratories
I.L.S.A. Client Interface

- Reminders
  - Angie is coming to clean your house at 3:00.
  - You have a doctor's appointment on Monday at 9:00 AM.

- Mobility
- Medicine
- Controls
- Caregiver

- Medicine Today
  - 07:35 AM Saturday, June 01
  - Last Medication access at 8:00 AM

- Honeywell Laboratories

Karen Haigh

IAAI 2004, July 25-29 2004, San Jose CA
Software Architecture Requirements

Each ILSA client and home will be very different and have specialized needs, so the system must be:

- rapidly deployable,
- easily configurable,
- highly modular, and
- adaptive to the environment.

Modularity is critical both to functionality as well as expandability for a number of reasons:

- Integrate 3rd party functional units
- Flexibility of sensor and actuator suites
- Expansion of ILSA capabilities over time
Agent Architecture

Response Execution
Talks to devices (displays & actuators)

Response Planning
Based on situation, creates general response plan -- what to do or who to talk to, how to present it, on what device

Situation Assessment & Response Monitoring
Based on evidence, predict ramifications.

Clustering
Combine multiple sensor reports into a single event.

Intent Inference
Infer goals of actors; put multiple events together.

Agent Layer

Sensor Adapter

Device Layer

Actuators

Sensors

Honeywell Laboratories

Karen Haigh

IAAI 2004, July 25-29 2004, San Jose CA
I.L.S.A. Agents

Agents group functionality, e.g.
- Mobility monitor
- Medication monitor
- Client interaction module
- Device controllers

Agents group technical capability, e.g.
- Machine Learning
- Task tracking
- Response Planning
Lessons - Agents: Multi-person development

Expectation: independent agents could be assigned to independent developers
Result: not true – considerable development overhead

- Agents that communicate with each other must be developed together
  - Communication protocols
  - Recovery from failures
  - Ontology development
  - Logical protocols
Lessons - Agents:
Testing & Debugging

Expectation: independent agents could be independently tested

Result: not true

- Free communication means every possible interaction needs to be tested
  - In a monolithic system, testing can focus on the single point of change

- Errors can propagate along communication channels, and therefore are hard to isolate

- No enforcement of logical protocols – same bug may appear in multiple agents
Lessons - Agents: Scalability

Expectation: distributed architecture would support scalability

Result: not true

- Bottleneck agents
  - e.g. database, communication with elder

- Scoping is very difficult because there is no mechanism to enforce logical protocols

- New capability (agent) meant new interfaces for existing agents
  - Compounded by multi-person development
  - Adds to testing effort
Lessons - Agents: Robustness & Reliability

Expectation: distributed processing would mean no single point of failure

Result: not true

- Certain capabilities need to be centralized
  - e.g. communication with elder
  - Redundancy is not a solution

- There is no general solution to persistence over restarts
  - Each agent must have its own solution
Lessons - Agents: Summary

Agent technology is not ready for this domain. It needs much more support for

- Debugging & Testing
- Reliability
- Enforcing logical protocols

The more capabilities need to be centralized, the less likely agent technology will be appropriate.
A common vocabulary that lets agents communicate with precision about the world

It provides standard interpretations for words

- that might otherwise be dangerously ambiguous

It structures the domain knowledge in ways that allow it to be analyzed,

- making assumptions more explicit

Being presented to CAST / HL7
Ontology: Lessons

One ontology for multiple purposes means no duplicated concepts, but is harder to learn

Don’t waste effort defining terms not explicitly dictated by the application

Be conscious of cross-cultural compatibility
Artificial Intelligence

Task Tracking
Response Planning & Coordination
Machine Learning

Short answer: AI is very useful for this domain, especially as tasks grow in complexity
AI: Task Tracking

Recognize what the client is doing

System must handle:

- Multiple hypotheses
  - One sensor sequence may mean two different things
  - Be aware of how confident it is in the recognized sequence (e.g. competing possibilities, or noisy sensors),
- Unobservable actions
  - e.g. when a sensor failed
- Abandoned plans & Failed actions
  - Recognize what the person was TRYING to do, even if they didn't actually succeed or have not yet completed the task
- Partially ordered plans
- Actions used for multiple effects

Barriers:

- Richness of sensor suites
- Libraries of activities
AI: Response Planning

Generate interactions with the client/CG
System must coordinate responses

- *(who, what, where, when, how)*
- Timely
- Prioritize messages
- Multiplex messages
- Without overloading the resources (device or human)

**Challenges**

- Accurate use of context
  - » Never cry wolf
AI: Machine Learning

Learn models of the actors and environment to automatically improve the performance of the system:

- what is normal / unusual
  - Patterned behaviours, Schedules, Unexpected activities
- what is the most effective technique to use
- understand sensor reliability

For configuration, adapting to the changing elder, and capturing preferences of users

Barriers:

- Evaluation (no ground truth)
- Automatic incorporation of learned models
Reactions to I.L.S.A.

Clients were engaged and interested

- Most clients checked their page at least once a day, even in the last month of testing

Clients did not appear to become dependent on reminders

- In fact, avoiding telephone reminders helped them exercise their memory

Clients liked the minimal disruption to their normal routine.
Summary

AI is perfect for this domain

- But don’t forget the other problems you’ll encounter when you field a system

Agent architectures need more support tools before they will make it out of the lab

- Particularly since AI researchers are not software engineers

http://www.htc.honeywell.com/projects/ilsa

(Deployment lessons – see AAAI-04 workshop “Fielding AI Technologies”)