Location & Context Awareness

Alexander Nelson
March 4th, 2019

University of Arkansas - Department of Computer Science and Computer Engineering
Conversation between humans includes more than the verbal exchange of words.

We use multiple sources of information to increase conversational bandwidth.

i.e. Bits of information per word is greater than the words themselves.

- Tone
- Body Language
- Current context
- Previous conversations
Communication with computers is (traditionally) more verbose
Must translate \textbf{exact} meaning to the computer

Improve Computer’s contextual access to create more rich interaction
Two basic approaches to improve input:

1. Improve the language humans use to interact with computers
   - Ontologies
   - Natural Language Processing

2. Increase situational information available to computers
   - Context Awareness
Grounding
Grounding – Clark and Brennan 1991
Collection of "mutual knowledge, mutual beliefs, and mutual assumptions" essential for communication between 2 or more people

Grounding Criterion – Mutual belief between conversational partners that everyone has enough understanding to move forward
Establishing Grounding

To establish grounding, partners exchange information in a two phase sequence

- Presenting Utterance – Speaker presents an utterance to the Addressee
- Accepting Utterance – Addressee accepts utterance by providing evidence of understanding
Conversational Grounding

In a conversation, grounding presents three types of evidence:

- **Acknowledgments** – Utterances that prove understanding. e.g. Nodding, “yeah”
- **Relevant Next Turn** – Invitation to respond between speakers e.g. Questions & Answers
- **Continued Attention** – Mutual belief that addressees have identified a referent
  Typically identified through gaze – Looking at the speaker
Chai et al (2014) – Mediation of shared perceptions results in better human/robot interaction

H1: The green cup is called Bill.
R1: Ok, the green cup is Bill. (point to the inferred object)
H2: Do you see a car?
R2: Yes, I see a car.
H3: Do you see a blue can behind it?
R3: I don’t see a blue can. But I see a blue unknown object there. Is that the can you are talking about? (point to the inferred object)
H4: Yes, it’s called Mary.
R4: Got it, the blue can is Mary.

(a) Situated Setup

(b) Sample Dialogue
Anticipation of Communication

Intermediate communication may be skipped if speaker can anticipate addressee knowledge

Three factors of anticipation:

- Community Co-membership – Group with knowledge of particular field use technical jargon
- Linguistic Co-presence – Party in a conversation can use pronouns to refer to someone previously mentioned
- Physical Co-presence – Other parties who are physically present may be physically referenced (e.g. pointing)
Shared visual information aids anticipation
30-40% increase in speed when sharing visual workspace
“Using Visual Information for Grounding and Awareness in Collaborative Tasks” – 2013
Constraints to mediated communication

1. Copresence – Colocation; group members are in the same location
2. Visibility – Group members can see each other
3. Audibility – Group members can hear each other speaking
4. Contemporality – Group members are receiving information as it is produced
5. Simultaneity – Members receive/produce information at the same time
6. Sequentiality – Group remembers receive information in a consecutive sequence
7. Reviewability – Can review information received for others
8. Revisability – Can review their own message before imparting information
Contextual Information
How do the principles of grounding inform a calm computing strategy for Human/Machine communication?
Collection of Context

What context should the computer collect?
How does the computer collect the contextual information?
How does the computer convey inferences made from context?
Calm Collection of Context

Calm strategy – System collects all implicit context
Application designers can choose what pieces are relevant

**Advantages:** User does not need to provide context; User does not need to decide what is important to share

**Disadvantages:** Power; Privacy/Security implications
What is Context?

Schilit et al (1994) – Where you are, who you are with, what resources are nearby

Broken down further into:

- Computing Environment – Available processors, devices for input & display, network capacity, connectivity, costs of computing
- User environment – Location, collection of nearby people, social situation, time
- Physical Environment – Light & noise level
Implicit vs. Explicit Context

Explicit – User tells system directly about context
Implicit – System determines context automatically

Example: User identity
Face recognition – Implicit
Type in username/password – Explicit

*Implicit contexts require little human/computer communication*

1. **Presenting** information and services to a user
2. Automatic **execution** of a service
3. **Tagging** of context to information for later retrieval
Presentation of information/services comprises:

- Proximate Selection – Interaction where items more relevant to user are made easier for selection. e.g. Printers near to you
- Contextual (re)Configuration – Retrieve information relevant to my context. e.g. Weather in my location, my stocks
- Presentation of Context (Pascoe 1998) – Present the collected context to the user
Contextual Execution of Services

Context-triggered Actions (Schilit 1994) – Services executed automatically when the right combination of context exists
If/Then Rules

Contextual Adaptation (Pascoe 1998) – Execute or modify a service automatically based on current context

Intercede for user by determining proper course of action
Tagging of Context

Tagging of Context – Contextual Augmentation (Pascoe 1998)

Ability to associate digital data with the user’s context

Example: Know what documents the user sent to the printer when physically close
Context-Aware Applications
Some historical examples:

- Active Badge System (Want 1992)
- PARCTAB (1994-1995)
- Reminders – Location (Schilit 1995), Targeted (Dey 2000, Ludford 2006)
- Environmental Control – Rules based (Elrod 1993), Learning (Mozer 1998)
More recent historical examples:

- Interpersonal Communications (Avrahaimet 2008)
- Interruptibility in the office and while mobile (Fogarty 2005)
- Phone calls (Schmidt 2000)
- Health Care (Baradram 2008)
- Location Aware Systems (Many)
- Agriculture (Kjaer 2008)
- Application Personalization (Weiss 2008)
- Peripheral Displays of Information (Wisneski 1998)
Steps to design Context Aware Applications:

1. Specification – Determining what context-aware behaviors exist
2. Acquisition – Determine what sensors are needed to collect the contexts
3. Delivery – Specify how context should be delivered from sensors to the applications
4. Reception – Application “subscribes” to contexts that are of interest
5. Action – Analyzing received context to determine how to execute
Ontologies
Motivation for Ontologies

Perera 2014 – Survey of Context Aware IoT
Survey paper of 231 models, systems, taxonomies around IoT and Context Awareness

How do these connected devices communicate contexts?
Ontologies

Ontology – set of concepts and categories in a subject area with properties and relationships
Semantic Web – Web of Data – Data interchange project by W3C (WWW Consortium)

Ontologies are vocabularies for “grounding” data interchange
Example Ontology

IoT-Lite Ontology
Example Ontology

IoT-Lite Example
Challenges
Challenges to Context Awareness

Challenges for Context Awareness:

- Proxy for Human Intent
- Context Inferencing
- Context Ambiguity
- Rules vs. Machine Learning
- Privacy
- Evaluation
- End User Issues
Ideally, context awareness completely understands human intent.
Example: Interest in art/movie/book by amount of time changes by person.

Developer must properly weight and scope relevance of contexts.
Classification of user’s context from uncertainty
Data is often incomplete, or not enough to verify context
The inference must be maximally useful, without being obtrusive or complex

Weigh false positive/false negatives for usability
How expensive, bulky, and useful is the system
Ambiguity and errors
How to compensate for incorrect conclusions?
Ambiguity Strategies

Reduce Ambiguity:
Sensor Fusion – Combine multiple sources of data to improve reliability of context
Manual Disambiguation – Ask the user (often presented with best-N list)

Pretend Ambiguity Doesn’t Exist:
Accept ambiguity errors, and let the user deal with them
Attempt to recover from ambiguity errors
Rules vs. Machine Learning

Context aware often built from If/Then type rules
e.g. If I am home and if it is night, turn on the lights

Advantages:

• Easy to define
• Easy to evaluate
• Users can understand (Mental Model)

Disadvantages:

• Complex dependencies can create conflicts
• Difficult to debug when number of rules is large
• Can be inefficient to evaluate all rules
Gather data on situations where adaptation is desired
Use machine learning to apply *probabilistic* relationships

Advantages:

- Skip context-inferencing step – ML learns the contexts
- Not static, can improve recommendations

Disadvantages:

- Users may not understand rules (Mental Model)
- Large amount of data to learn
- Requires infrastructure for processing
Context-aware systems often gather a large amount of data. These data are often inferred into specific contexts. Appropriate steps need to be taken to prevent improper or malicious dissemination of these data.
Understanding – Mental Model
Context aware developers need to make sure users know why contextual adaptations are made

- Present the context
- Use human-intelligible rules
User Issues – Control

Control vs. One-size-fits-all
Users often prefer to be able to adapt systems to their usage patterns
Allow evolving usage patterns based on users preferences