A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications

Anind K. Dey, Gregory D. Abowd, and Daniel Salber.


Presented by Kareem Bedri
Anind K. Dey

- Associate Professor and the Director of the HCII at CMU.
- BSc in CE from Simon Fraser University 1993.
- MSc in AeroSpace from Georgia Tech in 1995.
- MSc and a Ph.D. in Computer Science at Georgia Tech in 2000.
- Senior Researcher at Intel Research Berkeley from 2001-2004
- Adjunct Assistant Professor in the EECS Department at UC Berkeley, 2001-2004.
Gregory D. Abowd

Regents’ Professor and J.Z. Liang Chair in the School of Interactive Computing at Georgia Tech.

B.S. in Honors Mathematics in 1986 from the University of Notre Dame

M.Sc. (1987) and D.Phil. (1991) in Computation from the University of Oxford in the United Kingdom

1989-1992 he was a Research Associate/Postdoc, the University of York

1992-1994, he was a Postdoctoral Research Associate with the Software Engineering Institute and the Computer Science Department at Carnegie Mellon University.

Dr. Abowd was elected into the CHI Academy and was named an ACM Fellow. In 2007
Daniel Salber

- He was a postdoc at Georgia Tech from 1997 to 1999.
- He has a Ph.D. in Human-Computer Interaction.
- He worked as a Research Scientist at IBM in the US and France.
- Currently he is a Senior Research Scientist at Joost and VP at Mycelia.
- He is the founder of several startups,
- Lives in Amsterdam, Netherlands.
The problem
What is context?

“Context: any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity, and state of people, groups, and computational and physical objects.”
Requirements

1. **Separation of concerns**: provide abstraction for the inputs.

2. **Context interpretation**: provided by the architecture and not the applications for a unified interpretation.

3. **Transparent, distributed communications**: to facilitate transferring context information to all parts and nodes of the system, which highlight the need for a global time clock.
Requirements

1. **Constant availability of context acquisition**: Because these components run independently of applications.

2. **Context storage and history**: to help in pattern discovery and making predictions.

3. **Resource discovery**: The system should be able to define new components and their properties and provide application access to them.
Conceptual framework

- Widgets
- Interpreters
- Aggregators
- Services
- Discoverers
Example
Example
Discussion

1- Do you agree with the author's definition of context? What would you change? Do you think this definition can apply to the current era of IoT?

2- For sure privacy is a big concern when it comes to context aware applications. The authors explained a few features in their widget design and other components that introduce security levels for user information. The authors mentioned that this is not a complete solution. What would you suggest to improve privacy protection in context aware apps?

3- What context aware applications or features you would like to see now or in the near future? Will the same requirements defined by Dey et al. still apply to them?