

A Framework for Understanding Contexts in Interactive Systems Development

Eric H SWANSON

Institute of Design, Illinois Institute of Technology
Chicago, IL 60610, USA

Adriano B GALVAO

Institute of Design, Illinois Institute of Technology
Chicago, IL 60610, USA

and

Keiichi SATO

Institute of Design, Illinois Institute of Technology
Chicago, IL 60610, USA

ABSTRACT

The context of an interactive system is not a single, unified entity. As multiple users interact with a system, engaging in multiple tasks in multiple environments, the nature of the context changes. This research proposes a framework for managing context, the Context-Influence Framework. The Context-Influence Framework defines such a system as having multiple contexts, each associated with a particular combination of users and use of the system. These combinations, called “Use Subsystems,” are affected by contextual Influences. Influences provide a means of standardizing and sharing contextual information within a team. For capturing contextual influence, context itself is divided into “Domains,” areas of interest from which elements of context can be described. Once contextual Influences are captured and described, the descriptions of the contexts themselves can be managed and manipulated through the use of three hierarchies: People, Activity and Environment.

Keywords: Context, Context-sensitive Design, Context-aware Computing and Context-Influence Framework.

1. CONTEXT IN INTERACTIVE SYSTEMS

As technology systems within organizations grow more complex, encompassing a greater variety of users, tasks and environments, the contexts in which they are used increase in both number and diversity. “Context” no longer is a single monolithic entity encompassing the entire system, but a collection of smaller, more specific contexts.

Each time a system is used, the combination of user, task and interface mechanism creates a “Use Subsystem” (Figure 1) with its own inputs and desired outputs. These subsystems are not permanent; they pop in and out of existence with each different use of the system. A change of the user, physical environment or task establishes a new Use Subsystem.

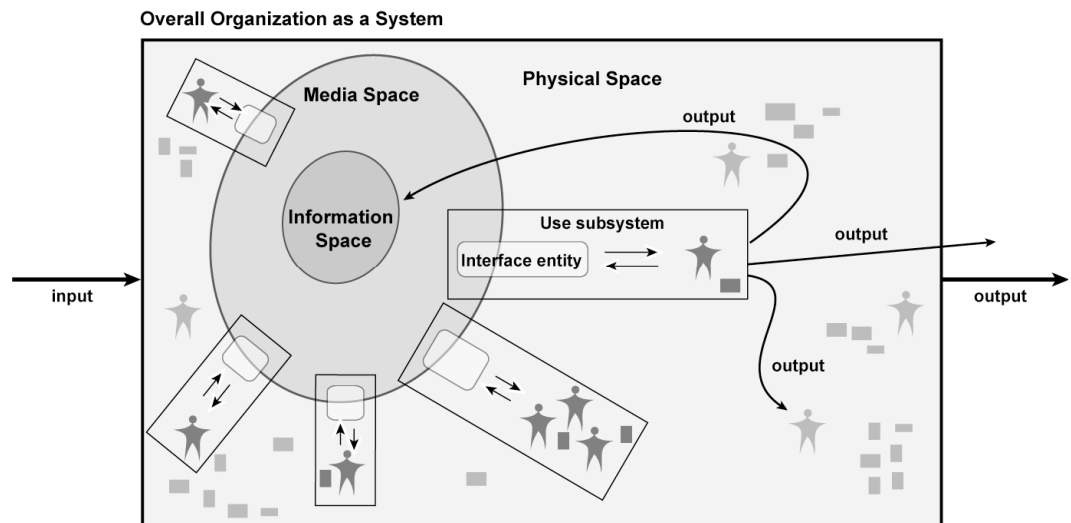


Figure 1: Use Subsystems within an organization system

These subsystems contain the both the Interface Entities used to interact with the system and the users themselves. The Interface Entities act as the bridge connecting the Information Space of raw data and algorithms to the Physical Space of objects and people through the Media Space, a combination of Information Space and Physical Space, which consists of interface hardware and the algorithms that run it.

Increasingly, desktop computers are not the only Interface Entities; other interactive devices, such as PDAs, wearable computers and bar-code readers, as well as one-way communication devices, such as automated messages, physical documents and visual displays also act as interfaces.

The success of each Use Subsystem is highly dependent on its context. With each different user, task or environment, different contextual factors influence the ability of that subsystem to generate useful output (Figure 2). These influences make up the context of a Use Subsystem.

2. CURRENT METHODS FOR MANAGING CONTEXTS

The field of systems design currently addresses context for technology systems in two major ways: broad approaches, which emphasize the persistent elements of context (such as *Contextual Design*), and local approaches, which emphasize the moment-to-moment changes of context for a single environment or interface (such as *Context-Aware Computing* and *Context-of-Use*). Global approaches emphasize longevity, seeking to isolate influences that remain constant from use to use, e.g. contextual information “remains constant during processing [but] its value changes when the process is used for another application” (Bremont & Thonnat, 2001). Local approaches emphasize the changing context of situations, e.g. “Context is any information that can be used to characterize the situation of an entity” (Dey, 2001).

An ideal framework for managing contextual information would combine the systemic overview of the global approaches with the rich understanding of situational changes in context of the local approaches.

To manage context for interactive systems design, this research proposes the Context-Influence Framework. This framework is built on four principles:

- Contexts are analyzed at the **Use Subsystem level of granularity**;
- Contexts for system design are **collections of influences**;
- Contexts can be **divided into multiple domains** of concern;
- Contexts can be **organized hierarchically** into general and specific contexts.

The resulting framework serves several beneficial functions:

- **Eases communication**: Provide common ground for cross-disciplinary communication
- **Provides flexibility**: Allow disciplines to work from within their idioms
- **Promotes completeness**: Prevent the omission of relevant contextual information
- **Manage complexity**: Enable adding of contexts without increasing complexity

3. CONTEXTS AS COLLECTIONS OF INFLUENCE

Within this framework, a context is defined as *the collection of influences or potential influences affecting the behaviors, actions and states of the system, its subsystems and its constituents.*

These influences remain stable over time for a particular Use Subsystem (capturing the persistent depicted described by the global approaches), but may change when different Use Subsystems come into effect (capturing the situational nature depicted by the local approaches).

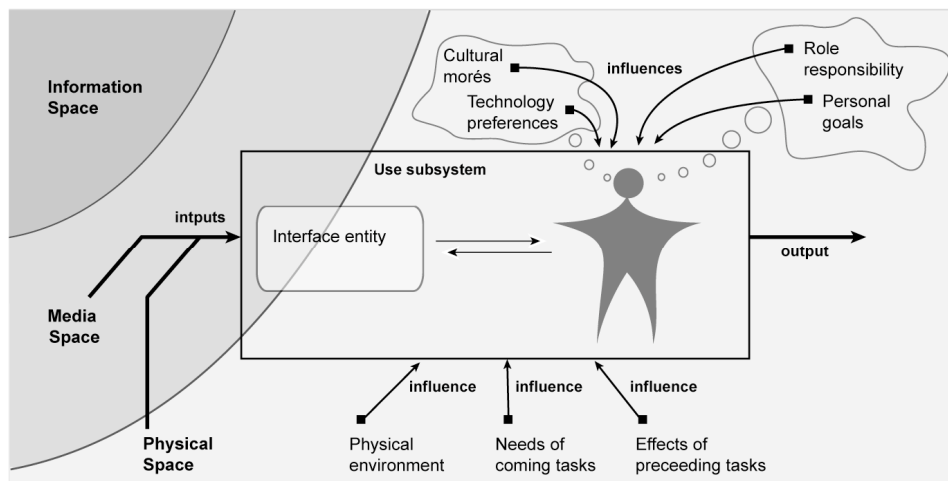


Figure 2: Influences on a Use Subsystem

Influences can be stated in the following form:

what (source of influence)
exerts what effect (nature of the change)
on what (conditions or entities).

For instance, an influence relating to the design of vehicle navigation systems may be described as “pedestrian congestion INCREASES attention to the outside of the vehicle.”

The use of influences helps resolve a common issue in researching context: differentiating a context from the objects and users it encompasses. An influence is “a power affecting a person, thing, or course of events” (American Heritage Dictionary of the English Language, 4th Edition). An influence is measured in its effects, not in its actions. If something causes changes in a system through direct action, it is not part of the context; if it indirectly causes changes in a system by altering the behaviors, choices or feelings a constituent element of the system, then it is context.

These definitions limit the need for description of context to those items that influence the subsystems. It makes the connection between context and its effects explicit. Members of a multi-disciplinary team still do their analyses in their own idioms; however, when they share information regarding, context, they are able to use the coding influences as a more common language. The full complexity of a contextual model from one discipline’s analysis does not need to be understood by all members of the team; only the output from such a model, in the form of influences, needs to be understood.

4. DOMAINS FOR DESCRIBING CONTEXTS

The complexity of representing context requires mechanisms for incorporating information from multiple viewpoints, as well as a means for avoiding omission of relevant contextual detail. To address this, the Context-Influence Framework divides the types of information that can be collected about a context into Domains. Each Domain consists of a related set of disciplines or methodologies.

Domains should be chosen to provide thorough coverage of the potentials sources of influence (Figure 3). The need for completeness of coverage outweighs the need to prevent overlap among domains.

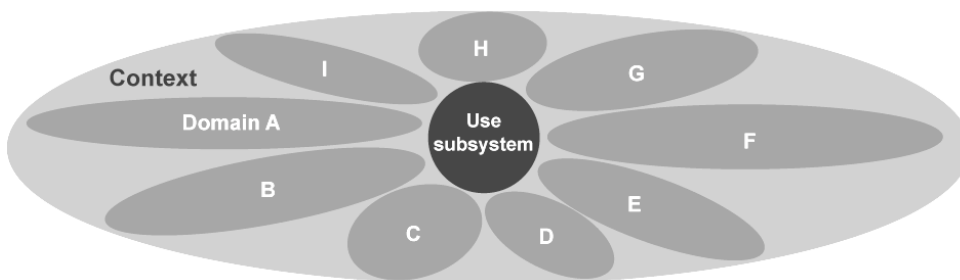


Figure 3: Domains of a context

Table 1: User-centered questions relating to sample domains

| Sample Domain | Sample Question |
|----------------|--|
| Physical | How does the environment look/smell/taste/feel? |
| Technological | What technologies does the user trust most highly? |
| Cognitive | What values do users hold towards the roles that they inhabit? |
| Social | How do users negotiate status in conversation? |
| Cultural | What are the unstated expectations of the organization for the individual? |
| Biographical | What demographic characteristics do members of the class of users share? |
| Organizational | What are the stated organizational structures surrounding the potential user? |
| Chronological | What activities typically precede the activity in question? What follows? |
| Economic | How does the cost of use compare to other things the user considers valuable? |
| Regulatory | What liability issues surround the industry in which the system is to be placed? |
| <i>etc...</i> | <i>Teams may select different or additional domains important for their projects</i> |

The division of context into domains enables the creation of tools specifically for the capture of context within the constructs of each discipline. As an illustration, we have created a sample collection of Domains (Table 1). The questions in Table 1 elicit information specific to how a context may affect the behavior, attitudes or feelings of a user.

The use of domains encourage the team to look for influences from multiple disciplines for each subsystem, even those subsystems that do not, initially, appear relevant to a particular discipline. The domains selected also serve to remind all members of the team of the information needs of other disciplines within the team.

5. CONTEXTS GENERAL AND SPECIFIC

To enable manipulation of context at multiple levels of detail, the collection of contexts is organized into three class

hierarchies: People, Activities and Environments. Each context within the hierarchy is associated with a list of the influences that affect it.

The Context-Influence Framework handles changes of contexts through the recombination of contexts within the hierarchies. The top-level context of each class hierarchy covers influences that affect every context within the hierarchy. As one navigates down each hierarchy, each context covers more specific instance of its class. Combining these more specific contexts leads to a rich description of a situation's context.

For example, imagine a hospital information system that uses PDAs to deliver patient information to doctors. A use subsystem comes into being when the doctor talks with the patient while viewing that patient's records on his PDA. The context of that Use Subsystem can be described by combining contextual information from each hierarchy: that carried by the *environment* hospital room, that carried by the *person* doctor, that carried by the *person* patient, and that carried by the *activity* doctor-patient status discussion. The combination can be described as follows:

| | |
|---|---------------|
| <i>hospital room</i> | (environment) |
| + <i>doctor-patient status discussion</i> | (activity) |
| + <i>patient</i> | (people) |
| + <i>attending physician</i> | (people) |

This combination would generate a list of influences that affect this particular use subsystem. *Hospital room* may include "physical space restricts movement," *attending physician* may include "high workload increases cognitive load;" while *doctor-patient status discussion* may include "fears of patient increase anxiety." All these factors influence this Use Subsystem, effecting how successfully it generates the desired outputs of the patient both understanding their symptoms and trusting their doctor.

As future iterations of the system expand the body of users, tasks and environments, the new contexts they create can be added to the existing set of hierarchies.

6. CONCLUSION

The Context-Influence Framework provides a structure for understanding and manipulating the contextual influences that affect interactive systems. It is intuitive, built around the familiar concept of "influences." It is flexible, providing conceptual places for different disciplines to work within their own methodology. It manages the complexity of multiple contexts, allowing the addition of new contexts through a class hierarchy without complicating the view of an individual context.

Further research on the Context-Influence Framework should explore methodologies, answering the question, "how do we gather, organize and communicate contextual information so that its influence is reflected in design decisions?" Field studies need to be conducted to test the framework and its contextual assumptions against the messiness of real data in a variety of environments. The methodological tools need to be further developed to systematically integrate the framework into the development process.

7. ACKNOWLEDGEMENTS

Portions of this research were supported by the Tangible Research Consortium at the Institute of Design, Illinois Institute of Technology. The authors also acknowledge the contribution of Eui Chul Jung, Kei Hoshi and Youn-Kyung Lim for the many discussions and debates we had over this research.

8. REFERENCES

- [1] Beyer, H. and Holtzblatt, K., **Contextual Design**, San Francisco: Morgan Kaufmann Publishers, 1998.
- [2] Brémond, F. and Thonnat, M., "Issues of Representing Context Illustrated by Video-Surveillance Applications", **International Journal of Human-Computer Studies**, Vol. 48, 1998, pp. 375-391.
- [3] Button, G., "Designing in What Context?", **Designing in Context: Proceedings of DTRS 5**, Lloyd, P. and Christiaans, H. (Eds), Delft University Press, 2001.
- [4] Dey, A. K., "Understanding and Using Context", **Personal and Ubiquitous Computing Journal**, Vol. 5, No. 1, 2001, pp. 4-7.
- [5] Hekkert, P. and van Dijk, M., "Designing from Context: Foundations and Applications of the ViP Approach", **Designing in Context: Proceedings of DTRS 5**, Lloyd, P. and Christiaans, H. (Eds), Delft University Press, 2001.
- [6] Mantovani, G., "Social Context in HCI: A New Framework for Mental Models, Cooperation, and Communication", **Cognitive Science**, Vol. 20, 1996, pp. 237-269.
- [7] Mittal, V. O. and Paris, C. L. (), "Generating Explanations in Context: The System Perspective", **Expert Systems with Applications**, Vol. 8, No. 4, 2001, pp. 491-503.
- [8] Pugh, S., **Total Design: Integrated Methods for Successful Product Engineering**, New York: Addison-Wesley Publishers Ltd., 1990.
- [9] Thomas, J. C.; Kellogg, W. A. and Erickson, T., "The Knowledge Management Puzzle: Human and Social Factors in Knowledge Management", **IBM Systems Journal**, Vol. 40, No. 4, 2001, pp. 863-84.