

---

# Adding Intelligibility to Machine Learning-Based Interactive Systems

**Anind K. Dey**

Carnegie Mellon University  
5000 Forbes Avenue  
Pittsburgh, PA 15213 USA  
[anind@cs.cmu.edu](mailto:anind@cs.cmu.edu)

**Joe Tullio**

Motorola Labs  
1295 E. Algonquin Rd.  
Schaumburg, IL 60622  
[jtullio@acm.org](mailto:jtullio@acm.org)

**Abstract**

Current interactive systems that rely on machine learning techniques have little, if any, support for intelligibility. That is, users are unable to ask why a machine learning system made a particular prediction or classification, or why it suggested a particular action be taken. In our work, we look at the need for intelligibility in machine learning (and other complex) systems, and mechanisms for providing support for intelligibility.

**Keywords**

Intelligibility, Machine Learning, Human-Computer Interaction

**Introduction**

Context-aware systems use context – information regarding the state of entities that is relevant to interaction with users [5]. Based on the context, they present useful information or perform some service on behalf of a user. A canonical example of a context-aware application is a tour guide [1]. Bellotti and Edwards state that a key design principle for context-aware systems is informing the user of the system's understandings, or *intelligibility* [4]. A study of context-aware systems showed that users become very frustrated when they do not understand why a system

has performed an action, or have the ability to fix it [3]. Interactions that support intelligibility and need to be a significant part of context-aware applications and will have a large impact on adoption. There will always be situations where users want to understand or modify application state [2].

Particularly, when systems make a mistake or perform an unexpected action, users will want to understand what happened. Consider a context-aware home lighting application that turns on lights in a home for occupants, at the same time trying to save energy costs. During normal operation this action is completely implicit, turning lights on and off based largely on user movements and object locations but not according to user commands. However, if the system performs unexpectedly or erroneously, *e.g.* turning off a light in a room where a user is reading, that user will likely shift into a set of explicit interactions with the application, perhaps trying to figure out why the system turned the lights off and almost certainly trying to turn the lights back on.

While an extreme case, evidence from the MavHome shows that the lack of an intelligibility and control interface can result in a very frustrating user experience [8]. The MavHome learned lighting behaviors over time with occupants who did not have visitors late at night. When an occupant moved in that had guests over late at night, the lights remained off, not having had time to learn the new occupant's patterns. Apparently the occupant chose to literally "remain in the dark" because there were no mechanisms for him to either understand why the home was behaving this way, or to control the home directly.

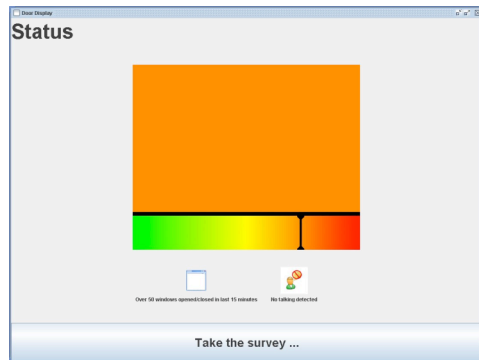
Context-aware systems are traditionally composed in two different ways: as a collection of if-then (or slightly more complex) rules, where an action is taken as a result of matching some input context; or as a machine-learning-based system which takes context as input and outputs an action to take. In the former case, intelligibility is easier to support, as, at the very least, appropriate or relevant rules can be presented to the user, to help the user form a mental model about how a system works and why it took a particular action. However, in the latter case with machine-learning-based systems, it is much more challenging to expose the underpinnings of the system to help a user form a reasonable mental model.

### **Case Study of Intelligibility**

In our work, we have begun to investigate what constitutes intelligibility for machine-learning-based context-aware systems. We deployed an intelligent system for six weeks that predicts interruptibility of an office worker to four managers in our human-resources department [7]. Our goal in this deployment was to determine how users form mental models of such systems and what the impact of providing simple forms of intelligibility was on that mental model formation.

All four managers trained the interruptibility system for three months to produce individual models of interruptibility. The system was based on the Subtle toolkit, that aids in the construction of sensor-based statistical models [6]. The models created by our managers had accuracies ranging between 93.8% and 98.2%.

Our human resources department is split between two buildings, with little interaction between the occupants



**Fig. 1:** Door display indicating interruptibility of office occupant. Color on top indicates interruptibility (in this case, not very interruptible). Images and text on bottom indicate features that led to interruptibility estimate.

of the buildings. Our managers were also split, with two managers in each building. Our subject pool consisted of eight direct reports for these managers, also with half in each building. We asked our subjects to interact with their manager(s) as usual, except to look at a door display conveying the manager's interruptibility before deciding whether to interrupt or not. In one building, the managers' door displays simply contained a graphical indication of how interruptible the manager was; we used a color scale, ranging from green (very interruptible) to red (very uninterruptible) (top Fig. 1). In the other building, the door display also included a list of the top three (human-understandable/readable) features that led to the interruptibility classification (bottom Fig. 1). We conducted weekly interviews with our subjects, where we asked them to express their mental models of the interruptibility system including: inputs to the system and priorities/weights of those inputs, any computation the system was performing on those inputs, and any experiences during the week that helped strengthen or change their pre-existing model.

Our most interesting finding was that the intelligibility we provided (*i.e.*, top three features) had little impact on the mental models that users constructed. Users described the system as being Wizard-of-Oz driven, decision trees, rule-based, and statistical/probability-based, but there was no general trend of model types within or across the two groups. In both groups, users tended to stay with their original conception of the system, and not make radical changes. For example, users who first saw the system as rule-based, maintained that system was rule-based by the end of the six-week study.

The individual features or inputs of the system did change over time, but again, was not driven by which group the user was in. Users who received the intelligibility feedback typically used the features they saw in their mental model description, however these features did not carry over in their descriptions in the following interview. All users did significantly improve, however, in the correctness of the inputs they suggested, as the study continued.

### Implications for Intelligent System Design

From this initial exploration of support for intelligibility, we have some design implications both for interfaces and for interactive machine learning systems. We had several participants that discussed the use of history, statistics, and continuous evaluation of features to arrive at interruptibility estimates. However, others could not determine that these were key factors, and needed more information about how the system worked. Any interface that is designed to support intelligibility must make these higher-level concepts clear. These concepts are necessary for both 1) correcting incomplete or incorrect models and 2) breaking users out of their original models of how the system works. As well, more explicit and clearer indications of the inputs to the system are needed to help users refine their models and produce more accurate mental models over time.

From a machine learning standpoint, as many of our users were able to describe reasonably complex models of how our system worked, we feel that there are two avenues of research that should be explored. The first is to continue to conduct explorations such as ours and see how well this finding generalizes: *i.e.*, whether users can accurately describe models for different types

of machine learning systems. The second avenue to explore, to help users that had difficulty forming reasonable mental models, is to make the underlying machine learning systems simpler (likely at the cost of performance) to improve the formation of accurate mental models.

### Future Work

We are currently exploring all of these different approaches for improving intelligibility for machine-learning based systems, both for context-aware applications and more general interactive applications. In one of our projects, we are building a system to model the routines of dual-income parents and their school-age children. We are collecting data about their locations at all times, their electronic communications, their prospective plans, and how their current plans played out. In particular, we are interested in understanding where deviations from routines occurred, the reasons why, and whether the family members involved in the deviation would have wanted technological support to help predict or recover from the deviation. As we collect the data, we are considering what information to present to users to support a particular technological intervention, which will inevitably be required to support intelligibility.

### References

[1] G.D. Abowd, C.G. Atkeson, J. Hong, S. Long, R. Kooper and M. Pinkerton. Cyberguide: A mobile

context-aware tour guide. *ACM Wireless Networks* 3(5): 421-433, 1997.

[2] M. Assad, D.J. Carmichael, J. Kay and B. Kummerfield. PersonisAD: Distributed, Active, Scrutable Model Framework for Context-Aware Services. *Proceedings of Pervasive 2007*, 55-72, 2007.

[3] L. Barkhuus and A.K. Dey. Is context-aware computing taking control away from the user? Three levels of interactivity examined. *Proceedings of Ubicomp 2003*, 149-156, 2003.

[4] V. Bellotti and K. Edwards. Intelligibility and accountability: Human considerations in context-aware systems. *Human-Computer Interaction*, 16(2-4):193-212, 2001.

[5] A.K. Dey, D. Salber and G.D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human-Computer Interaction*, 16(2-4):97-166, 2001.

[6] J. Fogarty, J. and S.E. Hudson. Toolkit support for developing and deploying sensor-based statistical models of human situations. *Proceedings of CHI 2007*, 135-144, 2007

[7] J. Tullio, A.K. Dey, J. Chalecki and J. Fogarty. How it works: A field study of non-technical users interacting with an intelligent system. *Proceedings of CHI 2007*, 31-40, 2007.

[8] G.M. Youngblood, D.J. Cook and L.B. Holder. A learning architecture for automating the intelligent environment. *Proceedings of Innovative Applications of Artificial Intelligence 2005*, 1576-1583, 2005.