

# Presence Awareness in Global Software Engineering

Anders Bech Mellson  
anbh@itu.dk

Kristian S. M. Andersen  
ksma@itu.dk

Mads D. Christensen  
mdch@itu.dk

## ABSTRACT

### Author Keywords

Activity Monitoring

### General Terms

Documentation, Theory

## INTRODUCTION

### RELATED WORK

The idea of a context aware systems have existed for a while now. The idea was embodied at Xerox PARC through the idea of smart spaces, smart rooms and other context aware technologies. These context aware systems made it possible to have the system react on a given situation, e.g. if a person was in the smart room, his work related documents would be accessible in an easy fashion. A more modern examples of this is Noosphere which is a service based activity-centric infrastructure that support the development and deployment of distributed interactive systems. It conceptualizes activity as its base of context, meaning its first class object is an activity that is minimally composed of users, meta-information, actions and resources related to that activity. Noosphere enables the development of context aware systems both through its infrastructure and because it has a built in discovery system that makes it easy to sense other devices around it.

Another example of an infrastructure that supports context aware applications is Ocon. Ocon is a context aware system built to emulate and enhance the properties of a SCRUM-board. Even though the papers proof-of-concept and motivation was the SCRUM-board, their architecture could be used for other context aware applications. Ocon is built in .NET and consists of three components (i) central (center of infrastructure, decides when and how to notify the clients), (ii) client (acts as actuators, notified when a specific event occur) and (iii) widget (translates and sends sensory input to central).

For this specific project we feel that Noosphere is too extensive for our needs and the Ocon paper have not indicated that Ocon is scalable. Because of this we have decided to create our own infrastructure.

For this infrastructure we will be using the actor model. A way to model computer programs where everything is split up into the smallest possible unit of work. These units are encapsulated in an actor. And the actors communicate by sending messages to each other. Actors does not share any state making it possible to run actors in parallel and even on different machines. This makes the actor model a good candidate for building highly scalable applications.

For evaluation we will use Gutwins awareness framework and the C3 model as points of measurement.

## TECHNICAL IMPLEMENTATION

## DISCUSSION

## CONCLUSION

## REFERENCES