Multiagent Architecture CONSORTS for Ubiquitous Computing Environment

Koichi KURUMATANI

Cyber Assist Research Center
e-mail:
k.kurumatani@aist.go.jp
AIST Today Vol. 3, No.10 (2003) 10

Ubiquitous computing is expected to realize an environment, where anyone can get benefit of information services anywhere and anytime, and to become a complex system consisting of enormous numbers of software agents working parallel in a distributed manner.

To provide various information services in such ubiquitous computing environment, we are designing and implementing multiagent architecture CONSORTS that is a middleware to organize several kinds of information services with linking digital information and real physical world.

Multiagent is an approach to describe, implement, and analyze systems as emergent phenomena through interactions among agents that have their own behavior algorithms.

One of the key concepts of CONSORTS is 'semantic grounding' that grounds objects in digital world to real physical world. Another key concept is 'service coordination', by which we can flexibly access information services through service composition.

The application images in CONSORTS cover wider areas discussed in other approaches in ubiquitous computing. For instance, it includes 1) social resource allocation: coordination among users' behaviors in traffic system or appointment system for public services by spatio-temporal resource allocation, and 2) information circulation with semantic formalization by ontology that is general dictionary of concepts in the sense of information science.
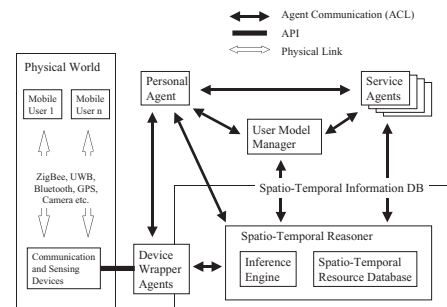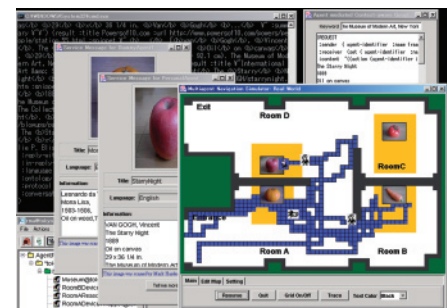

Fig.1 CONSORTS architecture (Version 1)

Fig.2 Demonstration image of museum scenario

System Verification using Formal Method
## *- A new verification method of information processing systems -*

Kouji HAYAMIZU

Laboratory for Verification and Semantics
e-mail:
kouji-hayamizu@aist.go.jp
AIST Today Vol. 3, No.10 (2003) 11

We are developing techniques of system verifications using formal methods in the designing phase of software development in industry. As a case study, we applied a model checking technique to specifications of an embedded system under development in a company. Model checking was proceeded in parallel with the designing process in the company and was completed in a reasonably short term. As a result of this collaboration, we could successfully detect six unknown bugs. Consequently our technique could contribute to improve the quality of the developing system and to cut down its developing cost. Thus this case study shows that an application of model checking technique to software specifications is feasible and effective.
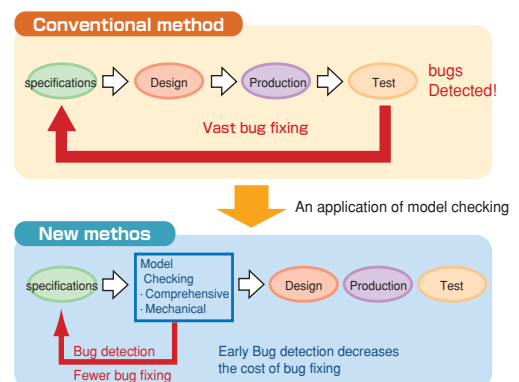
Comparing a conventional method and a new method