Amelia

Reid Simmons **Sebastian Thrun** **Greg Armstrong** **Richard Goodwin**
Karen Haigh **Sven Koenig** **Shyjan Mahamud** **Daniel Nikovski** **Joseph O'Sullivan**

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3891
mahamud@cs.cmu.edu

Amelia was built by Real World Interface (RWI) using Xavier—a mobile robot platform developed at CMU on a B24 base from RWI—as a prototype. Amelia has substantial engineering improvements over Xavier. Amelia is built on a B21 base. It has a top speed of 32 inches per second, while improved integral dead-reckoning insures extremely accurate drive and position controls.

The battery life is six hours, and are hot-swappable. Two Pentium-100s are the main CPU's on board with special shock-mounted hard drives. A 75Mhz 486 laptop acts as onboard console. All these are interconnected by an internal 10M Ethernet, and to the world via a 2MBs Wavelan wireless system. Sonar and infrared sensor arrays ring the robot, mounted on Smart Panels for quick and easy access to internal components. These Smart Panels also contain bump sensors. A Sony color camera is mounted on a Directed Perception pan/tilt head for visual sensing. Finally, an arm can extend for 4-degree of freedom manipulation of Amelia's world.

Like Xavier, Amelia has a distributed, concurrent software system, which runs under the Linux operating system. All programming is done in C, and processes communicate and are sequenced and synchronized via the TASK CONTROL ARCHITECTURE (TCA) (Simmons 1995).

Communication with Amelia is graphical (via the laptop), remote (via zephyr), and speech-driven. An off-board Next computer runs the SPHINX real-time, speaker-independent speech recognition system and a text-to-speech board provides speech generation. Thus, we can give verbal commands to the robot and the robot can respond verbally to indicate its status. In addition, a graphical user interface is available for giving commands and monitoring the robot's status.

Amelia plans to participate in the "Call a meeting" event. She will use the ROGUE system (Haigh & Veloso 1996) for high-level task planning through PRODIGY, a planning and learning system (Veloso et al. 1995). ROGUE is able to create a branching plan based on the observations of the real world, select appropriate orderings of goal locations, and monitor the robot's progress towards those goals. Planning actions are defined at the granularity of "goto-location" and "observe-room", requiring more detailed schema for execution.

Navigation is accomplished using a Partially Observable Markov Decision Process (POMDP) model of the environment (Simmons and Koenig 1995). POMDP models allow the robot to account for actuator and sensor uncertainty and to integrate topological map information with approximate metric information. They also allow the robot to recover gracefully if they are uncertain about their current location.

Amelia will use vision for detecting doorways as well as for detecting faces. Doorways are detected without the help of markers. Detection is accomplished with a combination of region-growing, appearance-based matching and sonar range readings. Faces are detected in two stages. The first stage uses a fast color histogram technique to identify candidate regions that have appropriate sizes and shapes. The second stage consists of a computationally costlier step that verifies whether the chosen regions are face-like or not.

References

Simmons, R., and Koenig, S. 1995. Probabilistic Robot Navigation in Partially Observable Environments. In Proceedings of the IJCAI, 1080–1087.

Veloso, M.; Carbonell, J. C.; Perez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence* 7(1).

Haigh, K. Z., and Veloso, M. 1996. Interleaving Planning and Robot Execution for Asynchronous User Requests. In the Proceedings of the AAAI-96 Spring symposium on Planning with Incomplete Information for Robot Problems.

Simmons, R. 1995. Towards Reliable Autonomous Agents. In AAAI Spring symposium on Software Architectures.