

# Analyse der System-Software für einen Medizin-Roboter

**Projektleiter**  
Prof. Dr.-Ing.  
Michael Ludvik

**Zeitraum**  
Januar–August 2004

**Kooperation**  
Firma carat robotic  
innovation GmbH,  
Dortmund

**Förderung**  
carat GmbH,  
Dortmund,  
Fachhochschule  
Dortmund,  
Forschungssemester,

**Kontakt**  
Prof. Dr.-Ing.  
Michael Ludvik,  
Fachbereich  
Informations-  
und Elektrotechnik,  
Fachhochschule  
Dortmund,  
Sonnenstraße 96,  
44139 Dortmund,  
Telefon:  
(0231) 9112-241,  
E-Mail: ludvik@  
fh-dortmund.de

## Beschreibung

### Zielsetzung

In Zusammenarbeit mit der Firma carat robotic innovation GmbH in Dortmund sollten, ausgehend von den speziellen Anforderungen für einen Medizin-Roboter, Ansätze für die Entwicklung von Systemsoftware analysiert und untersucht werden. Da es sich bei Medizin-Robotern um Echtzeitsysteme mit sicherheitskritischen Fragestellungen handelt, kommt der Systemsoftware eine bedeutende Rolle zu. Welche Methoden des Software-Engineerings relevant sind, um in einem sicherheitskritischen und unter Echtzeitbedingungen laufenden System verwendet zu werden, sollte in diesem Projekt erörtert werden.

### Arbeitsprogramm

Zunächst wurden die Anforderungen an einen medizinischen Roboter untersucht und aufgelistet. Grundsätzlich wurde festgestellt, dass keine gravierenden Unterschiede zu anderen sicherheitskritischen Echtzeitsystemen bestehen, die im allgemeinen auch als Embedded Systems bezeichnet werden. Hierbei muss allerdings erwähnt werden, dass Embedded Systems nicht grundsätzlich echtzeitfähig sind.

Für die Entwicklung von Softwaresystemen bietet sich die objektorientierte Modellierung an. Hier hat in den letzten Jahren die Unified Modelling Language (UML) deutlich an Bedeutung gewonnen. Die UML ist eine formale Sprache zur objektorientierten Spezifikation, Visualisierung, Konstruktion und Dokumentation von Softwaresystemen. Allerdings gibt es Probleme der Einbindung von Echtzeitbedingungen, dies war in der damals vorliegenden Version noch nicht vorgesehen und wurde erst mit der UML-Version 2.0 eingeführt. Da die UML eine Reihe von Diagrammen zur Verfügung stellt, sollte geprüft werden, welche Diagrammtypen unbedingt notwendig sind. Hier reicht die Bandbreite von den Anwendungsfalldiagrammen (use cases) bis zu den Verhaltensdiagrammen, die die dynamischen Sachverhalte innerhalb eines objektorientierten Modelles charakterisieren. Gerade die Verhaltensdiagramme, wie Sequenzdiagramm, Zustandsdiagramm und Aktivitätsdiagramm sind dem Entwicklungsingenieur aus seiner beruflichen Praxis bekannt und zeigen anschaulich die Zusammenhänge innerhalb des Softwaresystems.

Die Realisierung auf einer Hardwareplattform bedarf des Einsatzes eines Betriebssystems. Das verwendete Betriebssystem muss die objektorientierte Denkweise unterstützen und sollte nach Möglichkeit Schnittstellen besitzen, den Meldungsverkehr zwischen den Objekten zu Debugging-Zwecken aufzuzeichnen oder anzuzeigen. Eine Überwachung der Systemressourcen seitens des Betriebssystems wäre außerdem wünschenswert.

### Ergebnisse

Nach Abarbeitung des Anforderungskatalogs und der besonderen Fragestellung bezüglich des sicherheitskritischen Echtzeitbetriebes fiel die Wahl auf das Betriebssystem OSE® der Firma ENEA Embedded Technology GmbH, Ismaning. Als UML-Tool wurde das „Realtime-Studio“ der Firma ARTISAN Software Tools GmbH, Köln, eingesetzt. Dieses Tool beinhaltet eine Erweiterung der UML zur Echtzeitfähigkeit und neben der benutzerfreundlichen Bedienoberfläche auch ein besonderes Datenbanksystem zur Nachhaltigkeit der Entwicklung. Im Rahmen des Projektes wurde an der FH Dortmund zusammen mit der Firma ARTISAN ein Workshop „Embedded Softwareentwicklung live – von der Theorie zur Praxis“ im Juli 2004 in Dortmund durchgeführt.



Abb.: Dr. Grube und Prof. Ludvik an einer innovativen Roboterzelle in der Produktionshalle der Firma carat, Dortmund