

B. Zusammenfassung der Ergebnisse

Das Hauptergebnis dieser Arbeit ist der modular erweiterbare Rahmen ECL, der die Erstellung einer Koordinationssprache für die Entwicklung verteilter Applikationen mit einem zugehörigen Laufzeitsystem ermöglicht. Der Rahmen enthält keine Beschränkungen bzgl. der unterstützbaren Zielsysteme und Koordinationsparadigmen und eignet sich damit als Grundlage für beliebige Applikationsklassen.

Die Freiheit von Beschränkungen ergibt sich daraus, dass ECL kein allgemeines semantisches Modell besitzt, auf das alle Erweiterungen abgebildet werden müssen. Die einzige Einschränkung ist syntaktisch und besteht aus dem Einsatz einer Architekturbeschreibungssprache, so dass eine verteilte Applikation stets als statische Software-Architektur darstellbar ist.

Der Rahmen wird durch die Kombination mit Erweiterungsmodulen zu einem Entwicklungssystem, mit dem ein Applikationsentwickler verteilte Applikationen unter Einsatz eines grafischen Editors implementieren kann. Ein Erweiterungsmodul kann sowohl bestimmte Zielsysteme als auch bestimmte Koordinationsparadigmen unterstützen.

Hierfür enthält ein Erweiterungsmodul sowohl syntaktische als auch semantische Erweiterungen, die in den Rahmen integriert werden. Die syntaktischen Erweiterungen bestehen aus neuen Typen für Architecturelemente. Die semantischen Erweiterungen liegen in Form von Java-Klassen vor. Diese werden benutzt, um eine Applikationsarchitektur zu einem Geflecht von Java-Objekten zu übersetzen, welche dann zur Laufzeit die Koordination der einzelnen Applikationskomponenten durchführen.

Bei der Erstellung neuer Erweiterungsmodule können unter Einsatz von Vererbungsmechanismen vorhandene Erweiterungsmodule benutzt werden. Dies ermöglicht die Erstellung abstrakter Module, welche ein allgemeines Koordinationsparadigma unterstützen. Anhand verschiedener konkreter Implementierungen wurden Erfahrungswerte gewonnen, die als Entwurfsrichtlinien formuliert wurden.

Die allgemeine Praxistauglichkeit des System für den Anwendungsentwickler wurde anhand mehrerer Fallstudien evaluiert. Diese stammen aus zwei Applikationsklassen: verteilte Informations- und Kontrollsysteme und rechenintensive numerische Applikationen.

Es konnte demonstriert werden, dass sich aufgrund der komfortablen grafischen Entwicklungsumgebung der Entwicklungsaufwand für den Applikationsentwickler deutlich reduziert, da er sich nur wenig mit technischen Implementierungsdetails beschäftigen muss. Der durch den Einsatz des Systems zusätzlich auftretende Rechen- und Kommunikationsaufwand war bei allen Fallstudien vernachlässigbar.

B. Zusammenfassung der Ergebnisse