Parametrisierte Verträge zur Protokolladaption bei Software-Komponenten

Ralf H. Reussner

Lehrstuhl Informatik für Ingenieure und Naturwissenschaftler,
Universität Karlsruhe (TH)

reussner@ira.uka.de

Parametrisierte Verträge behandeln eine für die Praxis wichtige Klasse von Adaptionen von Software-Komponenten. Eine der wichtigsten Fragen beim Entwurf von Software-Komponenten ist die nach dem Funktionsumfang der Komponente. Eine umfangreiche Funktionalität läßt viele Kunden ihre spezifischen Funktionsanforderungen als eine Teilmenge der Komponentenfunktionalität wiederfinden. Allerdings erhöht eine hohe Funktionalität der Komponente in der Regel auch ihren Bedarf an externen Ressourcen und verringert so die Wiederverwendbarkeit. Die Hinzunahme von Funktionalität führt also zu einem Dilemma bezüglich der Wiederverwendbarkeit. Analog tritt bei der Verwendung der Komponente in der Praxis das Problem auf, daß nur eine Teilmenge der Funktionalität tatsächlich benötigt wird; dennoch aber alle benötigten Ressourcen der Komponente vorhanden sein müssen, auch wenn sie eigentlich nicht zur Erfüllung der benötigten Funktionalität beitragen. Hier lautet die Frage: Wie kann in Abhängigkeit von der benötigten Funktionalität die tatsächlich benötigte Funktionalität ermittelt werden? Umgekehrt gibt es bei der Eingliederung einer neuen Komponente in ein bestehendes System die Frage, welche Funktionalität diese Komponente in dem bestehenden Kontext erbringen kann. Dieselbe Frage stellt sich bei fehlertoleranten Systemen nach dem Ausfall einiger –aber nicht aller– Ressourcen.

Parametrisierte Verträge verwenden zur Beantwortung dieser Fragen den Zusammenhang zwischen Angebots- und Bedarfsschnittstelle einer Software-Komponente. Hierbei nutzt man den in der Praxis relevanten Fall, daß eine Komponente oft noch eine Teilfunktionalität erbringen kann, auch wenn nicht alle Ressourcen zur Verfügung stehen. Umgekehrt werden nicht alle Ressourcen benötigt, wenn nicht die Gesamtfunktionalität erbracht werden muß. Parametrisierte Verträge stellen eine Verallgemeinerung „klassischer“ Verträge dar. Die Nachbedingung (Angebotsschnittstelle) geht als Parameter in die Berechnung der Vorbedingung (Bedarfsschnittstelle) ein und umgekehrt.

Parametrisierte Verträge fordern vom Schnittstellenmodell lediglich die explizite Modellierung der Angebots- und der Bedarfsschnittstelle, sind aber ansonsten unabhängig von der Information, die in diesen Schnittstellen modelliert wird.

Protokollbehaftete Schnittstellen beschreiben im Vergleich zu herkömmlichen Schnittstellen (Signaturlisten) zusätzlich eine relevante Klasse von Zusammenhängen zwischen Methoden. In der Praxis sind bei der überwiegenden Zahl von Komponenten nicht alle Dienste in jedem Zustand verfügbar. Dies führt zu illegalen Aufrufreihenfolgen von Diensten, die durch die Kompatibilitätsprüfung herkömmlicher Schnittstellen nicht erkannt werden können. Die legalen Aufrufreihenfolgen bilden das Protokoll einer Komponente. Allein Dokumentation von Protokollen in der Schnittstelle ist nicht ausreichend; erwünscht ist eine automatische Überprüfung der Protokollkompatibilität zweier kooperierender Komponenten. Dazu muß neben der Angebotsschnittstelle auch die Bedarfsschnittstelle mit Protokollinformation ausgestattet sein. Die Modellierung dieser Schnittstelleninformation erfordert in der Praxis ein Modell, bei dem Inklusion und Äquivalenz effizient statisch entscheidbar sind. Dies wird von bestehenden Ansätzen wie Prozessalgebren oder Petri-Netzen nur unzureichend unterstützt. Weiterhin muß das Modell auch auf in der Praxis wichtigen Abstrakten Datentypen wie Kellerspeicher anwendbar sein. Allgemein betrifft dies alle Komponenten, die Ressourcen vor Benutzung stückweise allozieren müssen und wieder freigeben müssen.

Die Zusammenhänge zwischen Methoden, die Allozieren, Benutzen oder Freigeben, sind durch endliche Automaten in der Regel nicht beschreibbar. In dem Vortrag wird ein Schnittstellenmodell vorgestellt, welches Komponentenprotokolle durch eine Erweiterung endlicher Automaten mit Zählern spezifiziert.

Es wird beschrieben, wie parametrisierte Verträge auf dieses Schnittstellenmodell angewandt werden. Konkret werden also Algorithmen angegeben, die aus dem Protokoll der Angebotsschnittstelle das Protokoll der Bedarfsschnittstelle berechnen und umgekehrt. Weiterhin werden Ansätze und Werkzeuge vorgestellt zur Generierung der Schnittstelleninformation aus annotiertem Code oder aus Entwurfsdokumenten, um das formale Schnittstellenmodell für den Nutzer transparent werden zu lassen.