

Zusammenfassung

Bei Software-basierten Systemen kann man derzeit beobachten, dass ihr Verhalten teilweise von organischen Systemen inspiriert wird. Bestehende Ansätze haben versucht, ein Systemverhalten zu entwickeln, das zur Laufzeit auf variierende Umweltbedingungen reagieren kann. Manchmal können sie sogar aus früheren Zuständen und Adaptionen lernen. In der Echtzeitdomäne sind Systeme bei der Anpassung an veränderliche Umgebungen stärker durch eine Auswahl vordefinierter Verhaltensweisen eingeschränkt. Nur diese vordefinierten Verhaltensweisen können zur Laufzeit zur Optimierung der Systemleistung angewendet werden. Neue Verhaltensweisen können zur Laufzeit jedoch nicht hinzugefügt werden. Dies begrenzt die Fähigkeit des Systems, auf neu auftretende Umweltveränderungen zu reagieren. In unserer Arbeit entwickeln wir einen Ansatz, der in der Lage ist, zur Laufzeit zu reagieren und alle Echtzeit-Einschränkungen zu respektieren. Reaktionen können zur Laufzeit in dem Sinne auftreten, dass neue Tasks hinzugefügt oder bestehende aktualisiert werden.

Unser Ansatz kann auf alle Arten von Echtzeitsystemen angewendet werden. Er besteht aus einem Adaptionalgorithmus, der seinerseits als Echtzeittask fungiert.

Unter der Annahme, dass die Tasks der Echtzeitumgebung in Form von mehreren Varianten existieren, beinhaltet der Adaptionalgorithmus einen Auswahlmechanismus für alle Tasks im System. Dies geschieht, indem die aktuelle Variante aktiver Tasks potentiell ausgetauscht wird, um die geforderten Echtzeitrestriktionen garantieren zu können und gleichzeitig globale Systemziele zu optimieren, wann immer eine Anpassung erforderlich ist. Die Anpassung kann das Hinzufügen oder Aktualisieren von Tasks oder einer Gruppe von Tasks sowie das Löschen von Tasks umfassen. Die Optimierung zielt darauf ab, unter der Restriktion alle Echtzeitanforderungen zu erfüllen eine Kostenfunktion zu reduzieren. Dem Konzept des "Organic Programming" folgend nutzen wir das Konzept der Zellen. Eine Zelle ist eine Erweiterung einer Task. Jede Zelle kann mehrere Varianten haben und kann mehrere Aktualisierungen erfahren, wobei jede Aktualisierung wiederum aus mehreren Varianten bestehen kann. Alle Varianten einer Zelle haben die gleiche grundlegende Funktionalität, jedoch unter unterschiedlichen Zeitanforderungen und unterschiedlichen Kosten. Wir weisen nach, dass der Algorithmus begrenzt ist und schätzen die Zeitkomplexität analytisch ein.