

Abstract

Distributed control systems in industries are now being increasingly heterogeneous in nature. To deal with the complexity of developing software for this realm IEC 61499 evolved. It allows flexible distribution of the software components and reconfiguration of the definition of the components and their interactions. To fully use these features ensuring optimal resource usage and satisfaction of non-functional requirements, especially the time constraints, an algorithm is needed. This work presents an algorithm for this purpose. Further, this algorithm for finding a feasible as well as optimal deployment of the components needs to be integrated into a development process to ensure that the characteristic information of the application as well as the non-functional requirements can be easily collected. Therefore, a process model for development of IEC 61499 compliant applications is also presented. Apart from usual design activities the process model introduces use of model-based transformations to enable automatic generation of codes and analysis models. The analysis models are needed to analyze the temporal aspect of the application's behaviour. The process model also includes, apart from testing activities from Object Oriented software domain, a simulative validation approach to validate the deployment proposed by the deployment algorithm with respect to the temporal perspective.

Die in der industriellen Praxis eingesetzten verteilten Steuerungssysteme weisen eine zunehmend heterogene Struktur auf. Die Norm IEC 61499 wurde eingeführt um die Komplexität der Softwareentwicklung für solche Systeme beherrschbar zu machen. Insbesondere die flexible Verteilung von Softwarekomponenten auf Rechenknoten und die Rekonfiguration der Anlage zur Laufzeit können als Innovation der Norm angesehen werden. Zur optimalen Ressourcen-Ausnutzung und zur garantierten Einhaltung nicht-funktionaler Anforderungen, insbesondere Echtzeit-Anforderungen, wird ein Algorithmus benötigt, der die Verteilung der Softwarekomponenten vornimmt. Die vorliegende Arbeit schlägt einen solchen Algorithmus vor. Um den Algorithmus mit den notwendigen Daten versorgen zu können, wird ein Entwurfsprozess benötigt in den der Algorithmus eingebettet ist und der dafür sorgt, dass die notwendigen Daten und einzuhaltenden nicht-funktionalen Anforderungen während des Entwurfs geeignet spezifiziert werden können. Für die IEC 61499 wird ein solcher Entwurfsprozess vorgestellt. Neben den üblichen Entwurfschritten führt das Prozessmodell modellbasierte Transformationen ein, mit denen Quellcode und Analysemodelle erzeugt werden können. Die Analysemodelle werden zur Analyse des temporalen Verhaltens der Applikation benötigt. Neben den Testmöglichkeiten im Rahmen von objektorientierter Softwareentwicklung bietet das Prozessmodell auch die Möglichkeit die Einhaltung temporaler Nebenbedingungen simulativ zu überprüfen und somit die vom Zuordnungs-Algorithmus vorgeschlagene Verteilung zu validieren