

Abstract

The dissertation focuses on the problem of automatic generation of robotic system controllers out of a formal specification. The main objective of the dissertation was to develop a universal method of specification of robotic systems based on the embodied agent approach (EAA) and Model-Driven Engineering (MDE). EAA imposes on the system an adequate structure and provides the necessary concepts derived from robotics. Petri nets express the activities of the system. MDE merges the two and thus produces a system model that can be used at different stages of system development, e.g. verification and code generation. Another goal was to decide at what level of detail a robotic system specification should be developed. On the one hand, developing a specification that is too detailed may involve a cost that exceeds the direct development of the controller code. On the other hand, a specification that is not detailed enough will not provide the necessary guidance in the implementation phase. Moreover, the goal was to propose a specification method enabling the analysis of robotic system models. All three dissertation objectives have been attained.

The central element of the dissertation is a parameterised meta-model that describes a generic robotic system taking into account both its structure and activities. The meta-model is represented as a 6-layer Robotic System Hierarchical Petri Net (RSHPN). The required system model emerges from the meta-model by appropriately parameterising the RSHPN. Each layer describes the activities of the robotic system at a different level of abstraction. This guarantees the separation of concerns. The Robotic System Specification Language (RSSL) was developed to streamline the process of parameter definition. The specification developed in RSSL undergoes a two-phase translation process. In the first phase, based on the parameters provided in the RSSL specification and the RSHPN meta-model, an RSHPN model is generated. The generated model is subjected to analysis. Despite the complexity of robotic systems, due to the imposed structure of the RSHPN, the network analysis is surprisingly simple. In the second phase, the RSHPN model is extended by the user with additional source code (e.g. Python/C++) and then automatically converted into the target language code (using the developed RSHPN Tool). The proposed parameterisation enables the developer to focus exclusively on the concepts derived from robotics. It also greatly simplifies the process of creating a robotic system, as it provides a clear separation of what details need to be defined by the developer and what is imposed by the general design pattern. The resulting single RSHPN model is used both to verify certain system properties and to automatically generate controller code. The proposed approach is exemplified by specification and implementation of four different robotic systems.

Keywords: Embodied Agent Approach; Model Driven Engineering; Robotic System Hierarchical Petri Net; Meta-model parameterisation; Robotic System Specification Language; Controller code generation

Automatyczna generacja sterowników robotów na podstawie specyfikacji

W rozprawie podjęto problem automatycznej generacji sterownika systemu robotycznego na podstawie formalnej specyfikacji. Głównym celem rozprawy było opracowanie uniwersalnej metody specyfikowania systemów robotycznych wykorzystującej koncepcję agenta upostaciowionego (EAA) oraz Model Driven Engineering (MDE). EAA narzuca systemowi odpowiednią strukturę i dostarcza niezbędnych pojęć zaczerpniętych z robotyki. Sieci Petriego wyrażają czynności wykonywane w systemie. MDE łączy oba podejścia i prowadzi do powstania modelu systemu, który może być użyty na różnych etapach rozwoju systemu, np. podczas weryfikacji oraz generacji kodu. Kolejnym celem było ustalenie na jakim poziomie szczegółowości powinna być opracowana specyfikacja. Z jednej strony, zbyt szczegółowa specyfikacja może wiązać się z kosztami przekraczającymi bezpośrednie stworzenie kodu sterownika. Z drugiej strony, specyfikacja, która nie jest wystarczająco szczegółowa, nie zapewni niezbędnych wskazówek w fazie implementacji. Dodatkowym celem było zaproponowanie metody specyfikacji umożliwiającej analizę modeli systemów robotycznych. Wszystkie trzy cele rozprawy doktorskiej zostały osiągnięte.

Centralnym elementem rozprawy jest sparametryzowany meta-model opisujący ogólny system robotyczny, uwzględniający jego strukturę oraz sposób działania. Meta-model wyrażony jest za pomocą 6-warstwowej hierarchicznej sieci Petriego (RSHPN). Wymagany model systemu tworzony jest z meta-modelu poprzez odpowiednie sparametryzowanie RSHPN. Każda warstwa RSHPN opisuje działanie systemu robotycznego na innym poziomie abstrakcji. W celu usprawnienia definiowania parametrów opracowano Robotic System Specification Language (RSSL). Specyfikacja w RSSL przechodzi dwufazowy proces translacji. W pierwszej fazie, na podstawie parametrów zawartych w specyfikacji RSSL i meta-modelu RSHPN, powstaje model RSHPN. Wygenerowany model poddawany jest analizie. Mimo złożoności systemów robotycznych, ze względu na narzuconą strukturę RSHPN, analiza sieci jest prosta. W drugiej fazie model RSHPN jest rozszerzany przez użytkownika o dodatkowy kod źródłowy (np. Python/C++), a następnie automatycznie przekształcany w kod w języku docelowym (przy użyciu opracowanego RSHPN Tool). Proponowana parametryzacja umożliwi projektantowi skupienie się wyłącznie na koncepcjach wywodzących się z robotyki. Zapewnia wyraźne oddzielenie szczegółów, które muszą być zdefiniowane przez projektanta, od tych, które są narzucane przez ogólny wzorzec projektowy. Powstały model RSHPN jest wykorzystywany zarówno do weryfikacji pewnych właściwości systemu, jak i do automatycznej generacji kodu sterownika. Proponowane podejście zostało zilustrowane na przykładzie czterech różnych systemów robotycznych.

Słowa kluczowe: Agent upostaciowiony; Inżynieria oparta na modelu; Hierarchiczna sieć Petriego systemu robotycznego; Parametryzacja meta-modelu; Język specyfikacji systemu robotycznego; Generacja kodu sterownika