

発表概要

UML モデルの C 言語実装における TECS の適用事例

石川 拓也^{†1} 安積 卓也^{†1,†2} 一場 利幸^{†1}
柴田 誠也^{†1,†2} 高田 広章^{†1}

本発表では、UML によって設計された組み込みソフトウェアを C 言語で実装する過程の事例として、組み込みシステム向けコンポーネントシステムを用いた倒立二輪ライントレースロボットの制御ソフトウェアの開発について述べる。組み込みソフトウェアは、大規模化・複雑化が進んでおり、生産性の向上が求められている。C++ や Java などのオブジェクト指向言語を用いたソフトウェア開発では、生産性を向上させるために、UML を用いた設計が行われている。UML はオブジェクト指向を意識したモデル言語であるため、クラス図などの UML モデルからソースコードの自動生成が容易で、モデルとソースコードとの一貫性を保ちやすい。一方、組み込みシステムのソフトウェア開発では、実装言語として手続き型言語である C 言語が広く用いられている。C 言語はオブジェクト指向言語ではないため、UML モデルから C 言語ソースコードを自動生成するツールを利用したとしても、モデルとソースコードの間に隔りがある。この問題を解決するために、組み込みシステムに適したコンポーネントシステムである TECS を利用した。TECS は独自のコンポーネント図を入力とし、コンポーネントの結合部分および、コンポーネントの提供する機能のテンプレートの C 言語ソースコードを自動生成する。自動生成により、コンポーネント図とソースコードの一貫性を保つことができる。本事例により、UML モデルのクラス図から TECS のコンポーネント図への変換方法および、コンポーネント図とソースコードの一貫性を保持できることを示す。

UML Model to an Implementation in C Using TECS: An Application Case Study

TAKUYA ISHIKAWA,^{†1} TAKUYA AZUMI,^{†1,†2}
TOSHIYUKI ICHIBA,^{†1} SEIYA SHIBATA^{†1,†2}
and HIROAKI TAKADA^{†1}

In this presentation, we will describe the development of control software for a two-wheeled self-balancing and line-tracing robot using a component system for embedded systems, as the case study of an embedded software design in UML and an implementation in C. Embedded systems have become large-scale and complex. Therefore, improving software productivity has been demanded. In software development using an object-oriented language, UML is often used in software design to improve software productivity. Since UML is based on object-oriented concepts, it is easy to automatically generate source codes from a UML model. As a result, it is easy to keep a consistency between a UML model and source codes. Meanwhile, in embedded software development where C language is generally used, UML has not been an effective solution for improving productivity because C language is not the object-oriented language. There can be gaps between a UML model and C source codes, even if automatic generation tools are used. To fill the gaps, we have taken advantage of TECS, which is a component system suitable for embedded systems. TECS takes an original component diagram as an input, and generates C source codes, that describe connections between components and templates of functions that components provide. Therefore, it is possible to keep a consistency between a TECS component diagram and source codes. This case study shows a way to transform a UML class diagram into a TECS component diagram, and that TECS enables to keep the consistency.

(平成 22 年 3 月 15 日発表)

^{†1} 名古屋大学大学院情報科学研究科

Graduate School of Information Science, Nagoya University

^{†2} 日本学術振興会特別研究員

Research fellow of the Japan Society for the Promotion of Science