

## 審査結果の要旨

論文提出者氏名 熊 英飛

ソフトウェア開発においては、問題の仕様記述からプログラムによる処理の実現にいたるさまざまな段階で、それぞれの抽象度の水準に合わせたモデルの記述が行われる。当然のことながら、異なる段階で使われるモデル間には密接な関係があり、それらは相互に矛盾なく維持すべきものである。それには、ある水準のモデル上で変更を行うと、その変更を他のモデルにも反映させて全体としての一貫性を保持する必要がある。このような処理をモデル同期(model synchronization)という。モデルを記述するための言語はこれまでも提案され実用化されているが、それらにはモデル同期に関する機能を考慮したものはほとんど存在しない。このようなことから、近年、モデル同期の記述法とそれに基づく同期機構の必要性の認識が高まってきている。

本論文は “A Language-based Approach to Model Synchronization in Software Engineering” (ソフトウェア工学におけるモデル同期に関する言語論的研究)” と題し、英文で書かれ全8章から成る。本論文では、上に示したようなソフトウェア工学におけるモデル同期の課題に対して、モデル記述のための言語の設計と同期機構の実現に関して新たな方向を論じ、実際のシステムの構築法をも提示してその有効性を示したものである。

第1章 Introduction では、本研究の背景、動機を述べ、関連研究との関係に触れつつ本論文の貢献を示している。モデル同期を off-site と on-site の2種類に区別し、それぞれに適した記述言語の設計とシステムの構築を論じていると述べている。Off-site は異なるアプリケーションで扱われるモデル間の同期であり、on-site は同一アプリケーションによるものである。

第2章 Requirement of Model Synchronization では、モデル同期に要求される条件として、無矛盾性(consistency)、保存性(preservation)、および安定性(stability)を定義している。さらに、off-site と on-site のそれぞれの同期のモデルを定義して、それらの同期において満たすべき性質を形式的に扱っている。

第3章 Representing Models and Updates では、モデルの表現についてまとめている。ソフトウェア工学におけるモデルはMOF(Meta Object Facility)によって定義されるが、これはきわめて大きい枠組みであり、ソフトウェア開発者がさまざまな領域でモデルを開発するには適しているものの、モデルを研究する立場にとっては適しているとはいえない。本章では、モデルの表現法とモデルの更新を表現する簡便な記法として、キーから値への写像を表す dictionary を導入して、その表現が安定的であることを証

明している。

第4章 Off-Site Synchronization from Unidirectional Transformation では off-site 同期を扱っている。あるモデルから他のモデルへの単一方向のモデル変換は数多く存在するが、変換後のモデルが更新されたときにそれをもとのモデルに反映させる手段が存在しない。ここでは、広く使われている変換記述言語 ATL を用いて、変換経過に基づいて ATL プログラムから同期プログラムを導出する方法を提示している。これによれば、ATL の記述だけで、他に付加的なものを用意しないでモデル間の同期が可能であると主張している。

第5章 Off-Site Synchronization from Bi-Transformation では、2つのモデル間に双方向の変換が存在することに着目し、それに基づく同期手法を提案している。双方向変換は一方のモデルの更新を他方に反映させることができるが、双方のモデルに併行して更新がなされたときには双方向変換だけは有効な手法とはならない。ここでは、off-site 同期に対して、併行して更新が行われたときにも対応できる双方向変換に基づく手法を提示している。

第6章 Beanbag: An On-Site Synchronization Language では、on-site 同期のための記述言語 Beanbag の設計を述べている。統一的なモデル記述言語として知られる OCL(Object Constraint Language)には同期に関する機能は考慮されていない。これに対して、Beanbag は OCL に同期の機能を導入し、OCL と同様にモデルを記述して、モデルに対する無矛盾性等の検査を行うとともに、モデルが更新されたときに同期を実行するシステムが構築されている。ここでは、このような Beanbag の提案はこれまでにはなかったモデル記述言語であるとその新規性を主張している。

第7章 Implementation and Application of Beanbag では、第6章で扱った Beanbag の処理系の実現を述べている。現実的な Beanbag の実現にあたって重要な課題は、更新・変更を取り消すことのできる仕組みであり、関数型言語 Haskell による遅延評価を有効に使う方法を提案し、それを実現している。また、Beanbag によって開発したアプリケーションで実用性を実証している。

第8章 Concluding Remarks では本論文の成果をまとめるとともに、今後の研究課題に触れている。

以上を要するに、本論文はソフトウェア工学におけるモデル同期に関して新たな方法論を理論的に展開するとともに、システムを実現してその手法の有効性を実証したもので、数理情報学、計算機科学の発展に寄与するところが大きい。よって本論文は博士(情報理工学)の学位請求論文として合格と認められる。