氏　　名（本　籍）	SONG KUK HWAN 宋　国　煥　（韓　国）
学位の種類	博　士（情報科学）
学位記番号	情博第17号
学位授与年月日	平成8年3月26日
学位授与の要件	学位規則第4条第1項該当
研究科，専攻	東北大学大学院情報科学研究科（博士課程）情報基礎科学専攻
学位論文題目	A Flexible Software Development Method and Its Support System （ソフトウェアのやわらかい開発法とその支援システム）
論文審査委員	（主　査） 東北大学教授　白鳥　則郎　　東北大学教授　根元　義章 東北大学教授　牧野　正三

論文内容要旨

Chapter 1. Introduction

As information processing systems become large and complex, formal description methods are needed for specification of systems, their efficient and reliable designs. In the field of communicating systems, the approach describing specifications formally has been researched before, so that Formal Description Techniques (FDTs), e. g. SDL, Estelle and LOTOS, have been proposed as specification languages. FDTs in general fare well in describing a target system unambiguously, precisely and completely. However, the drawbacks of these FDTs are that they are generally not user-friendly (easy to read) and difficult to write. They are also not suitable for rapid prototyping, because we must enumerate and/or determine all system behaviors from an early stage of system design. These drawback could be overcome by a new technique of automatic synthesis of formal specifications from user requirements. A state transition system (STS) is an underling structure of such formal description techniques, and often used as a formal specification itself. However, it is often necessary to modify or change system requirements which may influence the whole system design in the early stage. In the community of communication network, the new concept of a *flexible system* has taken much attentions as a key concept of advanced information network systems. In this thesis, we propose a solution to cope with the changes of system requirements, using the concept of a *flexible software development method* based on the flexible system.

Chapter 2. Flexible Software Development

"Flexbility"of the system is defined based on the concept of structured stability which represents homeostasis of functionality against variance of system structure and user's requirements. Thus, a flexible system is a system whose services and utilities are not restricted or confined to a fixed standard set. A flexible system has the capability to reconfiguration itself autonomously and extend its services spontaneously based on change of user's requirements, operational situation, and so on. The flexible system needs to have the characteristics – *intelligence*, *homeostasis*, and *evolution*. Here, we apply the concept of a flexible system to software development method in the field of software development. Therefore, we propose some required properties of a flexible method must satisfy, and we define a flexible software development method (FSDM) along those properties.

Based on the concept of FSDM described in Chapter 2, we present our methods in detail from Chapter 3 through 5.

Chapter 3. Synthesis of Formal Specification

We propose a new requirement description method (RDP) for describing system requirements based on propositional logic, which copes with the modifications or changes easily in the system requirements. Further more, we show the correctness logically that a sound and complete state transition system (standard model) which is a kind of formal specifications can be obtained from RDP. We also show that function requirements can be modeled by a Logical Petri Net (LPN), which is a kind of extended Petri Nets, in order to derive a formal specification automatically.

Chapter 4. Verification of Requirement Description

There may exist some logical errors in a system requrement. In this chapter, we state to find out the logical errors from the requirement description using a LPN, and we discuss a reflection method to detect logical errors in the requirement phase. In properties and definition of an FSDM, we can say that a flexible development method is more flexible if a reflection of the detected logical errors can be performed in requirement phase. In this thesis, we consider the derivation of system specifications based on system requirements, and the causes of logical errors are occurred by the conflict in the relations among primitive function requirements. Therefore, we deal with logical errors in the view point of requirement level, even though some errors are represented in the level of system specifications.

Chapter 5. Refinement of System Specification

System requirements may often be changed, and specified step by step. Therefore, *flexibility*, *refinement* and *abstraction* are the central notions for rapid and reliable system／software development. In this chapter, we propose a method to refine and decompose requirement descriptions hierarchically. Further we discuss an integrated system satisfying the intended system requirements in the refined STSs. We also show that a system can be changed flexibly against changes of system requirements.

Chapter 6. Design Support System

In this chapter, we discuss a design support system which was implemented for supporting our methodology. Here, we show the configuration of our support system in Figure 1, and the procedure for synthesizing state transition systems (STSs) from acquisition of system requirement in the support system. After that, as a real application example, a CATV system is applied in the support system. The modules for the support system are constructed with 5 parts such as user interface, translator, synthesizer, verifier and reflector.

(1) User interface : This module plays a role to acquire system requirements easily, and also supplies an environment in which users can specify refined RDPs under the supported environment by only giving the information of constraints.

(2) Translator : In this module, RDPs are translated into a canonical form of requirement descriptions by applying 2 transformation rules. After that, the canonical form is also translated into an LPN form by this module.

(3) Synthesizer : In this module, an STS as a formal specification is synthesized from an LPN form od system requirements by firing LPN along the execution algorithm of LPN. The synthesized state transition systems are represented as graphics.

(4) Verifier : In this module, logical errors are examined by checking LPN, reachability tree derived from LPN, and reverse reachability tree generated from reachability tree. This module finds out 4 kind of errors.

(5) Reflector : This module shows users the STS pointed out the detected logical errors, and also gives a guideline which is useful when users rewrite requirement descriptions (RDPs) to remove the detected logical errors.

Chapter 7 . Conclusions

In a state transition system, it is generally difficult to decide states in a complex system in the view point of considering the meaning of states, and it is often necessary to modify or change system requirements which may influence the whole system design in the early stage of system design. The specifications until now are almost specified using states and state transitions directly like SDL when describing system requirements. However, the methods are not suitable for large and complex systems having a large number of states, and for rapid prototyping system design.

The purpose of this thesis is to present a new paradigm of a flexible software development method (FSDM) to overcome the difficulties mentioned above. In this dissertation, we proposed a new method to describe system requirements based on propositional logic, and to synthesize state transition systems automatically. we also proposed a verification method and a refinement method for system requirements, based on the concept of an FSDM. We also showed that a CATV system as a real application example of communication software could be applied well in our support system.

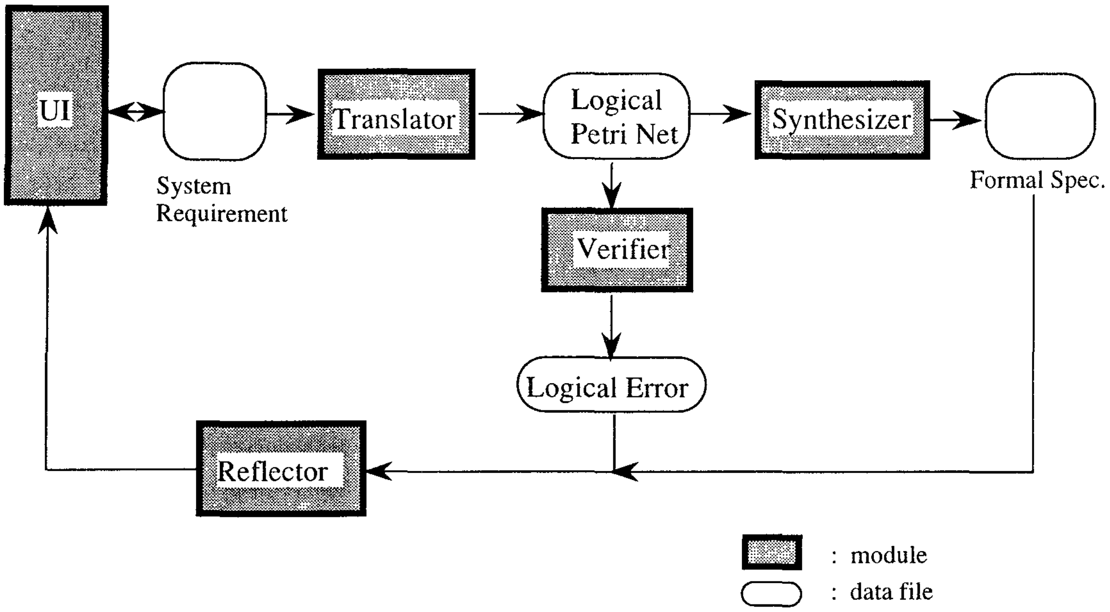

Figure 1 : Configuration of support system

審査結果の要旨

情報通信システムの開発において，初期段階ではシステム仕様が頻繁に修正され，その結果がシステム開発の全工程に大きな影響を及ぼしている。そのため，ユーザ要求の変更に柔軟に対応できるシステム開発法の確立が課題となっている。しかし，現状では，このような課題を解決するための有効な方法論や支援環境は十分には確立されていない。そこで，著者は，解決策としてソフトウェアのやわらかい開発法を提案し，これに基づく設計支援システムの構築に関する研究を行った。本論文はその成果をまとめたものであり，全編7章よりなる。

第1章は序論である。

第2章では，ユーザ要求の変更に柔軟な対応できるソフトウェアのやわらかい開発法を提案している。この開発法に基づいて，次章以降において具体的なシステムの開発について議論している。

第3章では，命題論理に基づいた情報処理システムの要求記述法を提案し，この記述法による表現から，その意味を与える状態遷移システムへの変換法を構成し，その妥当性示している。また，命題論理で与えられた要求を効率的に分析し研修するためにペトリネットを拡張した論理ペトリネットを提案している。次に，この論理ペトリネットを用いて命題論理で与えられたユーザ要求に対応する状態遷移システムの合成法を示している。これは，ソフトウェア開発の基礎となる重要な成果である。

第4章は，ユーザ要求に包含されている論理的誤りについて，これらを検出する効果的な方法を与えている。さらに，検出された誤りに対して，ユーザ要求の記述に反映する方法として，ユーザに誤りを訂正するためのガイドラインを提示することにより，ユーザ要求を効率的に修正し記述できることを確認している。

第5章では，第3章における命題論理により表現された要求を詳細化する記述法を提案している。本詳細化記述法により，大規模で複雑なユーザの要求を段階的に記述することができる。また，断層的分割の概念の導入により，状態遷移システムの分析と理解が容易となっている。これは，実用上興味深い結果である。

第6章では，第2章より第5章で議論したソフトウェアのやわらかい開発法の支援システムの構成について述べている。この支援システムをワークステーション上に構築し，具体的な応用としてCATVなどの通信システムの開発へ適用し，その有効性を確認している。

第7章は結論である。

以上要するに本論文は，情報通信システムの設計における基本問題である仕様記述とその支援システムの構成に関する研究を行い，高度な情報通信システムのための新しい開発法に関する有用な知見を与えたもので，情報基礎科学の発展に寄与するところが少なくない。

よって，本論文は博士（情報科学）の学位論文として合格と認める。