

氏 名 (本 籍)	PAIROJ TERMSINSUWAN (タイ)
学 位 の 種 類	博 士 (情報科学)
学 位 記 番 号	情 博 第 19 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	Support Methods for Formal Specification Description based on Reuse and Their Environments (再利用に基づく形式仕様記述支援法とその支援環境)
論 文 審 査 委 員	(主 査) 東 北 大 学 教 授 白 鳥 則 郎 東 北 大 学 教 授 西 関 隆 夫 東 北 大 学 教 授 阿 曾 弘 具 (工 学 研 究 科)

## 論 文 内 容 要 旨

### Chapter 1

Due to the increase in size, complexity and variety of the software and communication system nowadays, Formal Description Techniques (FDTs) which can describe the system in a precise, unambiguous and especially abstract (implementation independent) way are becoming more necessary and important in order to design and develop these systems efficiently. Abstract Data Type (ADT) is a powerful tool having a very good mathematical basis called many-sorted algebra and is also combined into several FDTs. By using ADT, a protocol or a software system can be specified in a very abstract way. It also facilitates the verification and validation of the specified system. However, although ADT has been successfully applied in research, practically ADT is rarely used because it is difficult for a non ADT-expert to specify. As a consequence of this, in order to promote a wide spread use of ADT and also FDT, support method for ADT specification is extremely necessary. Unfortunately, there is almost no such a support method so far.

We propose new approach to ADT specification support by adopting Case-Based Reasoning (CBR) which is a kind of problem solving by (i) searching (reusing) solution of the existing similar problem and (ii) modifying solution of the similar problem to the current problem. CBR has been used in several research projects but there is no research, which uses this technique for ADT specification support.

### Chapter 2

We describe the brief introduction and definition of ADT. First we introduce a simple ADT without other existing ADTs imported into the definition (a triple of a set of *sorts* (basic data elements), a set of *operations* further divided into constructors and functions, and a set of *equations*). Then the definition of ADT in general, where some existing ADTs may be used in the definitions, is given.

### Chapter 3

In this chapter, we propose an **ADT Specification Support system (ASS)** as a searching support part by asking the requirement of a new ADT, and then providing a similar ADT (ADT with similar requirement to the specifier's one) as shown in Fig. 1. ASS can ease the specification and prevent user from spending too much time and effort in specifying the whole ADT from the beginning. It mainly consists of four parts 1) Case base of ADTs, 2) Requirement Acquisition Template (Interface) 3) Search mechanism for finding the most similar case in 1) and 4) Similarity definition for similarity calculation. Among these parts, the core part is the search mechanism which is designed based on the similarity definition. To design this part, we propose (1) an **ADT specification Model (ADTM)**, which classifies four patterns of constructor and describes the constructor part based on combinations of these patterns, and (2) a **similarity definition**, which calculates how much two ADTs look similar. This definition guides specifiers in deciding if one ADT can be reused in order to specify another new ADT or not.

Then we design and implement ASS based on these concepts. By ASS, specifier can easily specify a new ADT by modifying a similar ADT retrieved from the case base. Therefore, specifier's load is quite reduced.

### Chapter 4

In this chapter, we propose an **ADT equation Modification Algorithm (AMA)** as the main part of modification support. AMA automatically modifies the equation part of ADT whenever specifier inserts some equations such that there is no error (incompleteness and inconsistency). Furthermore, it also guarantees that the modified equation part satisfies the inserted equations. The basic idea of AMA is (1) to search for equations in equation part which define function over the domain of the inserted equation. These equations are called the **overlapping equations** of the inserted equation. Based on this concept, the algorithm (2) deletes the overlapping equations, (3) incorporates the inserted equation into the equation part and (4) recreates equations with nonoverlapping domain. Finally, proof for the algorithm is given.

AMA enables a specifier to easily specify an ADT by only inserting the new semantics (equation). As a result, it can both solve the main problem about difficulties in modification and widen the scope of reusable ADTs. This in turn, improves the productivity and quality of ADT and FDT.

### Chapter 5

To solve the remaining problems in ASS and AMA, we first propose other three automatic modification algorithms to deal with other types of modification which are Algorithms to (1) **Delete a Function (ADF)**, (2) **Delete a Constructor (ADC)** and (3) **Add a Constructor (AAC)**. These algorithms including AMA are implemented as **ADT Modification Editor (AME)**.

Then, in order to provide a more effective support method, we propose an **ADT Specification Support Environment (ASSE)**, an extension of ASS by AME as shown in Fig. 1. From ASS, AME will get a similar ADT and acquire additional information for modification from specifier. As a result, the similar ADT will be modified automatically and returned to the specifier. We can see that ASSE facilitates the specification, thus greatly reducing the specifier's load, since he or she can avoid both specifying a required ADT from the beginning and modifying a similar ADT manually.

**Chapter 6** gives concluding remarks and suggestions for future works.

As a result, by using our systems, a specifier can easily specify a required ADT by inputting modification information for each similar ADT retrieved from the case base. Therefore, both of specifier's workloads in (1) specifying the whole required ADT from the beginning and (2) modifying the retrieved similar ADT are signifi-

cantly reduced. Furthermore, in order to evaluate the effectiveness of these systems, we experimented with data types of services and protocols in data communication field (in Chapter 3, 5). The evaluation confirmed the effectiveness of the systems.

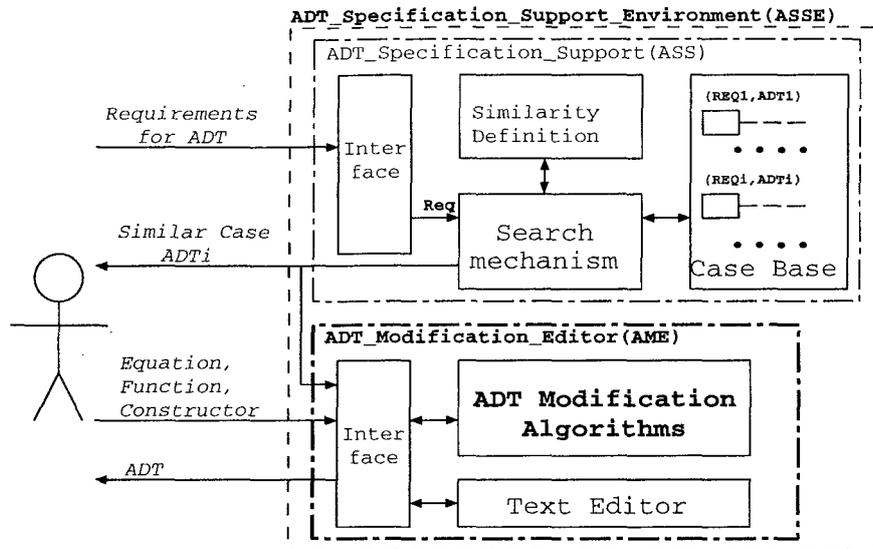


Figure 1 : Outline of ASSE

## 審査結果の要旨

情報通信システムを構築する上で、システムの正確な仕様を効率よく記述するための形式仕様の記述支援問題が、極めて重要な課題となっている。しかし、現状ではこのような課題を解決するための形式仕様の記述支援法とその支援環境は十分には確立されていない。そこで著者は、類似仕様の再利用に基づく仕様記述支援法を提案し、その支援環境の構成に関する詳細な研究を行なった。本論文はその成果をまとめたものであり、全編6章よりなる。

第1章は序論である。

第2章では、本論文で研究の対象とする抽象データ型 (ADT) の形式仕様記述法について述べている。

第3章では、ADTの記述を効果的に支援し、仕様記述者の作業の負担を軽減させる事例ベース推論に基づくADTの類似仕様検索支援システムASSを提案している。ASSは、仕様記述者から新しく仕様化するADTの要求を獲得し、その要求に対応する類似仕様を提示するシステムである。また、このシステムに基づく支援環境をワークステーション上に実現し、実際に情報通信システムを対象として記述実験を行ない、ASSの有効性を確認している。これは情報通信システムの形式仕様を効率的に記述する上で、基礎となる重要な成果である。

第4章では、ASSにより提示された類似仕様を修正し、所望のADTを得る修正問題における支援として、ユーザからの類似仕様における等式に関する修正要求を自動的に処理する修正アルゴリズムAMAを構成している。AMAで修正された仕様には論理的な誤りがなく、また仕様記述者の修正要求を満足することが保証されている。この結果は、類似仕様の系統的な修正法を与えるものであり、仕様の再利用効果的な方法として興味深い結果である。

第5章では、第4章で構成したAMAを拡張し、類似仕様における演算子に関する修正アルゴリズムを提案している。次に、提案したアルゴリズムをワークステーション上に実装し、これらと第3章で構成した検索システムASSを統合したADTのための総合支援環境ASSEを構築している。ASSEでは、ADTの記述において再利用可能な類似仕様の検索から修正まで、仕様記述の全体を支援する総合的な機能を提供している。また、実際にASSEを用いて情報通信システムにおけるサービスやプロトコルを対象として記述実験を行ない、その有効性を確認している。

第6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形式仕様の記述支援を目的として、ADTの類似仕様の再利用に基づく支援法とその支援環境の構成に関する研究を行ない、高度な情報通信システムの構成のための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床が少なくない。

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。