

# 状態遷移表を用いた分散オブジェクト環境向け設計支援環境の構築

5U-9

柴山 武彦\*, 河村 謙治\*, 田中 吉廣\*\*

\*日立中部ソフトウェア(株)、\*\*(株)日立製作所 ソフトウェア開発本部

## 1. はじめに

通信系ソフトウェアを中心に、状態遷移表を用いた開発環境が利用されてきた。状態遷移表は、プログラミング言語に比べ、対象の動きを状態と事象により整理出来る点で優れており、特に表形式で記述することから漏れが少ない。

一方、近年、分散オブジェクト環境が注目を浴びている。分散オブジェクト環境では、ネットワークの存在を意識せず分散環境上でのオブジェクトにアクセスが可能といった利点があり、今後も急速に普及すると考えられる。しかし、分散オブジェクト環境向けの開発支援環境という点で、まだまだ多くの開発環境、ツールが普及するにいたっていない。

これに対し、我々は、従来の状態遷移表を用いた設計支援環境を応用し、分散オブジェクト上でのソフトウェア、特にサーバ側ソフトウェアの開発を支援する環境を構築した。状態遷移表を用いた従来の支援環境を利用したのは、分散オブジェクト設計で状態遷移設計が重視される、従来環境の利点があるまま利用できる等の理由による。本報告では、以下、順に、本支援環境について述べる。

## 2. 従来の設計支援環境

我々は、状態遷移表のアイデアを利用した設計支援環境を開発し、実際のソフトウェア開発で利用してきた[ref.1]。従来の設計支援環境の概要を図1に

しめす。この設計支援環境では、仕様設計に近い上流工程からコーディング、テストといった下流工程まで、開発全般を支援している。支援環境を構成するツール群として、1)状態遷移表を編集するエディタ、2)状態遷移表を検証する検証機能、3)状態遷移表レビュー等のため、その動きを見やすく図示したシーケンス図を生成するシーケンス図生成機能、4)プログラミング言語ソースを生成するソース生成機能からなる。これらツール群は状態遷移表記述を含むテキストファイルである状態遷移表ファイルを共通の入力(編集)対象とする。

この支援環境では以下の狙いがある。

- 1)図1に示したとおり状態遷移表ファイルを中心に、全工程を一貫支援する。
- 2)設計工程、プログラム工程、テスト工程それぞれを支援するツール群を提供している。
- 3)ビジュアル化による品質向上、工数低減できる。すなわち、状態遷移表形式に記述したり、シーケンス図を生成したりすることで、母体理解が容易になる。また、変更箇所の特定が容易となる。本支援環境を実際のソフトウェア開発に適用し、品質向上、生産性向上の効果を上げてきている。

## 3. 分散オブジェクト環境向け設計支援

本設計支援環境では、設計部署でサーバ側ソフトウェアの開発比重が高かったため、サーバ側開発支援に重点を置いた。さらに、開発の中でも大きな割

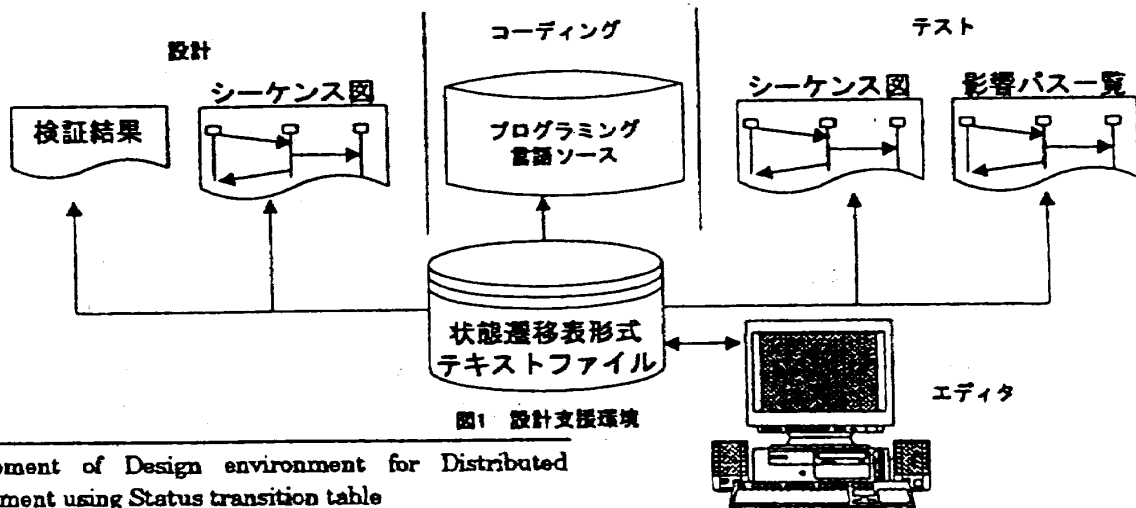


図1 設計支援環境

Development of Design environment for Distributed environment using Status transition table

Takehiko Shibayama, Kenji Kawamura, Yoshihiro Tanaka  
Hitachi-TS, Software Development Center, Hitachi, Ltd.  
5030 Totsuka-cho, Totsuka-ku, Yokohama, 244-0003, Japan

合を占めるテスト工程に対し、早急に支援するため、クライアントシミュレータを本支援環境とは別に開発、提供した[ref.2]。引き続き、上流工程、コーディング工程を対象に、既存の支援環境を利用した分散オブジェクト環境の設計支援を開始した。

従来の環境を構成するツール群では、各工程に対する支援を行っているが、分散オブジェクト環境への切り口がない。しかし、状態遷移表に記述した事象は、記述対象のインタフェースと考えることができる。従って、既存ツールの入力対象である状態遷移表ファイルを入力し、その中にある事象記述を抽出し、IDLソースを生成することで分散オブジェクト環境への切り口を開くことができる。表2に状態遷移表記述と、自動生成するIDLファイル、およびインプリメントクラスファイルの対応を示す。

この方法では、既存のエディタ、検証機能、シーケンス図生成機能をもそのまま利用でき、それら機能を利用する利点も失わない。すなわち、図1の環境をそのまま利用できる。

表2 状態遷移表記述と生成ファイルの対応

状態遷移表記述	IDLファイル	インプリメントクラスファイル
全体	interface	class
事象	operation	method
各事象対応記述 <sup>*)</sup>	- (生成せず)	(メソッドの先頭で各状態別に振り分ける)

\*1)事象を縦に並べて記述した時の横1行

#### 4. 新設計環境の概要

図2に上記狙いを実現するためのソース生成機能と関連するファイル構成を示す。従来の設計環境

と新ソース生成機能を利用することにより、以下の手順でサーバ側ソフトウェアを開発できる。

- 1) サーバ側オブジェクトに対して、状態遷移設計を行う。さらにエディタを利用して状態遷移表形式で記述し、漏れをチェックする。
- 2) 状態遷移表の検証機能により検証する。クライアント側の状態遷移表も含めて、デッドロック等のエラーを検出する。必要に応じてシーケンス図を生成し、動作を中心にレビューする。
- 3) ソース生成機能により、サーバ側ソフトウェアを生成する。
- 4) クライアントシミュレータによりテストする。不良修正時にはシーケンス図を生成し、レビュー、必要なテストケースを洗い出し、修正する。

#### 5. まとめと今後の課題

本環境では、分散オブジェクト環境のサーバ側ソフトウェアに対し、上流工程からコーディング工程までを支援する開発支援環境を構築できた。今後、以下の課題がある。

- 本環境による品質向上、工数削減割合の調査とフィードバック
- テスト工程支援充実のためのクライアントシミュレータ連携
- 実用レベルのソフトウェア開発支援

#### 参考文献

- [1] 柴山、他：状態遷移表検証ツールの適用；情報処理学会第52回全国大会論文集(1996)
- [2] 河村、他：分散オブジェクト環境向けクライアントシミュレータ；情報処理学会第57回全国大会論文集(1998)

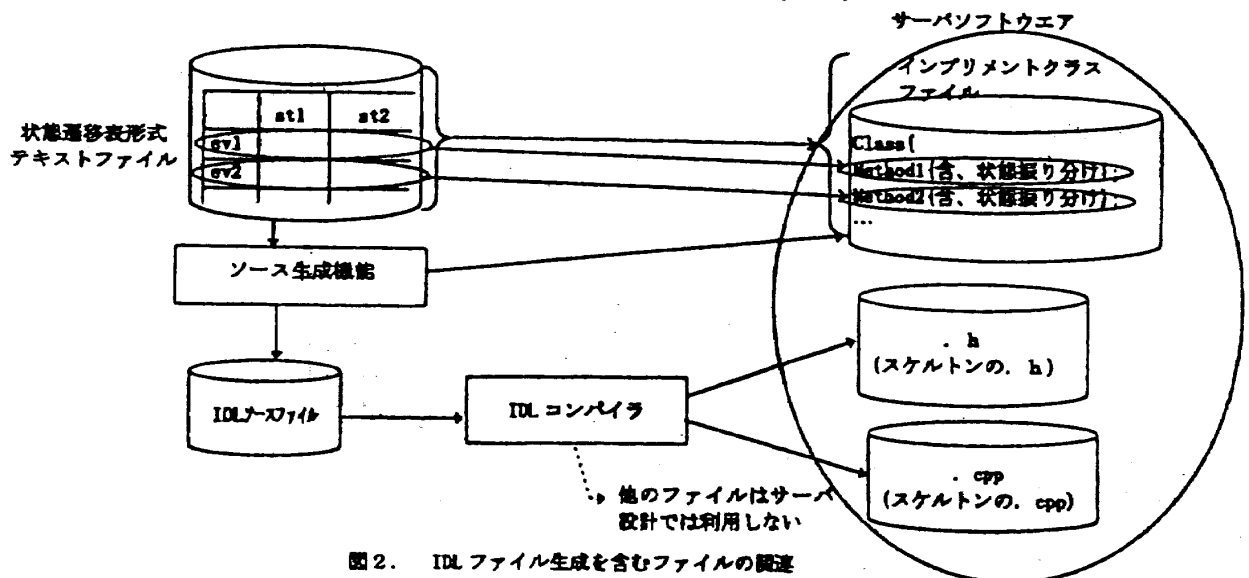


図2. IDLファイル生成を含むファイルの関連