

## ソフトウェア・プロセスによる 設計教育環境の開発及び評価

望月 純夫                      山内 顕  
三菱スペース・ソフトウェア(株)

過去の設計技術者の育成は、OJTに頼ってきたが、あまり効果的ではなかった。新しい設計教育環境の開発には、先ず設計技術を顕在化させ、それを中心とした教育環境を設定する必要がある。我々は、プロセス・モデルHFSPに基づき設計技術を分析、記述した。その結果、熟練技術者は長年の設計経験から普遍的モデルという形の設計技術を蓄え、それを設計対象システムの特徴に合わせて調整し、実行可能モデルに変換して実際の設計作業に適用していることが明かとなった。この様な研究成果を中心として設計教育コースウェアを開発し、技術者教育を試みてその効果を確認した。この設計教育コースウェアは、将来的には、設計コンサルティング・システムに発展させる積もりである。

### DEVELOPING AND EVALUATING OF DESIGN INSTRUCTION COURSEWARE BASED ON SOFTWARE PROCESS

Sumio Mochizuki              Akira Yamauti  
Mitsubishi Space Software Co.  
Nagashima Bldg., 524, Kamimachiya, Kamakura-shi, 247, Japan

In the past, we have been training the design engineers, relying on OJT largely, but it was not so effective. In order to develop a new training courseware, it is necessary to clarify and describe the design skills, and the training courseware should be developed on the basis of them. We analysed and described design skills, using the process model HFSP. As the results, it was clarified that the experienced engineers gather the design skills in the form of Generic Model. Designing a system, they adjust it into Executable Model by the characteristics of the objective system, and apply it to the real design works.

Using these results of our research, we developed a training courseware, and applied it to the instruction for the young and unexperienced engineers, and confirmed its effects. We are expanding it to a design consulting system in the future.

## 1 はじめに

最近のソフトウェア産業における技術者不足の傾向は著しく、今後とも一層激しいものになると予想されている。このような状況の基に、各社とも技術力、人材を確保すべくたゆみ無い努力を続けている。しかし、ソフトウェア開発技術、特にその設計技術を急速に向上させることは困難であり、また熟練技術者の設計技術を伝承させることは難しい事とされてきた。従来、設計技術を伝承させる手段としては、OJT(On the job training)がよく使われてきたが、一回のOJTによって学習できる範囲は限られるためシステム全体の設計を教育するまでには長い期間が必要となり、その教育効率は極めて悪いといわざるを得ない。また、いつも適切な教材(開発業務)及び教育指導者が用意出来るとは限らないという問題もある。ここに新しい設計教育環境の設定が必要となるのである。熟練技術者の設計技術を顕在化するための技術として、最近ソフトウェア・プロセス[1]の研究が盛んである。我々は、熟練技術者がシステムを設計する時の思考過程を、ソフトウェア・プロセス・モデルH F S Pに基づいて分析した。その結果、その部門における普遍的ともいえる標準的な設計プロセスを得た。さらに、熟練技術者がこのプロセスをどの様に調整して、他のシステムの設計に再利用しているかを解析した。

これらの設計作業分析の研究成果を基にして設計教育コースウェアを試作し、実際に教育を試行して設計技術教育に効果的であることを確認した。

## 2 ソフトウェア・プロセス・モデルH F S P

### (2)の概要

ソフトウェア・プロセス・モデルH F S Pでは、ソフトウェアの開発作業を関数の集合として捕らえ、その入力オブジェクト及び出力オブジェクトを合わせて明確化しようとするものである。開発を細かい作業の連続として記述し、それらを厳密な関数の形式に表現して、入力オブジェクトから出力オブジェクトへ変換するものとする。

即ち、次の手順に従ってソフトウェア・プロセ

スが定義される。

- ①ソフトウェア開発作業を関数として表現し、その各々の入力オブジェクトおよび出力オブジェクトを明確に定義する。(第1レベルのプロセス)
- ②プロセスの機能が複雑である場合は、そのプロセスを更に細かいサブ・プロセス(第2、第3レベル等のプロセス)に分解する。この作業を各々のプロセスが十分に単純化されるまで続ける。
- ③プロセスの分解に際しては、オブジェクトがプロセスの途中で消滅したり、急に現れたりしないように各々のオブジェクト間の関係を明らかにする。即ち、各サブ・プロセスが、どの様な入力オブジェクトを受け取り、その出力オブジェクトからプロセスの出力オブジェクトがどの様に構成されるかを明確にする。

本論文は、設計現場における標準的な技術を記述すると共に、熟練技術者が、それを調整して他のシステムの設計業務に適合させる、いわゆる設計プロセス再利用の仕組みについても述べる。また、これらの成果を基に設計教育コースウェアを試作し、これを評価するものである。

## 3 設計現場における熟練技術者の設計作業

熟練技術者は、自己の豊富な設計経験の中から、様々なシステムの設計に広く適用できる普遍的な設計手順を抽出し、蓄積している。具体的なシステムを設計する場合には、このような普遍性を持つ設計手順をその設計対象システムに合うように調整して、再利用している。我々は、いくつかの具体的なシステム設計作業の分析を通して、このような思考過程の仕組みを解析した。熟練技術者の保有している普遍的な設計方法を普遍的モデル(Generic Model)と呼ぶこととする。この普遍的モデルは、非常に簡単なシステムを設計する場合には殆どそのまま利用できるが、システムの規模がある程度大きくなるとこれに調整を加えてそのシステムに適した実行可能モデル(Executable

Model)に変換する必要がある。

### 3.1 普遍的モデルの抽出及びオブジェクト中心型記述方式による記述

我々は、これまでいくつかの実時間処理システム的设计手順を分析し、プロセス・モデルHFS Pに基づいてその设计手順(プロセス)及び成果物(オブジェクト)を明確化して記述した。[4][6][8][9][11][13][14]分析の対象としては、设计作業の中で最も重要と思われる基本设计フェーズを取り上げ、全てのオブジェクトをその内容によっていくつかのグループに分類し、それらがどの様に作られるかに重点を置いて设计プロセスをまとめた。我々は、この記述方式をオブジェクト中心型記述方式と呼ぶ。その結果得られた设计プロセスを図1(第1レベル)及び図2(第2レベル)に示す。

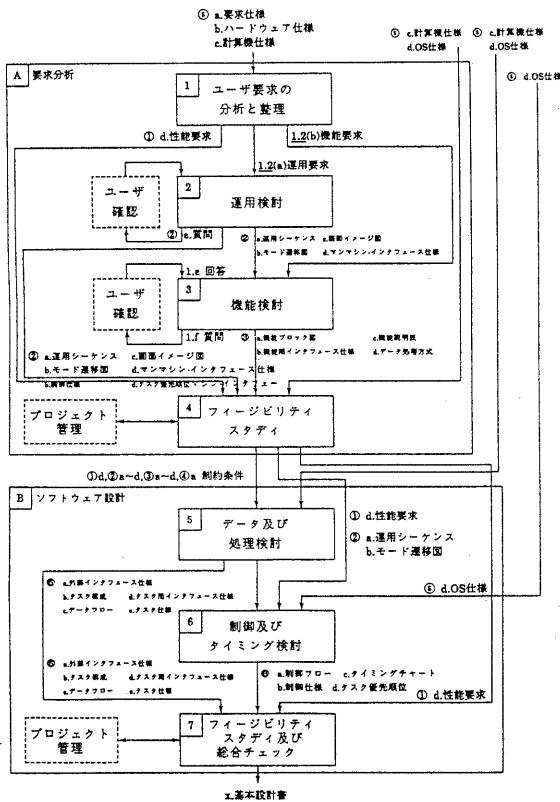


図1 実時間処理システム基本設計のソフトウェア・プロセス(第1レベル)

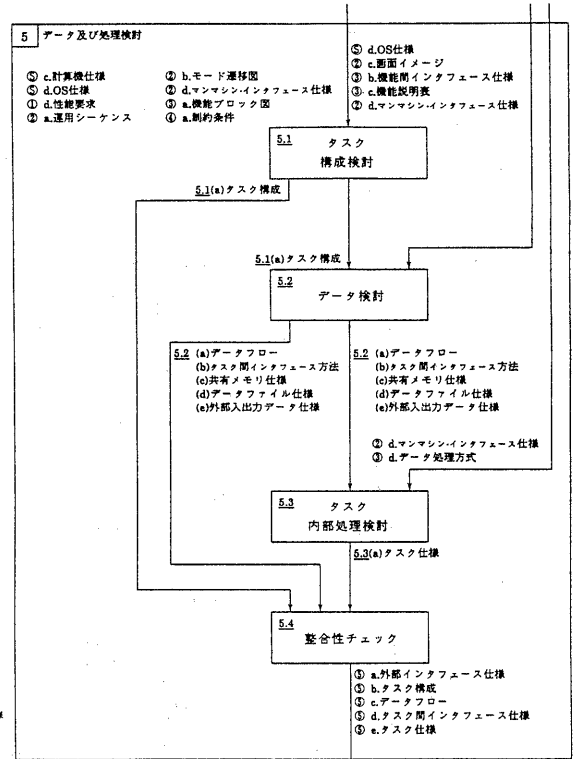


図2 実時間処理システム基本設計のソフトウェア・プロセス(第2レベル) - データ及び処理検

図中、ソフトウェア・プロセスをブロック図で示し、ブロック間にその間で授受されるオブジェクトを示す。この記述方式は、設計内容を中心にして各プロセスをまとめ、その各々の入出力オブジェクトをも明確にしたものであり、時間的なファクターは除き、あくまでも各プロセスとその入出力オブジェクトとの関係を示した、極めて静的なプロセスである。これらの内容(設計プロセス、その入出力オブジェクト及びそれらの関係)は、多くの計測制御システムの設計に共通のものであり、総合して普遍的モデルと呼ぶこととする。

実際の設計作業の中で各プロセスが実行される状況を図5、図6に示す。設計技術者は、各プロセスを比較的長期にわたり実行し、また一時点に複数のプロセスを併行して行う必要がある。

### 3.2 実行可能モデルの抽出及び手続き中心型記述方式による記述

実際の実用システムの設計手順を分析するため、我々がこれまでに豊富な設計経験を持つ、人工衛星の機能及び性能のチェックアウトを目的としたデータ収集解析システム(Data Acquisition System---以下DASと略称する。)を選択し、その具体的な設計手順を時間の経緯に忠実に分析し、表現した。(図3(第1レベル)及び図4(第2レベル)参照。)このように実際のシステム設計を時間の経緯と共にありのままに記述した設計プロセスを手続き中心型プロセスと呼ぶ。

図3の中で、プロセス1、2、5、6は、ソフトウェア基本設計の主体となるものであり、基本設計を4つの時間帯(フェイズ)に分けてその時間帯の中における設計プロセスとオブジェクトを記述したものである。図4は、その中の第2プロセスをさらに詳細化した第2レベルのプロセスである。この中のプロセスの内、例えば、2.1運用検討、2.2機能検討などは、図3(第1レベルのプロセス)の第5プロセス、第6プロセスにも含まれている。この他にも、いくつかのプロセスが異なった時間帯に繰り返して出てくるため、この手続き中心型記述方式によるプロセスは、理解しにくくなっている。

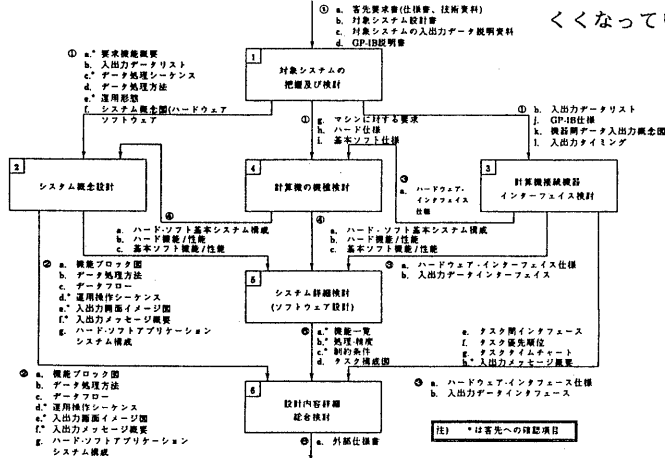


図3 DAS基本設計のソフトウェア・プロセス(第一レベル)

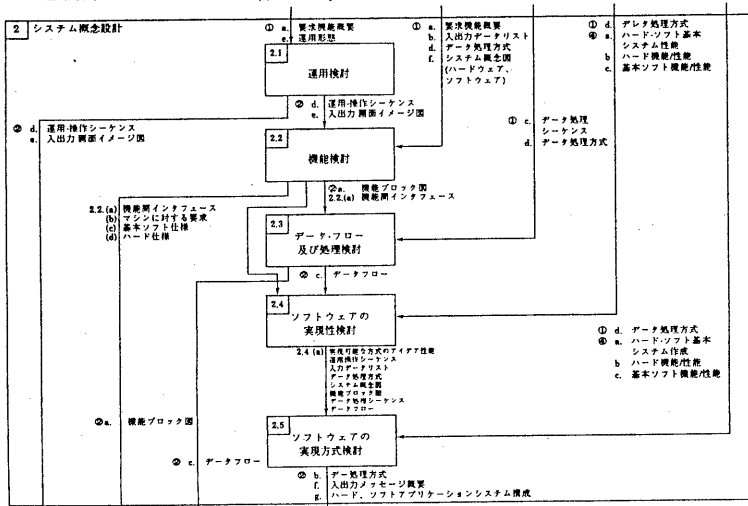


図4 DAS基本設計のソフトウェア・プロセス(第二レベル)-システム概念設計プロセス

### 3.3 普遍的モデル（オブジェクト中心型プロセス）と実行可能モデル（手続き中心型プロセス）の関係

普遍的モデル（オブジェクト中心型プロセス）と、大規模システム（DAS）及び小規模システム設計のための実行可能モデル（手続き中心型プロセス）との関係を図5及び図6に示す。

図中、横軸（時間軸）にそって手続き中心型プロセスが並んでおり、縦軸には、オブジェクト中心型プロセス（設計作業項目）が並んでいる。設計の進捗とともにいくつかの作業が併行して実行されるが、設計対象システムの規模が大きくなるに従ってプロセスの併行動作が著しくなる事が解る。このような設計プロセスの併行動作が必要となる原因には、次のようなものがある。

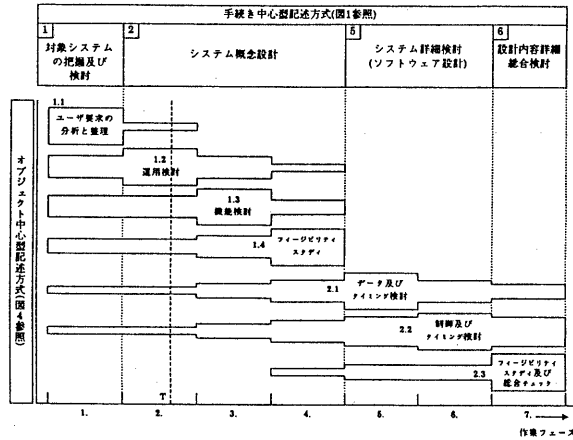


図5 手続き中心型記述方式及びオブジェクト中心型記述方式ソフトウェアプロセス（第1レベル）の比較（DAS設計プロセス）

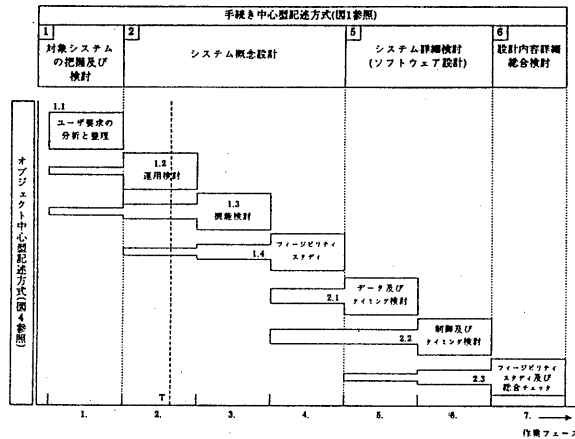


図6 手続き中心型記述方式及びオブジェクト中心型記述方式ソフトウェアプロセス（第1レベル）の比較（小規模システム設計プロセス）

①関係の深い複数のプロセス（例えば、運用検討と機能検討など）は、互いに並行して処理を進めながらその途中で互いにオブジェクトを交換し、調整するように併行的に作業を行う。（これによって大幅な手戻り作業を避ける。）

②実用システムの設計には、いくつかの制約条件（例えば、目標機能、性能の実現など。）が課せられる。この実現のためには、B. W. Boehmの提唱するSpiral Model [3]のように、設計作業を進めながら繰り返しフィジビリティ・スタディを行う必要がある。そのためには工程の最後の方に位置するソフトウェア設計（タスク構造、制御、データ及び処理、制御及びタイミングなど）等の検討を前もって行う必要がある。（これを後工程のプロセスの先読みと呼ぶ。）図5、図6は、初期設計の段階から基本設計の工程全体を通して、この様なフィジビリティ・スタディを繰り返し行う為、設計当初より併行作業が多く発生することを表しているのである。

大規模な実用システムでは、この併行動作が特に著しく発生するが、小規模かつ単純なシステムの設計においては少なくなり、時間の経過と共に、ほぼこのオブジェクト中心型プロセスをそのまま利用することができる。このように、種々のシステムに共通な普遍的モデルは、小規模システムの設計には、殆どそのまま利用できるが、設計対象システムの規模が大きくなると設計項目の実行の併行動作を工夫しなければならない。この様な調整作業の結果得られるプロセスが、実行可能モデルである。

#### 4 設計教育コースウェアの設定及び設計教育の実施

前章までの結果を基にして、若手技術者を対象とした設計教育コースウェア（図7）を設定した。

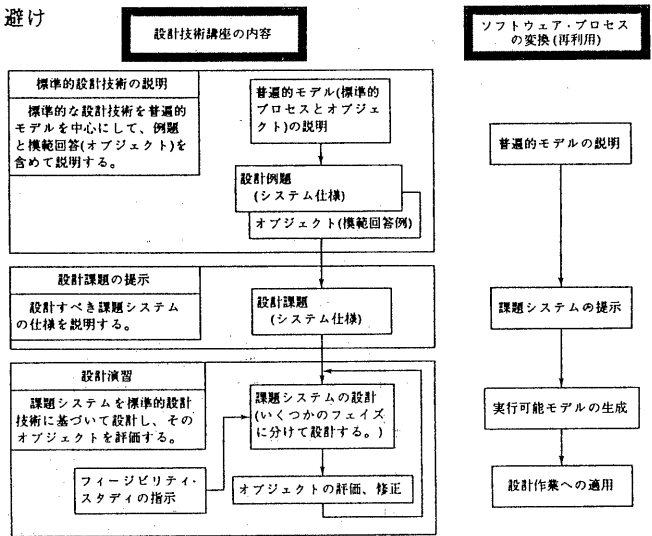


図7 設計教育講座の内容とソフトウェア・プロセス設計技術コースウェアの内容は、大きく3つのフェーズに分離されている。

##### (1) 設計標準技術の説明

普遍的モデルを基にして熟練技術者の標準的な設計技術を説明する。即ち、オブジェクト中心型プロセスで表現された標準的な設計プロセスとその入出力オブジェクトにより、設計の手順を説明する。このとき、設計の例題（小規模な計測制御システムの仕様）と、それを設計した結果得られる模範的な回答（オブジェクト一式）を基にして、システム設計の進め方をより具体的に細かく解説する。

設計そのものを理解させるためには、受講者はあらかじめ次のような技術要素を習得している必要がある。

- ①OSに関する基礎知識
- ②実時間プログラムの基礎知識
- ③計算機及びハードウェア・インターフェースに関する基礎知識

さらに、このような普遍的モデルを具体的な設計業務に利用する場合のプロセスの再利用についても説明する。

#### (2) 設計課題の提示

小規模な計測制御システムの仕様を設定し、それを実現するための計算機、計測制御対象、インターフェイス機器などの設計に関する条件を説明する。設計問題は、設計作業が40-50時間で完了する程度の内容とした。

#### (3) 設計演習

すでに説明した普遍的モデルに基づいて課題システムを設計させて、得られたオブジェクトの評価及び訂正を行う。これを繰り返して設計演習を完了させる。

また、演習の中でフィージビリティ・スタディのタイミングを指示し、次第にその精度を高めさせる。その結果として併行動作が必要であることを認識させ、その結果設計課題設計の実行可能モデルが得られることを体験させる。

この演習では、標準技術としての普遍的モデルを理解させるだけでなく、それをどの様に調整して課題システムの実行可能モデルを生成するかを体験させることも目的の一つである。即ち、標準技術としてのソフトウェア・プロセスである普遍的モデルを再利用する体験を与えることに重点を置いている。

#### 5 教育実習の評価 [5] [7] [10] [12]

これまで2回の教育実習を行った結果次のような評価を得た。

- (1) 標準技術及びその再利用を理解し、体験することが出来る。
- (2) 教育を目的としたソフトウェア・プロセスの細かさとしては、前述の二つのレベルのプロセスで充分である。(余り細かくすると個人の独創性を阻害するおそれがある。)
- (3) 標準的な設計技術ばかりでなく、標準的なオブジェクトのまとめ方を習得することが出来る。

(4) ソフトウェア・プロセスとその入出力オブジェクトだけでは、理解しにくいところがある。

これに対しては、次のような方法をとろうとしている。

- ① ソフトウェアの仕様をまとめた後、ソフトウェア設計を行う部分が特に初心者にとって理解しにくい。この設計に対しては、特に準最適な設計手順をまとめる。
- ② ソフトウェア・プロセス及びオブジェクトに対し、さらに詳しい情報を与える定義シートを作成する。

#### 6 まとめ

我々は、プロセス・モデルHFSFに基づいて、熟練技術者の設計技術を分析し、これを基にして設計教育コースウェアを開発した。この一連の研究の中で、熟練技術者の中には共通な設計技術とも言うべき普遍的モデルが存在し、それがオブジェクト中心型記述方式で適切に表現することが判明した。また、実際の設計に際してはこの中の各プロセスの並行動作、繰り返しなど実行順序を工夫する必要があり、最終的には、そのシステムのための実行可能モデルを生成して再利用を計っている事が明かとなった。

これは、標準的なソフトウェア・プロセスである普遍的モデルの存在とそれを再利用して新しいシステムを設計する仕組みを示している。

我々が開発した設計教育コースウェアは、まず標準技術としての普遍的モデルを基にして設計技術の基本を習得させ、さらに新しい設計課題を与えてそれを実際に設計させることにより、ソフトウェア・プロセスの再利用を体験させることいわゆる実際の設計業務への応用動作の体験を与えることもこの講座の重要な目的の一つである。

我々は、この教育コースウェアをさらに充実させるとともに、将来的には、設計コンサルティング・システムへ育てていく積もりである。

6. 参考文献

- [1] L. Osterweil, "Software Processes are Software too," Proceeding of the Ninth International Conference on Software Engineering, Monterey, California, pp. 2-13, April, 1987.
- [2] T. Katayama, "A Hierarchical and Functional Software Process Description and its Enaction," Proceeding of the 11th International Conference on Software Engineering, pp. 343-352, 1989.
- [3] B. W. Boehm, "A Spiral Model of Software Development and Enhancement," ACM Software Engineering Notes, vol. 11, no. 4, pp. 14-24, August, 1986.
- [4] 望月、山内、片山他：ソフトウェア・プロセス—実時間処理システムにおけるケース・スタディー：情報処理学会研究報告、90-SE-71、pp. 139-148 (1990)
- [5] 望月、山内、片山他：ソフトウェア・プロセスの設計教育用ツールへの適用及び評価：情報処理学会研究報告、90-SE-73、pp. 83-90 (1990)
- [6] 望月、山内、片山他：ソフトウェア・プロセスの分析及び評価：第41回情報処理学会全国大会論文集 2G-8、(1990.09)
- [7] 望月、山内、片山他：ソフトウェア・プロセスを利用した教育用ツールの開発：第41回情報処理学会全国大会論文集 2G-9、(1990.09)
- [8] S. Mochizuki, A. Yamauchi, T. katayama et. al., "Applying the Software Process to the Instruction Tool in System Design," Proceedings of 6th International Software Workshop, 1990.
- [9] 望月、山内：リアルタイム・システム開発のソフトウェア・プロセス：ソフトウェア・プロセス・ワークショップ (1991.02)
- [10] 望月、山内、片山：ソフトウェア・プロセスによるシステム設計教育・訓練の試行および評価：情報処理学会研究報告、91-SE-78、pp. 49-57 (1991.03)
- [11] 望月、山内、片山他：ソフトウェア・プロセスの分析及び評価：第42回情報処理学会全国大会論文集 7S-7、(1991.03)
- [12] 望月、山内、片山他：ソフトウェア・プロセスを利用した教育用ツールの開発 (その2)：第42回 情報処理学会全国大会論文集 7S-6、(1991.03)
- [13] S. Mochizuki, A. Yamauchi, T. katayama, "Analysing and Evaluating Fundamental Design Process of Checkout System for Artificial Spacecraft," Proceedings of The Fifteenth Annual International Computer Software and Applications Conference (COMPSAC'91), 1991.
- [14] T. katayama, S. Mochizuki, "What Has Been Learned From Applying A Fomal Process Model To A Real Process," Proceedings of 7th International Software Process Workshop, 1991.