

SEWBの教育体系

Curriculum of Software Tool SEWB

ソフトウェアの生産性は、技術者個人の資質によるところが非常に大きいですが、こうした中でも開発手順の標準化、開発支援ツールの適用により、高い生産性の確保と安定化が図られてきた。しかし、ソフトウェアの非可視性¹⁾などの性質から、現状では開発支援ツールによるソフトウェア生産の完全自動化は難しく、人間への依存度は依然として高い。本論文では、この点から開発支援ツールを使った生産環境で、生産性を左右するヒューマンファクタに着目し、そこから必要とされる教育要件を明らかにした。教育要件の分析は、人間と開発支援ツールの相互作用をベースとしたコミュニケーションモデルを設定し、各職能ごとに行った。これに基づきSEWBの教育体系を決め、プログラマ、設計者と管理者に対する教育科目を開発した。

山崎政志* Masashi Yamazaki

益子宗皓* Muneaki Mashiko

1 緒言

ソフトウェアの生産性を上げるための手段はこれまで数多く検討されてきたが、理論的裏付けに欠けるもの、システムライフサイクルを通して一貫性に欠けるものが多い。こうした中で、1968年NATO(北大西洋条約機構)の協議会で「ソフトウェア工学(Software Engineering)」という言葉が生まれ²⁾、ソフトウェア生産を工学として扱うようになった意義は大きい。このことは、ソフトウェア生産に他の工学と同様に、理論的な基礎と現実的な規則に従った生産方法を確立しようというものである。日立製作所も、これらの考え方をベースとしたシステム開発技法として、HIPACE(Hitachi Phased Approach for High Productive Computer System's Engineering)を開発し、昭和56年から提供している。本開発技法は、システム開発の標準手順、開発技法、それに開発支援ツール群から構成されている。SEWB(Software Engineering Workbench)はこのHIPACEのツールの一つであり、システム設計から単体プログラムテスト工程までを支援するツールである。

ソフトウェア作成作業は、本質的にシステム設計者やプログラマという人間の知的な作業に依存する部分が多いため、生産の完全自動化が困難である。したがって、開発支援ツールは人間の介入をベースとし、その思考過程を支援するものという考え方に立つべきである。この点からすると、人間と開発支援ツールのインタフェースで、その「使いやすさ」の生産性に及ぼす影響が大であると考えられる。そこで本論文では、開発支援ツールのヒューマンファクタに焦点を絞り、効率よくソフトウェアを生産する上で必要とされる知識・技能を洗い出した。更に、それら知識・技能を効率よく修得するための教育要件を定義し、具体的にSEWBについて、その

教育カリキュラムを設定し、教育科目を開発・改訂した件について述べる。

2 ソフトウェア生産におけるヒューマンファクタ

ソフトウェアの生産は、人間と開発支援ツールなどを含めた各要素との協同作業で行われる。そこで必要とされる構成要素は、ソフトウェア生産設備と生産体制に大別できる(図1)。ソフトウェア生産設備とは、ソフトウェアを生産するための技法と開発支援ツール群である。ソフトウェア生産体制とは、生産を行う組織に関するものであり、作業チーム編成、作業手順、手続き、規則などである。

そこで本論文では、ヒューマンファクタを、人間とこれらの各要素間の相互作用で、人間の思考や行動に影響を与える要因と定義する。ここで影響とは、主としてソフトウェア生産性にプラスになるか、マイナスになるかを言う。ソフトウェア生産性に影響を与える要因については、既に調査結果として文献3)で述べられている。ここでは、それをベースに更に追加、詳細化した。図2にそれを示す。ヒューマンファクタは大きく次のように分類することができる。

(1) 物理面

人間と生産設備の物理的接点であり、キーボード、アイコン、マウス、ディスプレイ画面に対する感覚的なものである。文献4)での分析結果である「使いやすさ」(表1)を引用するならば、同表の主として「読みやすさ」、「書きやすさ」の範ちゅう(疇)である。

(2) 論理面

人間の思考と生産技法・ツールの論理的接点である。操作コマンドの概念・文法、仕様を表現する文章・表・図形の記

* 日立製作所コンピュータ事業部教育本部教育センター部

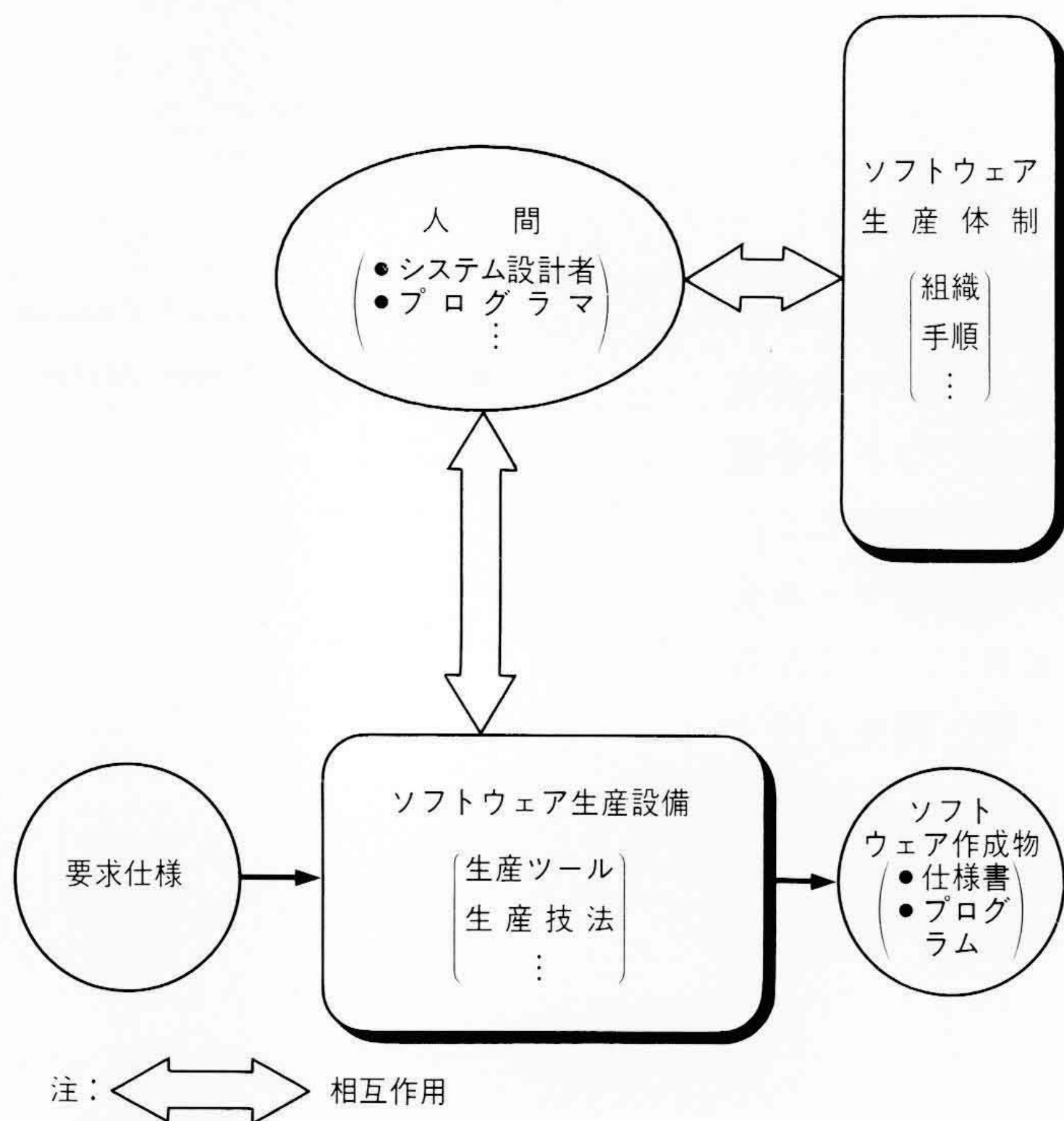


図1 人間とソフトウェア生産要素の関係
人間と生産設備・体制の相互作用で行われる。ソフトウェア生産は、

法などであり、人間の頭脳と技法・ツールの論理的なやり取りのことである。表1では主として「表現力の高さ」、「習得のしやすさ」、「問題解決の手助けになる」の範ちゅうに入る。

(3) 心理面

人間と開発支援ツールの相互作用で、人間の情緒・感情に起因する要因である。例えば、操作結果に対する応答の適時性、誤操作に対するエラー処理の適切さなどである。情緒・感情は人により差はあるものの、開発支援ツールを使って仕事をする上で重要な要素である。

(4) 組織面

人間とソフトウェア生産体制の相互作用で、主として人間の動機づけ、モラルに影響を与える要因である。例えば作業チーム編成、開発手順、プロジェクト管理の方法などがある。

3 ソフトウェア開発支援ツールの教育要件

3.1 人間と開発支援ツールのコミュニケーションモデル

ソフトウェア開発支援ツールの教育要件を洗い出す前に、人間と開発支援ツールの役割、及びその関係を分析した。両者がそれぞれの役割りを果たすためには、その間に「一定のルール」が必要である。例えば、開発支援ツールがその機

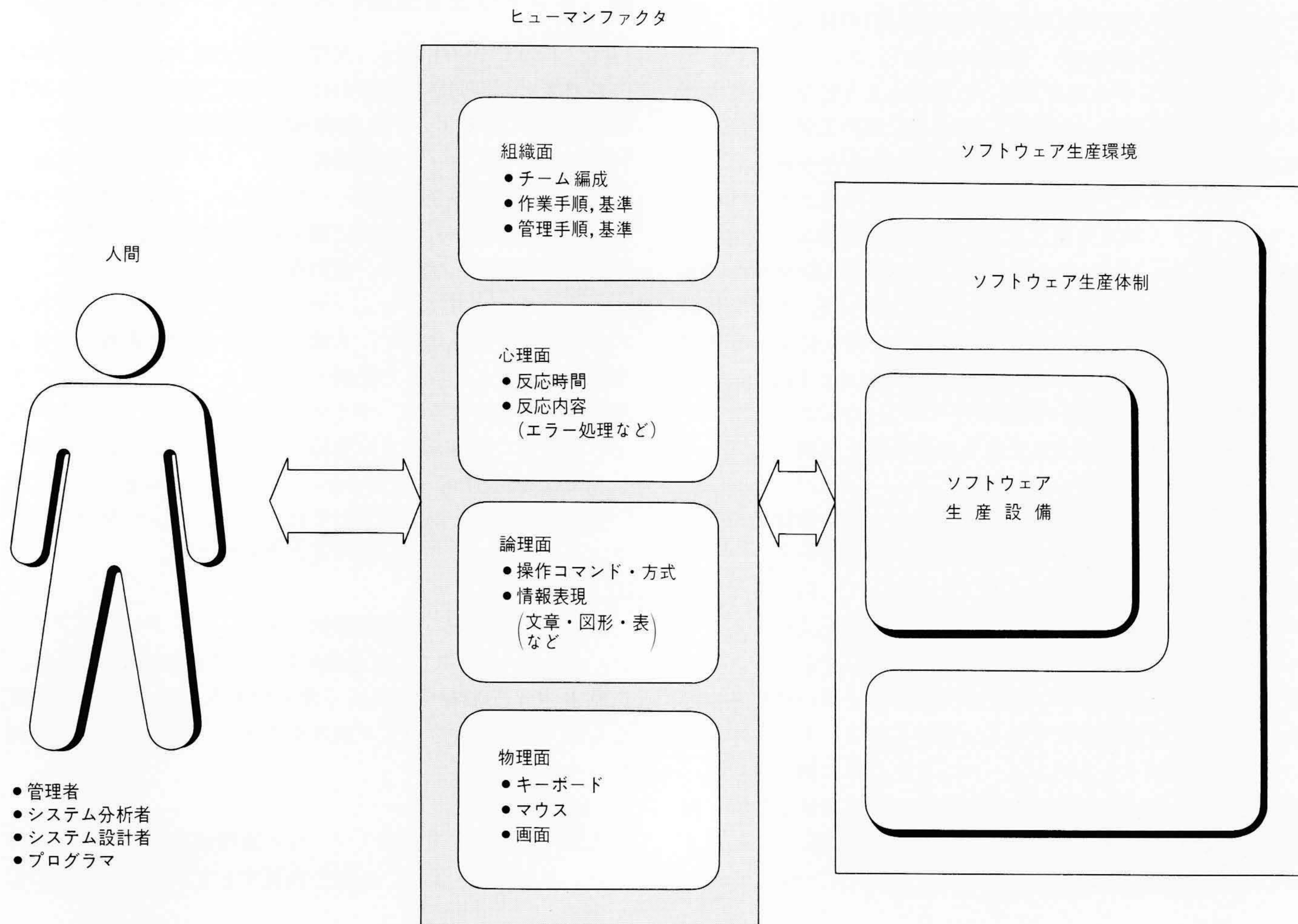


図2 ソフトウェア生産とヒューマンファクタ
ヒューマンファクタとは、人間と生産設備・体制の相互作用で、人間の思考や行動に影響を与える要因を言う。

表1 「使いやすさ」の各側面 「使いやすさ」は物理的(感覚的)な要素と論理的、心理的な要素から成る(文献4から引用)。

側面	要因	要素
習得しやすさ	覚えやすさ	覚えるべきものの数が少ない 覚えるべきものに新奇なものが少ない
	理解しやすさ	内部構造が単純である 内部構造が明確である 例外処理が少ない(一貫性) 論理がしやすい
	間違えにくさ	混同しやすい事項が少ない 主観の入る余地がない
読みやすさ (出力のわかりやすさ)	一覧性	文字の大きさ 利用平面の大きさ 量の少なさ 適当な空白がある 流れがわかる
	理解しやすさ	(習得しやすさ項参照)
	間違えにくさ	(習得しやすさ項参照)
問題解決の手助けになる	考えやすさ	思考の刺激となる 思考過程にあっている 思考過程を細分化する助けとなる 確認しながら思考を進められる
書きやすさ (入力のしやすさ)	手間の少なさ	量としての手間が少ない 重複が少ない
	入力の自由さ	制約事項の少なさ
	変更しやすさ	修正しやすい 拡張しやすい
	間違えにくさ	(習得しやすさの項参照)
表現力の高さ	—	多くのことが実現できる 機能が多い

能を実現するため、人間に行わせたいアクションであり、概念、表現・記法、命令や文章の記述方法、キーボード・マウスの操作方法などが該当する。人間と開発支援ツール間の関係は、通信ネットワークでのOSI〔ISO (International Organization for Standardization code)の開放形システム間相互接続〕のコミュニケーションモデルに相当し、「一定のルール」はプロトコルに類似するものである。人間と開発支援ツールの機能を対比させ、その間のコミュニケーションのモデル化を行った結果を図3に示す。ここでは階層を、物理層、操作層、表現層、管理層に分けた。

3.2 ソフトウェア開発支援ツールの教育要件

図3のコミュニケーションモデルから、各層での「一定のルール」を明確にすることが、言い換えると人間に対する教育項目を明確にすることに等しい。各層での教育項目は次のとおりである。

(1) 物理層

開発支援ツールのキーボード、マウス、アイコンなどの操作に関する項目であり、主としてハードウェアの機能の意味と操作方法を規定するものである。車の発進、停止、ハンドル操作に対応する最も基本的な操作であり、これだけの教育は意味を持たない。

(2) 操作層

ソフトウェア生産を行う上で意味のある最小単位の操作を規定するものであり、テキストエディタによる編集などがこれに当たる。操作コマンドの機能と文法、画面の表示意味をベースとし、ソフトウェアの作成、修正のルールが該当する。一般に操作教育とはこの層を指す。

(3) 表現層

文章、表、図などによるソフトウェアの表現を規定するものであり、だれもが同じく一意的に解釈できることと、その作成、修正の方法が容易であるということが重要である。こ

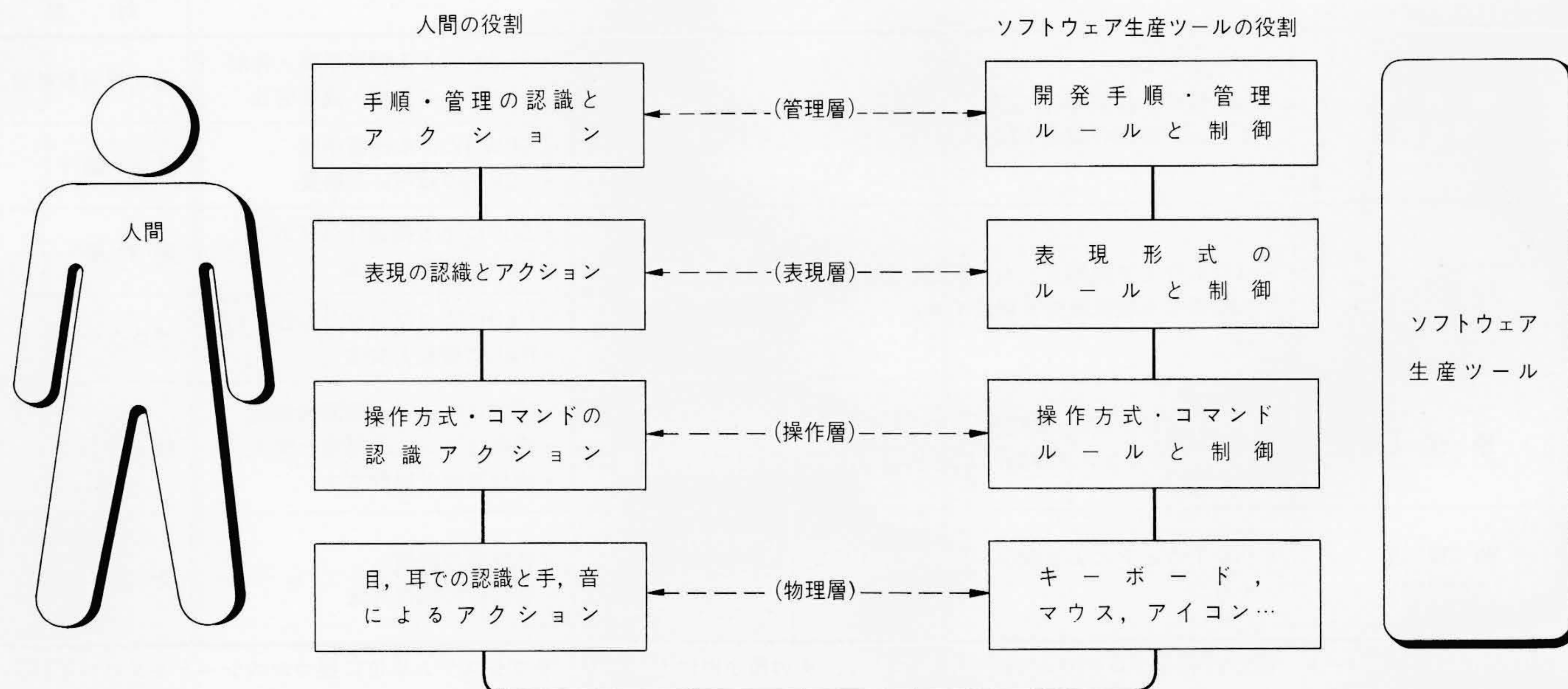


図3 人間とソフトウェア生産ツールのコミュニケーションモデル 人間とソフトウェア生産ツールの間には互いの役割を果たすために「一定のルール」が存在する。

の層に対しては、システム分析者、設計者、プログラマに対し、その概念と方法、記述のためのルールを中心とした設計技法教育が必要となる。

(4) 管理層

この層はシステム分析から運用工程までを含めた作業手順・基準、管理手順・基準に関する規定である。具体的には、開発工程での進捗管理、ソフトウェアの品質管理の手順・基準と、開発支援ツールの運用手順・基準などである。これらはプロジェクト管理を行う人及び開発支援ツールの導入と運用などの環境設定を行う人に対して必要となる教育である。

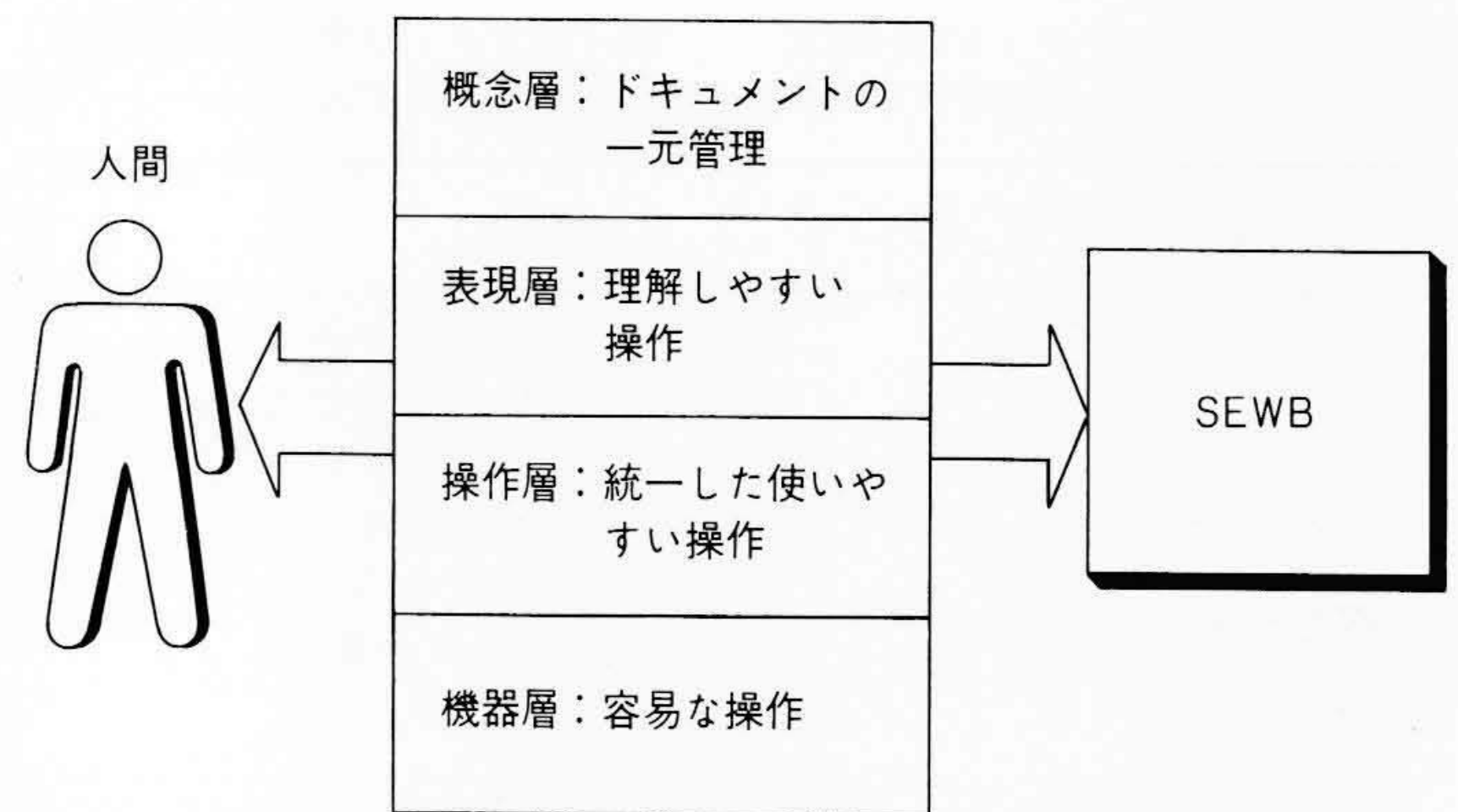
3.3 SEWBの教育要件

前章で展開したソフトウェア開発支援ツールに対する教育要件を、具体的にSEWBに当てはめてみる。SEWBでは図4に示すコミュニケーションモデルを設定し、人間と開発支援ツールの対話方式を決めている。同図で、機器層は先に述べた物理層に対応し、概念層は管理層に含まれる。SEWBでのコミュニケーションモデルの各層での規定項目と、それを具体化する手段及びそのために必要とされる教育要件をまとめたものを表2に示す。

4 SEWBの教育カリキュラム

4.1 SEWBの教育体系の考え方

ソフトウェア開発技法・ツールは、ソフトウェアライフサイクルを通して、一貫した標準手順の中に組み込まれているべきである。HIPACEはこれを実現したものである。HIPACEでの各工程と開発技法、主な開発支援ツールを表したものが図5の上部である。



注: 略語説明 SEWB(Software Engineering Workbench)

図4 SEWBのコミュニケーションモデル SEWBは4階層を持ったコミュニケーションモデルを設定し、対話方式を決めている。

HIPACEの発表と同時に、日立製作所コンピュータ事業部教育本部教育センタ部では、この開発技法を中心とした「システム技術講座」を開設し実施してきた。ソフトウェア生産に関係する主な教育科目を図5の下部に示す。これらの科目はHIPACEでの標準手順に基づいた技法を教育するものであり、開発支援ツール教育のベースとなるものである。

開発支援ツールに対する教育は、職能別に次のような考え方を基本とした。

(1) プログラマ、システム設計者

図3での操作層、表現層を中心とした教育に加え、実機を使った演習を行うことによる教育内容の具体化、知識の定着化、更に開発支援ツールに対する親近感を持ってもらうこと

表2 SEWBのコミュニケーションモデルと教育要件 SEWBに必要な教育はコミュニケーションモデルの各層に対し表のようになる。

コミュニケーションモデル	各層の規定項目	具体化する手段例	教育要件	
			教育要件	対象
管理層 (概念層)	・システムの仕様やプログラムなどの、ソフトウェア情報の構造を規定する。	・グループ ・シート	・SEWBによる開発手順・体制 ・システムの管理、運用方法	導入担当設計者
			・SEWBによる作業手順 ・SEWBのねらいと特長	共通
表現層	・ソフトウェア情報を、人間の目に見えるよう表現する表現形式を規定する。	・SDF	・SDFによる構造化分析方法 ・SDFの機能と文法	設計者
		・PAD	・PADによるプログラム設計方法 ・PADの機能と文法	プログラマ
操作層	・ソフトウェア情報を対話形式で編集するときの画面、コマンドの操作方法を規定する。	・操作命令＝ 目的語＋述語	・操作コマンド・画面の機能 ・操作コマンドの種類と文法 ・操作手順と操作方法	共通
物理層 (機器層)	・ハードウェアによる表示・操作方法を規定する。	・マウス ・アイコン ・マルチウインドウ	・各装置の機能 ・各装置の操作方法	共通

注: 1. グループ、シート: 設計仕様やプログラムを格納するもので、その最小単位をシート、サブシステム単位に幾つかのシートをまとめたものをシートと呼ぶ。
 2. SDF(Structured Data Flow Diagram: 業務処理仕様の図式表現技法)
 3. PAD(Problem Analysis Diagram: プログラムの構造化図式表現技法)

工 程	分 析	システム計画	システム設計	プログラム設計	プログラム作成	テ ス ト	移 行	運 用	
作業項目	<ul style="list-style-type: none"> ●現状分析 ●ニーズ分析 	<ul style="list-style-type: none"> ●システム開発計画 ●費用と効果の算定 	<ul style="list-style-type: none"> ●業務仕様の設定 ●DB/DCの設計 	<ul style="list-style-type: none"> ●プログラム構造設計 	<ul style="list-style-type: none"> ●ソースプログラム作成 ●単体テスト 	<ul style="list-style-type: none"> ●組合せ、総合テスト 	<ul style="list-style-type: none"> ●移行 	<ul style="list-style-type: none"> ●業務システムの運用 ●電子計算機室の運用 	
技 法	PPDS 問題発掘整理 技法	HIPACE-SA 構造化分析 技法	HIPACE-SD 構造化設計技法		HIPACE-SP 構造化プログラ ミング技法	HIPACE テスト支援 技法			
	HIPACE SPDSプロジェクト管理手順								
HIPACE SPDSシステム管理手順									
ツ ー ル	SEWB-D		SEWB-P		EAGLE				
	EAGLE								
教 育 体 系	プログラマ、 設計者教育	HIPACEによる情報システムの分析と設計*		プログラム設計*		プログラミング 言語教育			
		SEWB/EAGLEによるシステム開発							
	管理者教育	システム開発の方法と支援ツール							
プロジェクト管理*									

注：略語説明 PPDS(Planning Procedure to Develop System：問題発掘整理技法)
 HIPACE-SA(HIPACE-Structured Analysis：HIPACE構造分析技法)
 HIPACE-SD(HIPACE-Structured Design：HIPACE構造設計技法)
 HIPACE-SP(HIPACE-Structured Programming：HIPACE構造化プログラミング技法)
 SPDS(Standard Procedure to Develop System：システム開発標準手順)
 SEWB-D(SEWB for Designer：設計者用SEWB)
 SEWB-P(SEWB for Programmer：プログラマ用SEWB)

図5 SEWBの教育体系 SEWB教育はプログラマ用、設計者用、管理者用にそれぞれ用意した。

を目的とする。

(2) 開発支援ツールの環境設定、運用を行う人

図3での管理層を中心とした教育であり、導入したユーザーが効率よくソフトウェアを生産することができる環境の設定を行えることを目的とする。

(3) 管理者

開発支援ツールを導入する上で必要とする導入効果、導入事例を中心とした教育であり、導入検討に必要な情報を提供することを目的とする。

4.2 SEWBの教育内容

SEWBの教育は、管理者に対し、導入上必要とされる項目を中心にホストコンピュータで動作するEAGLE(Effective Approach to Achieving High Level Software Productivity)を含めた「システム開発の方法と支援ツール」科目と、環境設定を行う人に対し、開発支援ツールを導入したときの環境設定に必要な項目を中心とした「SEWB/EAGLEによるシステム開発」科目の2科目を開発した。

それぞれの科目内容を表3に示す。

表3 SEWBの教育内容 SEWBの教育はホストコンピュータとの連動を含め表のような内容とした。

教育科目名	教育内容	対象者	日数
システム開発の方法と支援ツール	1. システム開発方法論と支援ツール(SEWB/EAGLE)の開発背景, ねらい, 特長 2. システム設計, プログラム開発方法と支援ツール機能 3. 導入手順と期待される効果 4. 導入事例, デモンストレーション	管理者	1日
SEWB/EAGLEによるシステム開発	1. SEWB/EAGLEのねらいと特長 2. 開発手順, 開発体制と作業分担 3. システム設計・プログラミング手順・技法とSEWB/EAGLE機能 4. システムの管理, 運用方法	設計者 プログラマ	3日

更に, 既存「プログラム設計」, 「情報システムの分析と設計」科目にSEWBを取り込み実施する予定である。

5 SEWB教育実施上の留意点と課題

SEWB教育の効果については, 昭和62年度上期現在, 管理者に対する教育を, 東京, 大阪, 名古屋の3地区で実施し, 計125人の出席を得た結果から推定する。セミナーに参加してSEWBをどう思うかという問いに対し「まず1台でもよから導入したい(20%)」, 「機能がサポートされた段階で導入したい(40%)」, 「当面導入予定はない(23%)」と関心の高いことが分かる。この結果より管理者に対するセミナーの継続と, 製品リリースとあいまって導入担当者に対する技術教育が重要であることが推定される。更にアンケートから特に留意し, カリキュラムに反映すべき貴重な意見として次のものを得た。

「SEWBを導入する上で決定しておくべき環境条件を, 明確にしてほしい。」

導入上必要な資源の見積りなどは「SEWB/EAGLEによるシステム開発」科目に含めているが, 運用基準などについては今後実例をベースに適切な例を含める必要がある。また文献5)によれば, 開発支援ツールの導入について, 「ホストコンピュータと同じメーカーの開発したもの」が43.5%, 「自社で開発する」が22.4%と, 我々メーカーに対する期待が大きいことが分かる。したがって, 運用も含めた付加価値の高い「開発支援ツール教育」がますます重要になってくるものと予

想される。

SEWBはPAD(Problem Analysis Diagram)記法を前提としており, PADの教育についても考える必要がある。PADの有効性については, 文献6)によればフローチャートの場合に比べ1.6倍の生産性向上があったとの報告もあり, その効果は広く認められている。またPADの普及状態であるが, 文献5)によれば, 日立製作所ユーザー70社のうち約27%がPADを使用しているという結果が出ている。更に状況としては増加傾向にある。よってCOBOL, PL/Iなどのプログラミング教育への取込みをどうするかという課題がある。これは現在フローチャートを使っているユーザー数75%[文献5)]の兼合いから, 当面フローチャートとPAD両方の教育が必要と思われる。

6 結 言

ソフトウェア開発支援ツールを使用する上で必要とされる教育要件を導き出す方法として, 人間と開発支援ツールの相互作用に着目し, 両者のコミュニケーションモデルを設定した。この結果, 相互作用を4階層(物理層, 操作層, 表現層, 管理層)に分類できた。各層でのコミュニケーションルールや基準から, 必要とされる教育要件を洗い出した。これをSEWBに適用し, 図5に示す教育体系を決め2科目の教育科目を開発した。更に, 一部実施した結果から開発支援ツール教育では, 特に運用基準も含めたより実践的な教育が重要であることが分かった。今後このような点を含め, より付加価値の高い開発支援ツール教育を実施していく考えである。

参考文献

- 1) M. Jackson: ソフトウェア工学を考える, 日経コンピュータ(1982.8.23)
- 2) Naur, P, et al.: Software Engineering: Report on a Conference sponsored by the NATO SCIENCE COMMITTEE, 1969
- 3) ソフトウェアエンジニアリングに関する調査—ソフトウェア生産における人間要因—, 日本電子工業振興協会(昭和58年3月)
- 4) 青山, 外: ソフトウェア開発環境, 情報処理, Vol.24, No.6
- 5) 第二回バックログ/ソフトウェア生産技術利用実態調査, 日経コンピュータ(1986.1.6)
- 6) 松本: PAD/HSPMを利用した構造化設計の採用について, HITACユーザ研究会1983年論文集(昭和58年6月)