

WWWを利用したプロセスレポートイングツールの開発

島本 勝紀[†] 松下 誠[†] 飯田 元[†] 井上 克郎[†] 青山 和之[°]
simamoto@ics.es.osaka-u.ac.jp

[†]大阪大学基礎工学部 情報工学科
〒560 大阪府豊中市待兼山町1-3

[‡]奈良先端科学技術大学院大学 情報科学研究科
[°]日立製作所 ビジネスシステム開発センター

近年、ソフトウェアプロセスに基づいた開発環境についての研究が盛んに行なわれている。我々は、プロセス内の各開発者やプロダクトの情報を、積極的に公開することによりプロセス管理を支援するツールの研究を行なっている。

本ツールは World Wide Web(WWW) 上に構築される。WWW を使用することにより、開発者が自主的に、緩い制限下で作業中に直面する状況をプロセス管理者に伝えることができる。また情報公開機構を既存のシステムに追加することで、容易に実プロジェクトへ導入できるものと考えられる。

本稿ではツールの構成方法、ツールから得られる情報を示す。また、本ツールを実験的に使用しツールの評価を行ない、今後の改良点を示す。

Process Reporting Tool Using WWW

Katsunori Shimamoto[†] Makoto Matsushita[†] Hajimu Iida[†] Katsuro Inoue[†] Kazuyuki Aoyama[°]

[†]Faculty of Engineering Science, Osaka University
3-1, Matikaneyama, Toyonaka, Osaka, 560 Japan

[‡]Graduate School of Information Science,
Nara Institute of Science and Technology

[°]Institute of Advanced Business System,
Hitachi, Ltd.

In this paper, we present a software process reporting tool using World Wide Web(WWW). The purpose of the system we propose is that the manager manages the process based on the informations from the developers. The developers can report their information with loose constraints.

Most of current software development environments have the managers manage the processes based on structured information. However, this approach is insufficient to report various situations of the software development activities. WWW allows the developers to report thier situations in thier own ways.

1 はじめに

ソフトウェア開発の大規模化、ネットワーク技術の発達により、開発拠点の分散化が進んでいる。そのような状況下で、分散開発環境下におけるソフトウェアプロセスについての研究 [1] が盛んに行なわれている。

我々はこれまでプロセス定義に従った作業誘導や進捗管理を行なう専用システムについての研究を行ってきた。そのようなシステムでは以下のような問題点があり、実プロジェクトでの運用は難しいものと考えられた。

- 開発者を拘束してしまう。
- 他のツールとの整合性を考慮しなければならない。
- 特殊なシステムの導入が必要である。
- 分散環境下では扱いにくい。

そこで現在、我々は各開発者やプロダクトの情報を積極的に公開することによりプロセス管理を行なうプロセスレポティングツールの研究 [2] を行なっている。

本ツールは Internet における情報サービスシステム World Wide Web(WWW)[3] 上に構築される。WWW は (1) 情報の公開、閲覧において優れている。(2) 既存のツールを使用して容易に分散環境を構築できる。(3) 音声、画像による情報提供ができる。という利点を持っている。この利点を生かすことにより以下のような特長をもつプロセス管理のためのツールを構築することができると考えられる。

- 開発者が作業中に直面する状況を自主的に、緩い制限下でプロセス管理者に伝えることができる。
- 情報公開機構を既存のシステムに追加するだけで、容易に実プロジェクトへ導入できる。
- 音声、画像による情報の提供ができる。

本稿では、まずツールの構成方法について述べる。その後、ツールを使用して得ることのできるプロセス情報や、本ツールの使用例を述べる。

また、我々は実験的に本ツールを使用して小規模なプログラムを作成した。この実験を通して本ツールの評価を行なう。最後に実験から得た問題点についての考察を行ない、今後の改良点について述べる。

2 WWW 上のシステム構成方針

本節では、WWW を使用したツールの構成方針について述べる。ツールの構成の際には、プロセスのモデル化を行ない、そのモデルに応じて WWW ページを生成する。

以下では本稿で使用されるプロセスモデルと、それに基づくページの構成方法について述べる。

2.1 プロセスモデル

プロセスに関する情報を効率的に収集、公開するためにはプロセスをモデル化する必要がある。本稿では図 1 で示すプロセスモデルを使用する。

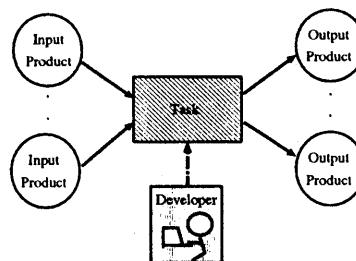


図 1: プロセスモデル

モデルの各要素は以下の通りである。

- タスク:ソフトウェア開発におけるひとまとまりの工程.1 つのタスクには 1 人以上の開発者が割り当てられる。
- プロダクト:タスクの入出力となりうる成果物。
- 開発者:タスクに割り当てられた作業員。

プロセスはタスクの集合として表現する。本プロセスモデルを使用したプロセスの記述例を図 2 に示す。

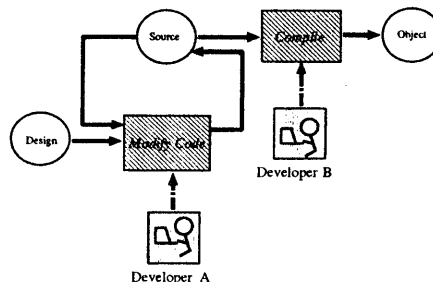


図 2: 提案モデルによるプロセスの記述例

図2ではプロセスは *Modify Code*, *Compile* の2つのタスクから構成される。*Modify Code* では Developer A が Design に基づき Source の修正を行ない Source の新たな version を出力する。*Compile* では Developer B が Source のコンパイルを行ない、Object を出力する。

2.2 モデルに基づくページの構成方法

前述のプロセスモデルに基づきツールを構成する。本ツールは以下の4種類のページとページを双方向に結ぶリンクから構成される。ページの記載内容については次節で述べる。

各ページは HxperText Markup Language(HTML)[4] で記述する。ページ閲覧者とのインターフェイスには Mosaic 等のブラウザを使用する。

図2に示したプロセスモデルから、実際にページを構成する場合、以下のようにモデルの情報を保存するページ、リンクを生成する。

- プロセスページ
プロセス全体を表すページを生成し、各タスクページとのリンクを結ぶ。
- タスクページ
各タスクに対応して1ページ用意する。図2では *Modify Code*, *Compile* のタスクページを生成する。タスクの入出力となるプロダクトのプロダクトページ、タスクを実行する開発者の開発者ページへリンクを結ぶ。
- プロダクトページ
各プロダクトに対応して1ページ用意する。図2では Design, Source, Object のプロダクトページを生成する。Source については、現行の version に修正を行ない、新たな version を生成するため、修正前、修正後の2種類のページを生成する。プロダクトの実体とのリンクを結ぶ。
- 開発者ページ
各開発者に対応し1ページ用意する。図2では Developer A, Developer B の開発者ページを生成する。

リンクについては、閲覧性を高めるために、上記以外のリンクを結ぶこともできる。

3 ページの記載情報

本節では生成された各ページに記載される情報について述べる。記載情報は主に2種類に分類される。1つ

はタスク、プロダクト、開発者の持つ属性についての情報である。もう1つはプロセス、タスク、開発者の進捗状況、連絡事項である。

3.1 属性

モデルの要素であるタスク、プロダクト、開発者はそれぞれに以下のような属性を持つ。それぞれの持つ属性は該当するページに自然語で記載される。属性の記載例を図3に示す。

- タスク：スケジュール、開始終了制限、入出力プロダクト名
- プロダクト：責任者、バージョン、型
- 開発者：割り当てられたプロダクト名、開発者名

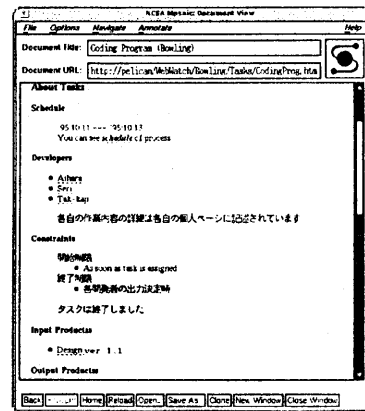


図3: 属性の記載例(タスクページ)

3.2 進捗状況、連絡事項

開発者ページ、タスクページにはそれぞれ進捗状況、連絡事項を記入する。進捗状況は、文章やメトリクスデータ等の数値で表す。連絡事項は文章で表す。

これらの情報はプロセスの進捗管理、状況理解のために必要となる。よって、進捗管理、状況理解をより円滑におこなうために、各ページに分散するこれらの情報をまとめる必要がある。本ツールでは、開発者ページに記入された進捗状況、連絡事項をタスクページに転載し、同様にタスクページの進捗状況、連絡事項はプロセスページに転載する。

以下に各ページに記載される進捗状況、連絡事項について述べる。

- 開発者ページ
開発者ページにはその開発者に割り当てられた各

タスクごとに進捗状況、連絡事項の記入欄が存在する。記入欄の例を図4に示す。開発者は記入欄に情報を記入する。

記入欄に記入された進捗状況、連絡事項はそれぞれ該当するタスクのタスクページに転載する。文章はそのままの形で、数値は表にまとめて転載する。タスクページに転載された例を図5に示す。

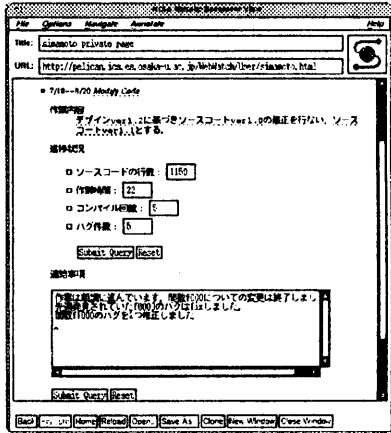


図 4: 記入欄の例 (開発者ページ)

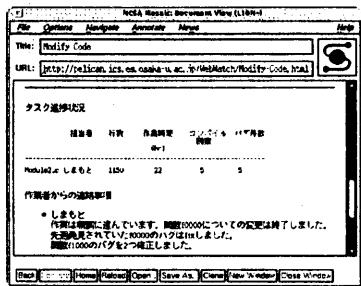


図 5: 進捗状況、連絡事項の転載例 (タスクページ)

● タスクページ

タスクページには開発者ページに記入された進捗状況、連絡事項が転載されている。それとは別にタスクの全体での、進捗状況、連絡事項記入欄が開発者ページと同様に存在する。

あらかじめ決められたタスク責任者が、転載された開発者の進捗状況、連絡事項を考慮に入れて、タスクページの記入欄に進捗状況、連絡事項を記入する。

● プロセスページ

プロセスページには各タスクごとに、それぞれ各

タスクページの記入欄に記入された進捗状況、連絡事項を転載する。

3.3 記載情報の更新, 変更

記載情報の更新, 変更は以下のように行なう。

属性情報の更新, 変更はプロセス管理者からの要請より各ページの責任者が行なう。例えばプロセス管理者からタスク開始日時の変更通知を受けとった場合、そのタスクの責任者はタスクページに記載されているスケジュールを変更する。

進捗状況, 連絡事項の変更, 更新は, 新たな進捗状況, 連絡事項の記入時に自動的に行なう。例えば開発者 A が X というタスクの進捗状況を記入すれば, X のタスクページに記載されている A の進捗状況の更新が自動的に行なわれる。

4 使用例

本節では, 本ツールの使用方法を管理者 (プロセス管理者, およびタスク責任者), プロセス内の開発者の 2 つの視点から示す。例として図2で示されるプロセスを実行するものとする。

4.1 管理者の視点からみたツールの使用方法

プロセス管理者の視点から本ツールの使用方法について述べる。

プロセス管理者はプロセスページに記載されている各タスク毎の進捗状況, 連絡事項に基づき, プロセス管理を行なう。

例えば *Modify Code* の作業が遅れ, スケジュールどおりに終了しそうでないとタスク管理者が判断すれば, *Compile* の開始を遅らせることができる。プロセス管理者は電子メール等の手段を使用して, 各タスク責任者に *Compile* の開始が遅れることを通知する。タスク責任者は連絡事項として, その旨をタスクページに記入する。タスクページに記入された連絡事項は, プロセスページに転載される。

タスク責任者も同様にタスクページに記載された進捗状況, 連絡事項からタスク管理を行なうことができる。また, タスク責任者はタスクの進捗状況, 連絡事項をタスクページに記入する役割を負う。

4.2 開発者の視点からみたツールの使用方法

プロセス内の開発者の視点から本システムの使用法について述べる。

各開発者には、それぞれの開発者ページが割り当てられる。開発者は各タスクについての進捗状況、連絡事項を開発者ページに記入する責任を負う。

開発者はタスクの実行にあたって、必要な情報を本ツールから得ることができる。例えば Developer A は Source のプロダクトページから Source の責任者や、Source の実体についての情報を得ることができる。また Design のプロダクトページを経由して Design の実体(ファイル)を得ることができる。Design について不明な点があれば、Design の責任者に電子メールなどの手段を使用して、問い合わせを行なうことができる。

また開発者はプロセスページ、各タスクページを閲覧することにより、プロセス全体の状況、プロセス上での自分の役割を知ることができ、それに応じた作業を行なうことができる。

5 実験

本ツールの有効性を確認するため、本ツールを実際使用し小規模なプログラム開発を行なった。本節では実験についての概要について述べ、実験の結果から本ツールの評価を行なう。

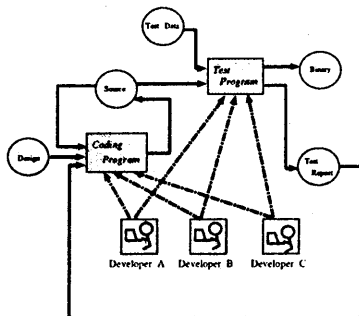


図 6: ボウリングスコア計算プログラムの作成

5.1 概要

本実験ではプロセスレポートングツールを使用してボウリングスコア計算プログラムを作成し、そのプロセス管理を行なった。

実験はプロセス管理者 1 人、開発者 3 人で行ない、プロセス管理者がタスク責任者を兼任した。開発者、管理者は地域的な分散環境下で作業を行なう状況を想定、作業についてのコミュニケーションはすべてツールまたは電子メールで行なった。

実験終了後、開発者とプロセス管理者とにインタビューを行ない、ツールについての評価を行なった。

5.1.1 タスクについて

本プロセスは以下の 2 つのタスクから成る。本プロセスのプロセスモデルを図 6 に示す。

1. Coding Program

予め用意された Design に基づき、3 人の開発者によりボウリングスコア計算プログラムのコード化を行なう。

本タスクの属性を以下に示す。

- スケジュール：'95-10-11 — '95-10-13
- 開発者：A, B, C
- 開始制限：タスク割当て直ちに実行。再実行の際は Test Program 終了後直ちに実行。
- 終了制限：Source の単体テスト終了時。
- 入力プロダクト：Design, Test Report
- 出力プロダクト：Source

プログラムを制御部、入出力部、スペア/ストライク計算部の 3 つのモジュールに分割し、各々を各開発者に割り当てる。各開発者は個々にコード化、単体テストを行なう。

Design では大域変数、制御の流れ、関数(引数、戻り値)、入出力のフォーマットが定義されている。各関数の細部は各開発者に委ねる。

Test Program の結果によりタスクの再実行もあり得る。

2. Test Program

Coding Program で生成した Source の結合を行ない、予め用意したテストデータに基づきボウリングスコア計算プログラムのテストを行なう。

本タスクの属性を以下に示す。

- スケジュール：'95-10-16 — '95-10-17
- 開発者：A, B, C
- 開始制限：Coding Program 終了後に実行。
- 終了制限：Test Report の完成時。
- 入力プロダクト：Source, Test Data
- 出力プロダクト：Binary, Test Report

全 Test Data を3つに分割し各々を各開発者に分配する。各開発者は Test Data に基づきテストを実行し Test Report を作成する。

Test Report は Test Data 番号とエラーの状況報告からなる。

5.1.2 進捗状況、連絡事項の記入について

図6のプロセスモデルからツールを構成するページを生成した。各開発者ページ、タスクページに記入される進捗状況、連絡事項を以下に示す。開発者ページへの記入は開発者が随時行なう。タスク責任者はタスクページに転載された、各開発者の進捗状況、連絡事項に基づき、タスクページに進捗状況、連絡事項を記入する。

- Coding Program
 - 数値：作業時間、コード行数、コンパイル回数、バグ件数
 - 文章：進捗状況、連絡事項
- Coding Program
 - 数値：作業時間、テストデータ数、バグ件数
 - 文章：進捗状況、連絡事項

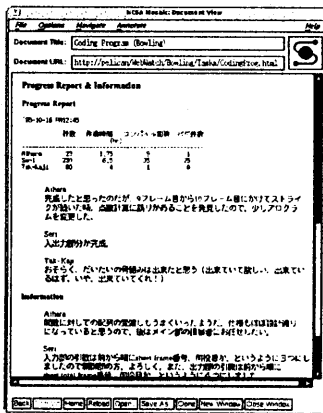


図7: 実験中の進捗状況、連絡事項(タスクページ)

実験中にタスクページに転載された進捗状況、連絡事項を図7に示す。実験時には、進捗状況、連絡事項の転載機能の実装がなされておらず、タスクページ、プロセスページへの転載はプロセス管理者が1時間に1回手動で行なった。

進捗状況、連絡事項に応じてプロセス管理者はプロセスの管理を行なう。各開発者への連絡手段として電子メールを使用した。

5.2 評価

実験の結果を基に、管理者側(プロセス管理者、タスク責任者)側、開発者側の双方の視点からツールの評価を行なう。また、実験から得た問題点についての考察を行なう。

5.2.1 管理者側からのツールの評価

プロセス管理者はプロセス実行前にプロセスの定義、モデル化を行ない、必要なページを手動で生成した。プロセスの規模が大きくなればこのページ生成にかかる手間はかなりのものになると考えられる。モデルからページを自動的に生成する仕組みが必要であると考えられる。

プロセス実行期間では、プロセス管理者は本ツールを通じて、プロセスの進捗状況を把握することができた。本実験中に発生したプロセス上の問題、プロセス管理者の対応について述べる。

- Design の変更

Design 中の各モジュール間のインターフェイス定義が曖昧であったため、各開発者から Design 変更要求が起こった。プロセス管理者は各開発者の連絡事項に応じて Design の実体、Design のプロダクトページの変更を行なった。
- Test Program 開始が1日遅れる

これは被験者がいずれも学生であったため講義等により十分な作業時間がとれなかったことによる。プロセス開発者はタスクの進捗状況から Test Program の開始を1日遅らせ、Test Program のタスクページの変更を行なった。

プロセス管理者は、タスクページの進捗状況、連絡事項、あるいはプロダクトの実体そのものから、タスクの進捗状況を把握することができた。しかし、プロセスが大規模になれば、文章、数値だけでプロセス、プロダクトの状況を把握できるかどうかは疑問である。また今回の実験では、進捗状況、連絡事項の転載、更新を手動で行なっていたため、その作業に殆んど時間をとられた。進捗状況、連絡事項の転載、更新の自動化は本ツールにとって必ず必要となるものである。

次に Coding Program についての各開発者の進捗状況、連絡事項の記入回数を表1に示す。表中の空白は開発者がその日作業を行なわなかったことを示す。

開発者 B はタスク開始当初は進捗状況、連絡事項の記入を比較的頻繁に行なった。しかし、タスクが進むにつれ1日の作業終了時にまとめて記入を行なうように

表 1: Coding Program における開発者の進捗状況, 連絡事項記入状況

日付	A			B			C		
	進捗状況		連絡事項	進捗状況		連絡事項	進捗状況		連絡事項
	数値	文章		数値	文章		数値	文章	
10/11	1	1	1	4	1	2	1	1	1
10/12	3	2	3	1	1	1	5	3	5
10/13	2	1	2	1	1	1	3	1	1
10/16							1	0	0

なった。これはタスク開始当初は、作業の方針が掴めず、頻繁にタスク責任者との連絡を行なったためであり、作業方針が決まった後は作業のみに集中したためである。

各開発者が1日当たり1回程度の進捗状況報告を行なう場合、本ツールではプロセス管理者の把握するプロセスの状況が、必ずしも現在のプロセス状態を反映していないということが起こり得る。しかし、記入回数の増加は開発者の負担を増大する。この両者のトレードオフは重要な問題となる。

10/12には連絡事項の更新が多く行なわれている。これは上述の各モジュール間の Design についての問い合わせが多く発生したためである。Design についての開発者間、開発者と管理者間の問い合わせがツール上、あるいは電子メールにより行なわれた。

本ツールでは、連絡事項を通して管理者と開発者間のコミュニケーションをとることができる。しかし、本ツールへ記入する連絡事項と電子メール等を使用して、直接連絡をとることの境界が曖昧であったため、本ツールに記載された連絡事項はあまり意味をなさない場面が多かった。

5.2.2 開発者側からのツールの評価

実験終了後、各開発者にインタビューを行なった。以下に開発者からの本ツールについての意見をまとめる。

1. 作業に必要な情報がどこにあるかわかりづらい。
2. 作業に熱中すると開発者ページへの記入を忘れてしまう。
3. タスクページ、プロセスページがいつ更新されたのかわからない。決まった時刻に更新すれば、その時間にページを見に行けば良い。またその時間に合わせて、ページ記入をすれば良い。
4. 開発者同士の連絡のやりとりには、本ツールは不向きである。

5. すぐに返答をしてもらいたいような連絡事項にはメールの方が適している。本ツールとメールとの境界をはっきりさせるべきだ。

1については、プロセス実行前のツールに対する説明が不十分であったため生じた問題であると考えられる。しかし、各ページの閲覧を円滑に行なうため、ページ間の誘導機能を充実させる必要がある。

2, 3については、一定時間毎にページ記載内容の転載を行なうことにする。それにより、開発者は転載時間に合わせて、ページへの記入を行なうことができる。また、転載時間に合わせて各ページを閲覧すれば良い。

4,5については、本ツールは開発者からの情報を閲覧することに適しており、開発者間、あるいは開発者と管理者間のコミュニケーションには電子メールなどの別の枠組を利用することを考えている。その際、ページに記載する連絡事項と、別の枠組でやりとりする連絡事項の区別を明確にする必要がある。

6 改良点

本節では、実験から得られた問題点に基づくツールの改良点について述べる。

● ページ生成の自動化

プロセスモデル、あるいは別に定めるプロセス記述から自動的にページを生成する。すなわち、プロセスモデル内の各要素間の関係、並びに属性定義を与えることによりページ生成を自動的に行なう。このためには現行の属性よりさらに厳密な属性の定義が必要となる。

ページの自動生成によりプロセス開始前の手間を大幅に省くことができると考えられる。また、プロセス内においてプロセス変更が生じた場合にも、各ページの変更を容易に行なうことができる。

- ページに記入する進捗状況、連絡事項の定義の明確化

記入される進捗状況、連絡事項の定義を明確に行なう。すなわち、ページに記入する連絡事項と、電子メールなど他の枠組で行なう連絡の区別を明確にする。

区別が明確にされていることにより、開発者は情報に応じた連絡手段を選択することができ、よりの確に作業状況を他の開発者、管理者に伝えることができる。また、大規模なプロセス内でも的確な情報伝達ができると考えられる。

- 進捗状況、連絡事項記入の促進
進捗状況、連絡事項の記入を促進するため以下の点について改良を行なう。

- 画像、音声によるプロセス情報の伝達
より柔軟にプロセスについての情報を伝達するために、音声、画像を積極的にとり入れる。画像を利用して、進捗状況、連絡事項を報告することにより、文章では難しい状況の報告を容易に行なうことができる。

Hot Java[5]等のブラウザのもつ機能を使用することにより、画像、音声を利用した進捗状況等を記載できると考える。

- メトリクスデータの自動収集
現行のツールではメトリクスデータを各開発者が直接記入していた。一定時間毎にメトリクスデータを自動で収集し、それをページに自動的に記載することにより、開発者の負担を軽減できる。また、作業の状況が的確にページに反映される。

- ページの更新時間の明確化
現行のツールでは、記入と同時に進捗状況、連絡事項の転載を行っていたが、これをある決まった時間毎に行なうようにする。これにより進捗状況等の記入が促進されるものと考えられる。すなわち、開発者はその時間に合わせて、進捗状況、連絡事項の更新を行なうことができる。更新回数は開発者に委ねられる。また、その時間にあわせて、他のページを閲覧することにより、プロセス、タスクの状況を把握することができる。

しかし、この場合緊急を要する連絡事項については、早急に更新、転載を行ないそれを管理者、各開発者に伝える必要がある。これに

については連絡事項に優先度を付ける、または別の枠組により連絡を行なうことで解決できると考える。上述の連絡事項の定義とあわせて、よりツールにあった手法を選ぶようにする。

- 進捗状況、連絡事項の履歴の保存より正確にプロセスの状況を把握するためにはページに記載される進捗状況、連絡事項の履歴をとる必要がある。進捗状況、連絡事項をその記入時間を付随して保存するようにし、該当するページから保存された過去の進捗状況、連絡事項を閲覧できるようにする。

7 まとめ

本稿では、WWWを使用したプロセスレポートングツールの実現方法について述べた。プロセスモデルからのツール構成方法、ツールを構成するページに記載される情報について述べ、本ツールの使用例を示した。

また、本ツールを使用して行なった、実験の概要について述べ、実験に基づくツールの評価を行なった。最後に実験から得られた本ツールの改良点を示した。

今後、本稿に示した改良点に沿って本ツールの改良を行ない、さらに規模の大きなプロセス上での実験を行なっていきたい。

参考文献

- [1] 落水浩一郎: "ソフトウェアプロセスに関する研究の概要", 情報処理, Vol. 36 No. 5, pp 379-391(1995).
- [2] 島本勝紀, 松下誠, 飯田元, 井上克郎: "マルチメディアツールを利用したプロセス環境の実現方法", 情報処理学会第51回全国大会, pp 271-272, 1995
- [3] Nathan Torkington: "World Wide Web Primer", 1993.
- [4] Laura Lemay 著, 武舎広幸, 久野禎子, 久野靖 訳: "HTML 入門-WWW ページの作成と公開", プレンティスホール, 1995
- [5] "Java: Programming for the Internet", <http://java.sun.com/>
- [6] 溝口文雄, 大和田勇人, 柳田正博: "情報サービスシステムによる教育支援環境の構築と評価", 日本ソフトウェア科学会第11回大会, pp. 373-376(1994).