

標準プロセスの検討におけるプロセス・ モデルの活用事例

宇宙航空研究開発機構
情報・計算工学センター

静永 誠

発表概要(全体)

プロセス・モデルの概要・意味・定義など

プロセス・モデル作成事例

プロセス・モデル活用を試みた所感

発表概要(1)

プロセス・モデルの概要・意味・定義など

プロセス・モデル作成事例

プロセス・モデル活用を試みた所感

プロセス・モデルの意味・定義

「モデル」の意味(岩波書店「広辞苑 第五版」より):

モデル : 型・型式

型 :

- 個々のものの形を生ずるもととなるもの、または個々の形から**抽象**されるもの。
- ものを類に分けた時、それぞれの**特質をよく表した**典型。そのような形式・形態。タイプ。パターン。

プロセス・モデル:

プロセスを、その**特質を良く表す**
ように**抽象化**したもの

キーワード:

- 抽象化
- 複雑さを減らす
- 重要な部分に影響しない詳細を取り除く
- 重要な部分を示す

過去の論文での説明例([Curtis,Kellner,Over1998]より):

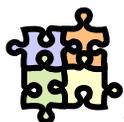
A model is an **abstract representation of reality** that excludes much of the world's infinite detail. **The purpose** of a model is **to reduce the complexity** of understanding or interacting with a phenomenon **by eliminating the detail that does not influence its relevant behavior**. **Therefore, a model reveals what its creator believes is important** in understanding or predicting the phenomena modeled.

...

A **process model** is an **abstract description of an actual or proposed process** that represents selected process elements that are **considered important to the purpose of the model** ...

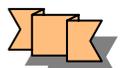
プロセス・モデルの目的

プロセス・モデルを作成／使用する主な目的は以下のように整理することが可能([Rolland1998]参照)。



• プロセス記述 (DESCRIPTIVE)

プロセスがどのようにして実際に実施されるかを示し、改善点を見つける。



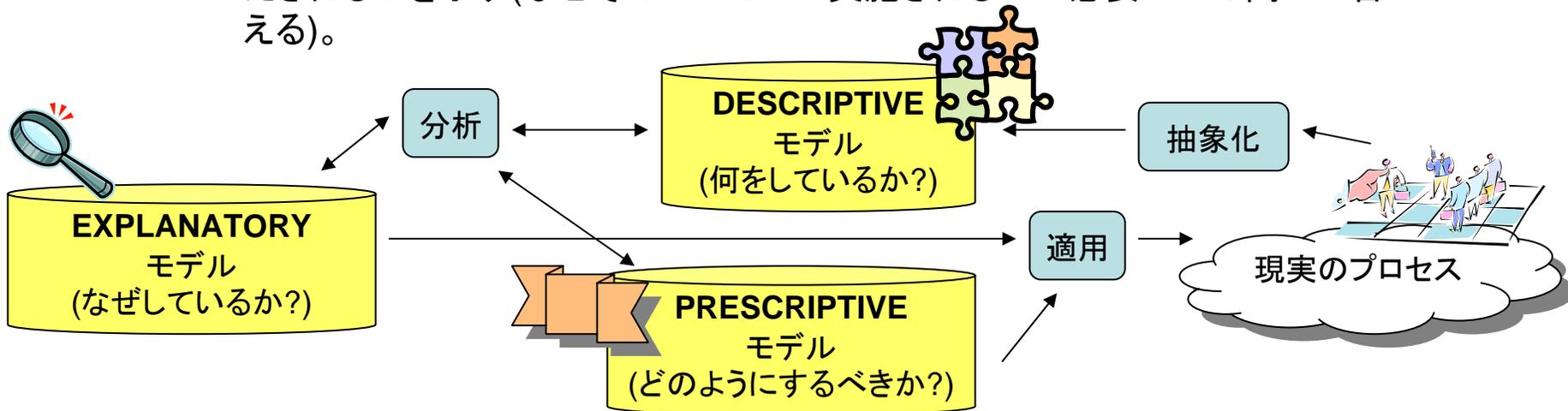
• プロセス規定 (PRESCRIPTIVE)

あるべき姿を示し、作業が上手く実施されるよう導く。



• プロセス説明 (EXPLANATORY)

プロセスが実施される理由を説明する。プロセスによってどのような要求が満たされるかを示す(なぜそのプロセスが実施されるか？ 必要か？ の問いに答える)。



プロセス・モデルの表記方法

“Software Processes Are Software Too”

[Osterweil1987]

プロセスのモデル化を扱った、初期(最初?)の論文

ソフトウェア・エンジニアリングの技法を使って、ソフトウェア・プロセスをモデル化する考え方を提示(プロセス・プログラミング)

コンピュータ・サイエンスはソフトウェアという目に見えない(純粹に概念だけで構成される)対象を扱ってきたため、プロセスという目に見えない対象を扱うことにも適していると主張

→ Pascalに類似した表現でプロセスをモデル化した例を提示

以後、ソフトウェア開発で様々なモデル化が提案されているのと同様に、プロセス・モデルの表記方法も様々なものが提案されている。

ETVX, UML, IDEF0, APPL/A ...

“6th International Software Process Workshop”では、18種の表記方法が比較されているという報告がある

[http://www2.umassd.edu/SWPI/STARS/SPD/section_4.html]

発表概要(2)

プロセス・モデルの概要・意味・定義など

プロセス・モデル作成事例

プロセス・モデル活用を試みた所感

プロセス・モデル作成事例

JAXA／宇宙機メーカ双方を含んだ衛星搭載ソフトウェア開発に対する標準プロセスを検討する作業の中で、プロセス・モデルを作成／活用した事例を紹介。



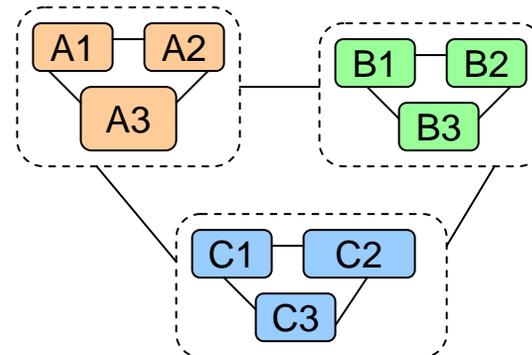
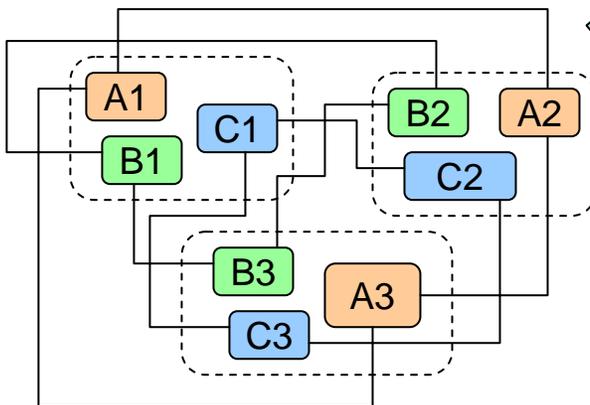
現実のプロセス

現状(AS-IS)モデルの検討

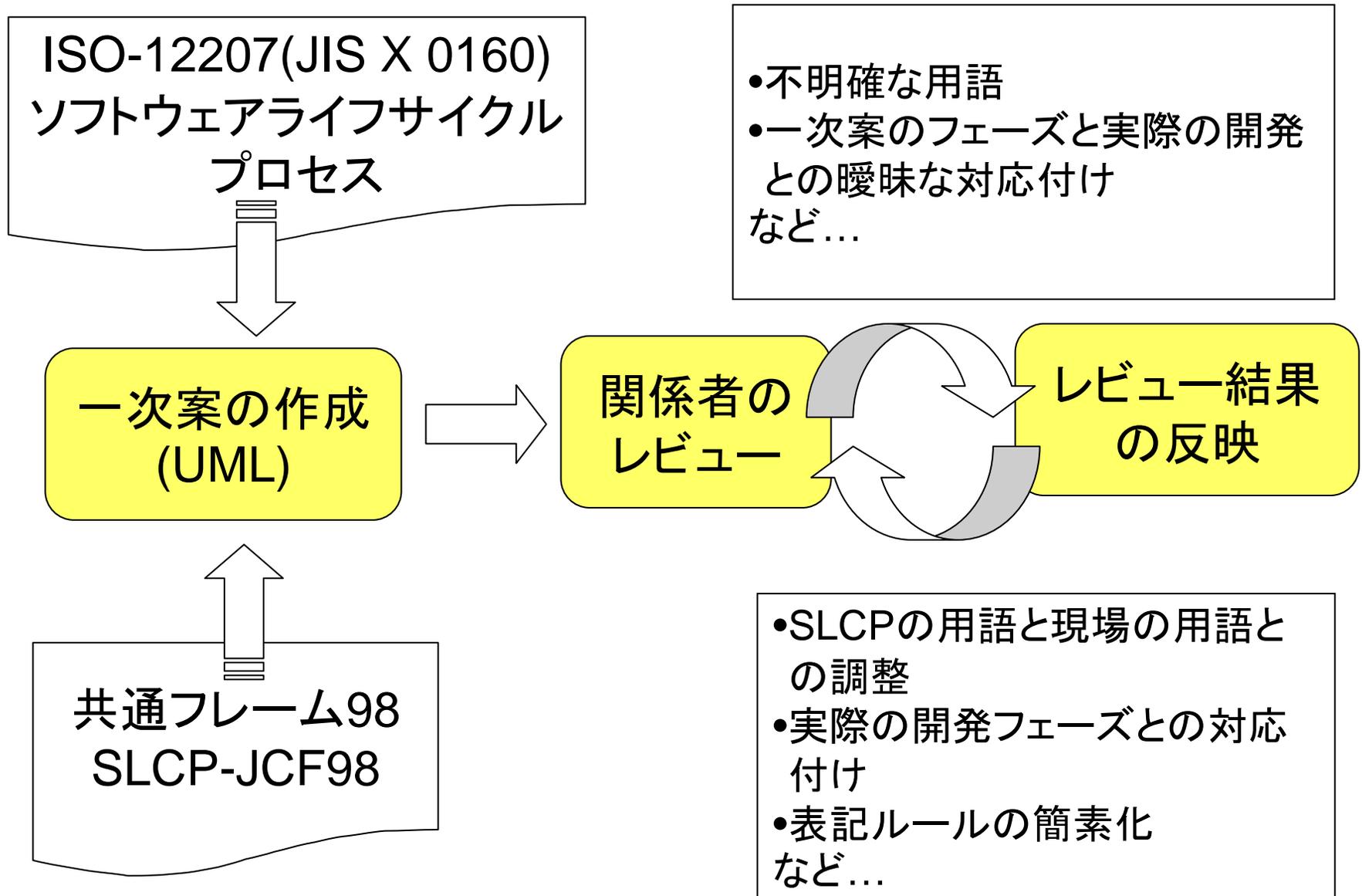
- 現状分析におけるモデルの活用
- DESCRIPTIVEモデルの作成

標準用(TO-BE)モデルの検討

- あるべき姿の検討におけるモデルの活用
- PRESCRIPTIVEモデルの作成



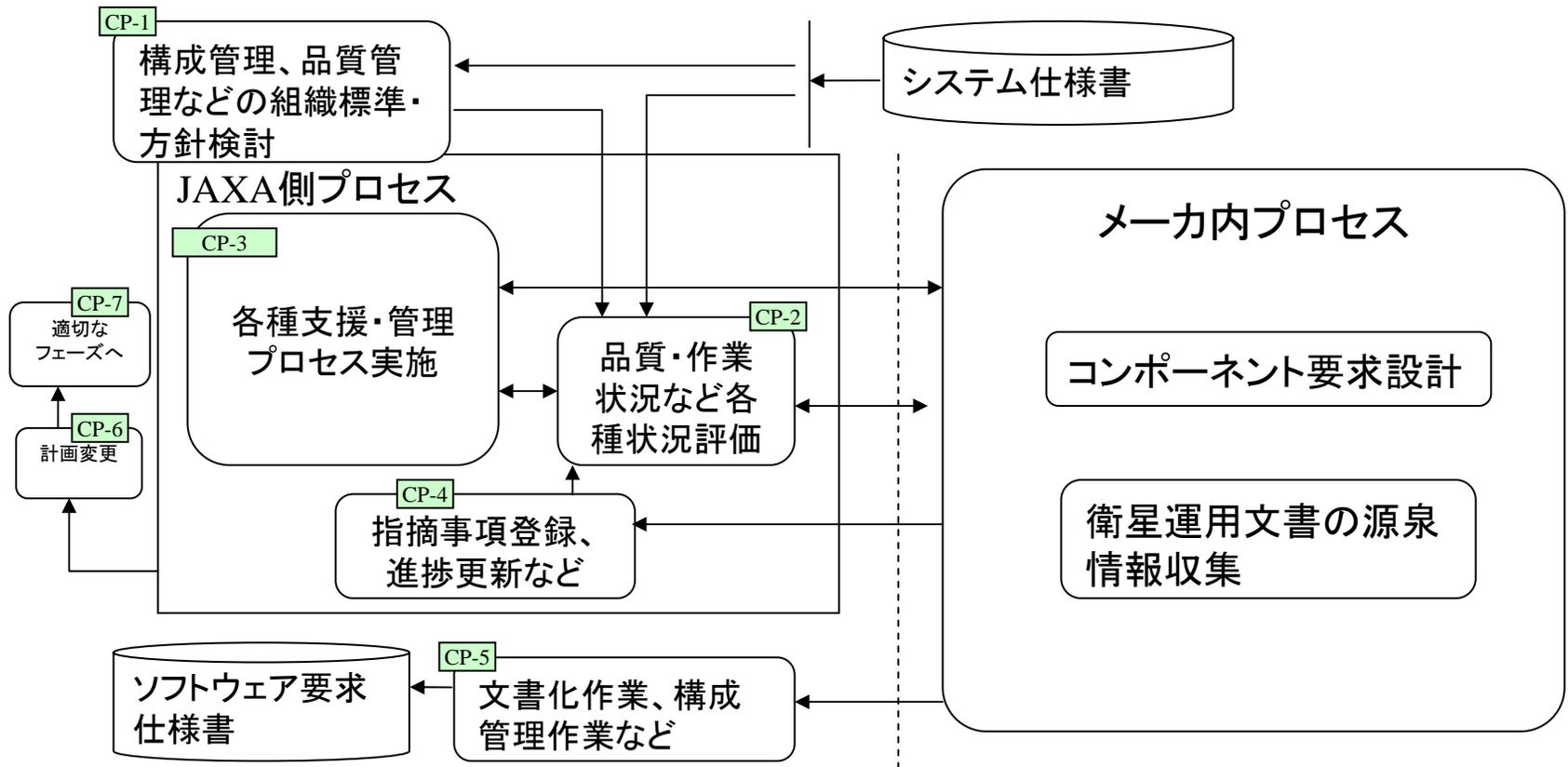
現状 (as is) モデルの作成 (手順)



現状 (as is) モデルの例

一次案からの主な変更点

- 各フェーズのプロセス・フローにて、JAXAで実施している部分を含め、管理系の作業を多く追加した。
- 現状を完全に示すのではなく、現状分析に用い意見を収集することが重要であったため、表記法については厳密なルールは定めずに作成した。
- 一次案に対するコメント、検討会での意見、設計基準文書、開発担当者へのインタビュー結果をもとに、フローや用語を実際の作業に近いものにした。



現状 (as is) モデルの作成 (成果)

現状 (as is) モデル資料から得られた成果

- 開発プロセスに関する資料を基にした意見の獲得
 - 現行プロセスの明確化
 - ソフトウェア開発各フェーズ毎の問題点の識別

図の説明に終始せず、プロセスそのものに関する意見を獲得することが出来た。
- 開発プロセスに対するより広範囲な意見の獲得
 - 将来的な視点からのコメント(仕様のレイヤー化の必要性、など)
 - 再プログラミングに関する意見(衛星の長寿命化に伴う要求の変化、など)

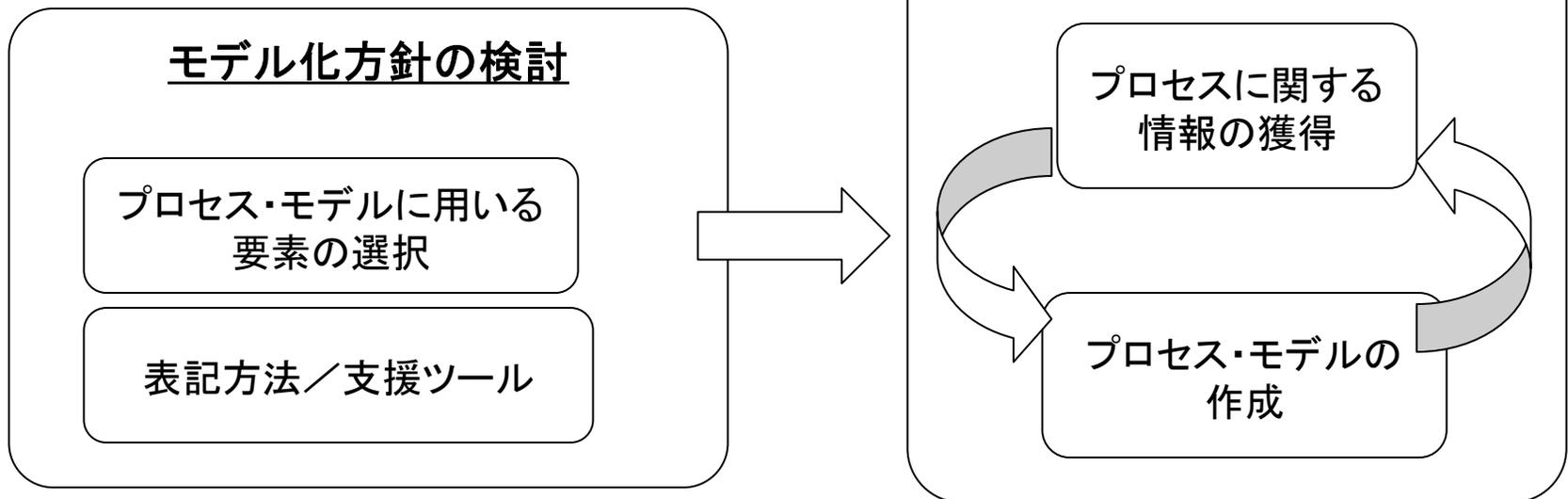
図の内容に関連して、さらに担当者が関心を持った部分を聞き出すことが出来た。
- 標準モデルへの展開
 - 標準 (PRESCRIPTIVE) モデル作成の基盤として活用

標準用 (to be) モデルへの展開手順

「あるべき姿」のモデルを作成するために
モデル化要素の選択やツールの選択など
モデル化方針の検討をモデル作成以前に実施。

実際のプロセスに適用することが意図されるため、通常、現状分析のモデルよりも:

- 表記形式を厳密にする必要がある
- モデル内での一貫性・整合性をより厳密に維持する必要がある



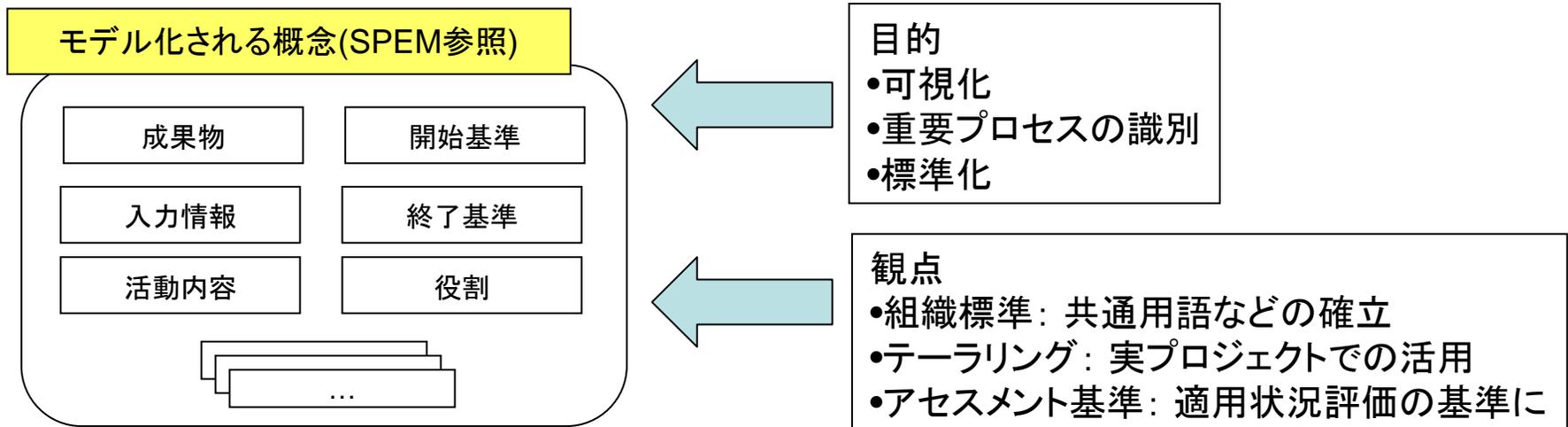
モデル化方針の検討(要素の選択)(1)

SPEM (Software Process Engineering Model) :

<http://www.omg.org/technology/documents/formal/spem.htm>

RUP, ISO-12207などプロセスを記述するレベルに対して、プロセス・モデルを表現するためのテンプレートとして、さらに1レベル抽象度を高めたモデル。

プロセス・モデルを表現するための要素として、Role, WorkProduct, Activity, Phase, Dependenciesなどを用意している。



SPEMで定義されている要素を基にし、プロセス・モデルの「目的」「観点」から、モデル化の対象とする概念の選択を行った。

モデル化方針の検討(要素の選択)(2)

SPEM 要素	モデルの観点
WorkProduct	すべて
WorkDefinition	すべて
Activity	すべて
Step	組織標準プロセス プロジェクト・テーラリング
ProcessRole	組織標準プロセス プロジェクト・テーラリング
Lifecycle	プロジェクト・テーラリング
Phase	プロジェクト・テーラリング
Iteration	プロジェクト・テーラリング
Precondition, Goal	プロジェクト・テーラリング

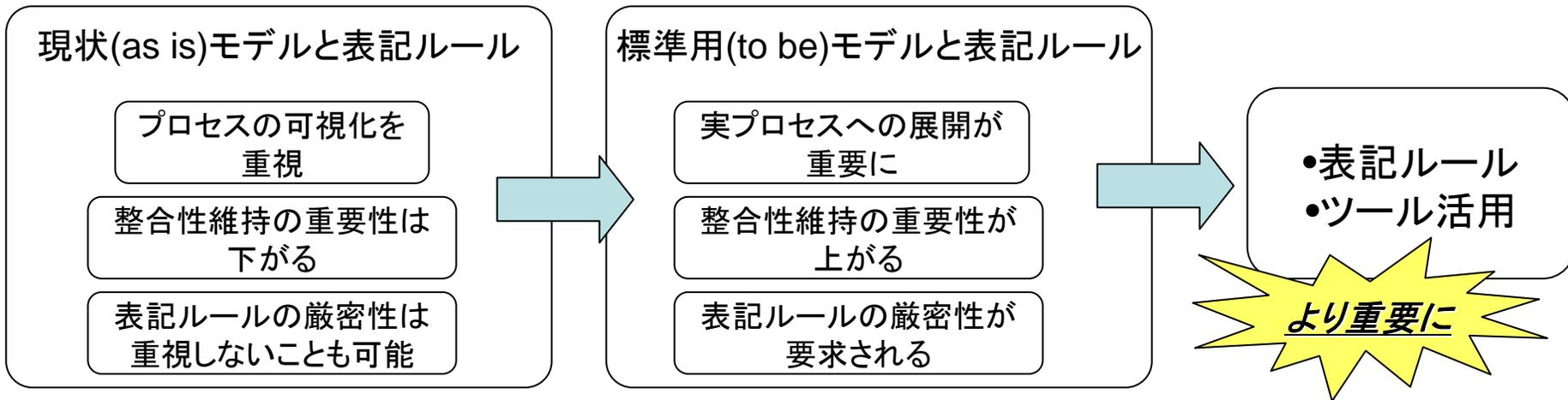
「観点」と、プロセス・モデルに必要な要素とのマッピングを実施して、以下の要素を選択(一部)

- WorkProduct
- WorkDefinition
- Activity
- Step
- ProcessRole

など

必要と判断したSPEM要素(一部)

モデル化方針の検討(表記方法／ツール)



•表記の選択:

データフロー図をベースに、各プロセスの説明文で要求事項を記述する方法を選択した。

- 現状分析モデルにある各活動に対して入出力情報を分析することで展開できたため、作業を進めやすかった。
- 「標準プロセス(あるべき姿)」として、入出力情報(とくに出力情報)の検討は重要と考えられた。

•ツールの選択:

プロトタイプのプロセス・モデル表記ツールが利用可能であったため、選択した。

- プロセス・モデルの作成を目的としたツールで、かつ簡単に試行できるものが、他にあまり見当たらなかった。
- SPEMの概念に(部分的に)対応していた。
- 作成したモデルをhtml形式のガイドとして出力する機能があり、関係者への説明や意見の獲得に有効と考えられた。

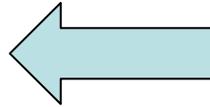
源泉情報の獲得とモデルの作成

源泉情報の獲得

- プロジェクト関係者とのインタビューを実施
- ソフトウェア開発者(委託先)とのインタビュー実施
- 既存の標準文書／ガイドライン文書



数回繰り返す



モデルの作成／更新

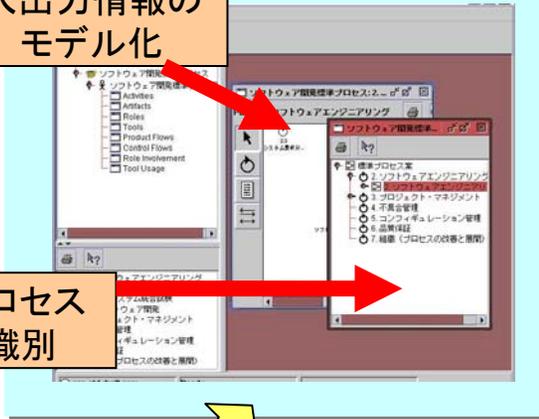
- 既存の標準文書／ガイドライン文書、過去のプロジェクト成果物による確認、更新
- インタビュー実施者へ問い合わせ

ツールの活用

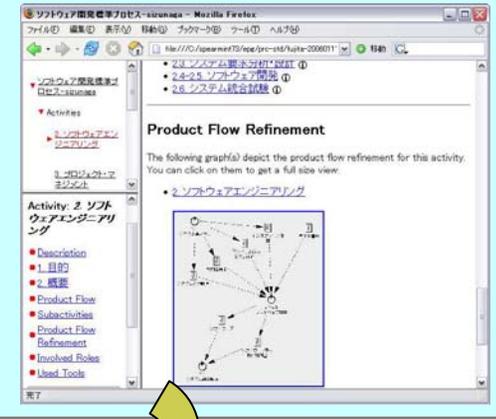
ツールによるモデル化

入出力情報のモデル化

プロセス識別



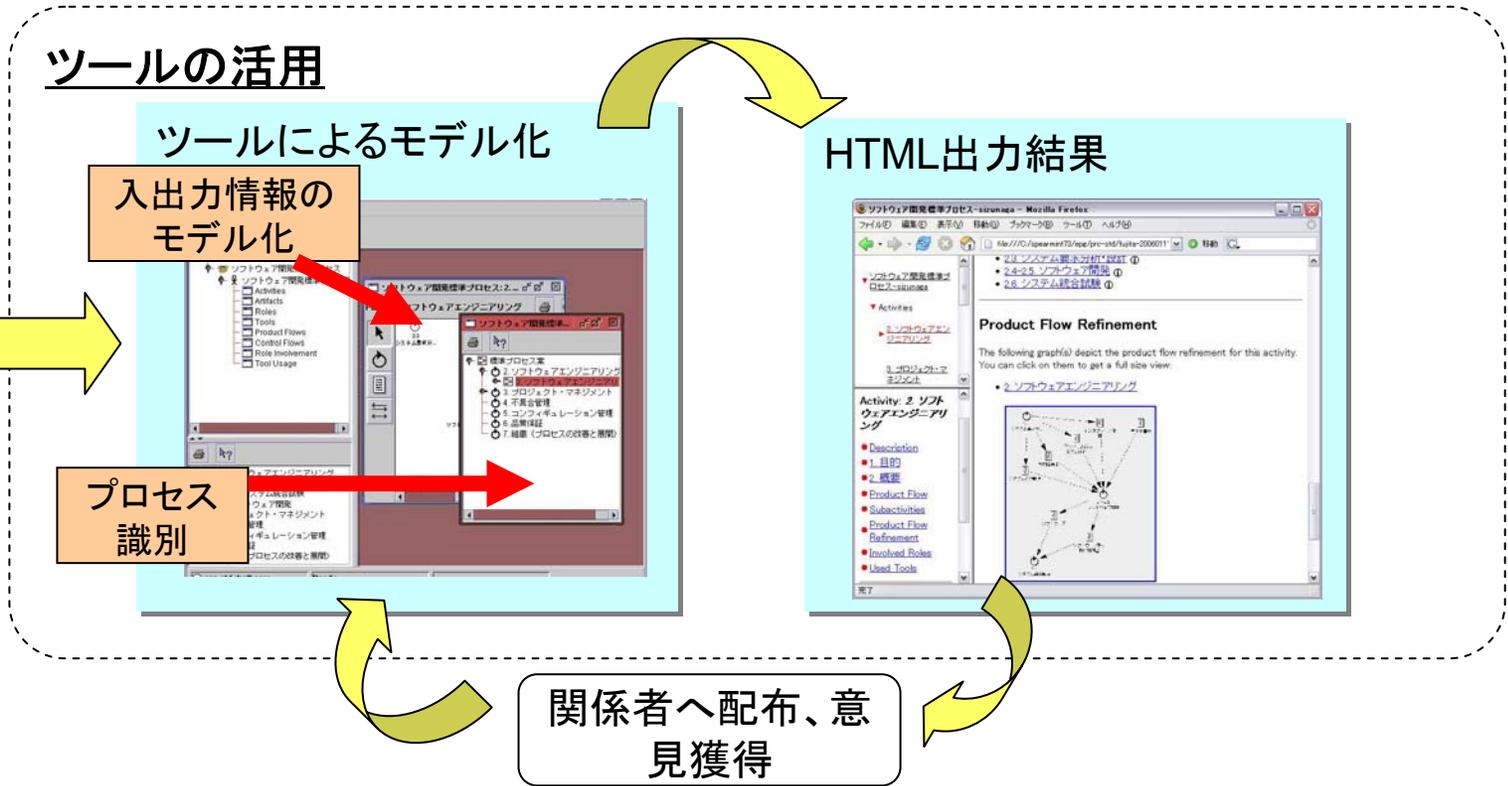
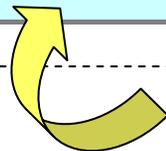
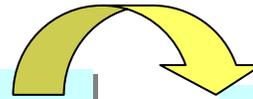
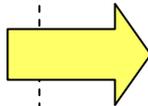
HTML出力結果



関係者へ配布、意見獲得



源泉情報



標準用(to be)モデルの成果

標準プロセス用のプロセス・モデルを作成した成果:

- 標準プロセスとして必要な要素をモデルに追加
 - 目的／入出力／実施内容を主要なプロセス毎に識別
 - プロセス間インタフェースの明確化
 - 入出力情報を明らかにしたことにより、プロセス間のインタフェースを明らかに
 - プロセス・モデル内の整合性確保
 - ツールにより、モデル作成時にモデルのルールを適用
 - 不整合となる可能性の高い部分の識別
 - 入力のみ存在し何も出力していないプロセス
 - 違う階層のプロセスで、同じ成果物を使っている部分
- など、構造上不整合が起き易くなっている可能性がある。

発表概要(3)

プロセス・モデルの概要・意味・定義など

プロセス・モデル作成事例

プロセス・モデル活用を試みた所感

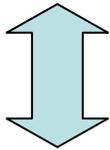
プロセス・モデル作成活動の所感(教訓)(1)

- 視覚的な表現を使ってプロセスを表現することは、プロセスに関する**意見交換・コミュニケーションに効果的**
- データフロー図/UMLなど、既存の**システム分析に関する知識・技術**が、プロセス・モデルの作成でも役立つ
 - プロセスの「入力／出力情報」は、プロセスの検討を行うときに手をつけ易く、データフロー図の知識もそのまま生かしやすい
- プロジェクト担当者など、実際にプロセスを実施している人から**情報を収集するには工夫が必要**
 - プロセス実施者の中では「当然の知識」のように暗黙的に前提としている部分が存在する
 - インタビュー／アンケートなどを行う場合、質問に対して直接的な回答は返ってくるが、暗黙的に前提としている部分は返ってこない
 -
 - モデルを見せるだけでなく、適切に説明や質問をしていく必要がある。
 - CMMやISOなどの用語ではなく、現場の用語・文書名を使うと、効果が高くなる。

プロセス・モデル作成活動の所感(教訓)(2)

- ツールの支援が重要

- 現状分析モデルでは PowerPoint などのドローイング・ソフトも選択肢に
→ モデルの整合性などが強く求められず、意見交換などに利用することを重視する場合は対応可能



- 標準用モデルではツールが強く望まれる
→ 整合性の維持など、ツールの支援がないと非常に難しい

モデル化する範囲が広がっていくと:

- 図が複雑になる
 - 全体としての整合性を維持することが難しくなる
 - ハイパー・リンクのようなナビゲーション機能が重要になる
- プロセス実施者からの**フィードバックを効率よく得られるか**が、作業時間やモデルの品質に大きく影響する

参考文献・資料

[Rolland1998]

“A Comprehensive View of Process Engineering”

<http://sunsite.informatik.rwth-aachen.de/CREWS/reports98.htm>

<ftp://sunsite.informatik.rwth-aachen.de/pub/CREWS/CREWS-98-18.pdf>

<http://citeseer.ist.psu.edu/rolland98comprehensive.html>

[Curtis,Kellner,Over1992]

“Process Modeling”

<http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=130998>

[Osterweil1987]

“Software Processes Are Software Too”

L.Osterweil, Proceedings of the 9th International Conference on Software Engineering, 1987, pp. 2-13.