

ソフトウェア開発システムのインテリジェント化に関する一考察

6R-4

武内 惇 藤本 洋  
日本大学工学部

1. はじめに

ソフトウェアの開発や開発管理を支援する技法やツールの開発が進み、実務への適用が図られているが、大規模なシステムの開発では人海戦術に頼るところが多い。本稿では、これらの技法やツールの効果的な適用を促進するための開発システムのインテリジェント化について検討したので報告する。従来取り上げられることの少なかった、「文書の推敲」の重要性にも言及する。

2. ソフトウェア開発のインテリジェンス

(1) インテリジェンス

人間は対象物や事象の「望ましい姿（モデル）」を所有しており、モデルと現実の姿との比較結果を基に対象物や事象を認識し、それ等に関する動作や操作を決定し、実行する。ここでは、モデルと比較の方法は対になっていることから、比較の方法も含めて「モデル」と呼ぶこととし、このモデルをインテリジェンスと考える。即ちインテリジェンスの高い人は、良いモデルをたくさん持っている人のことであると考えられる。

(2) ソフトウェア開発過程と成果物の特性

ソフトウェア開発の過程は、目標とするシステム化要件を要求仕様書に表現し、以後設計仕様やプログラムコードに表現を順次変換するものである。この過程で作成されるもの（成果物）は、思考結果の表現（表現物）である。この表現物が正しいものかどうか、良いものかどうかは、人間が読み解釈し判断しなければならない。またこの過程は、要求者や開発チームの特性、開発環境の変化に影響されやすく、常に監視し制御しなければならない。

(3) ソフトウェア開発に求められるモデル

ソフトウェア開発過程と成果物の特性から、ソフトウェア開発に求められるモデルを表1の3つに分類する。

モデル1は、開発や開発管理の作業そのものを実行する時に必要とされるもので、定式化、定型化された静的なモデルであり、書籍や技術講習会、試行実施などを通して構築されるものである。ソフトウェア工学はこれらのモデルの構築に大きく貢献している。

モデル2は、規模の大きなシステムの開発のリーダーや管理者には必須のもので、定式化が難しく、非定型な動的なモデルであり、「何かおかしい」

表1 モデル分類

分類	内容
モデル1	・技法やツール、手順、事例の機能や能力、内容、使用法 ・適用対象の特性によりどれを適用すべきかの判断に使用するもの
モデル2	・開発過程の進行状況から、問題を予測し、検出し対応策を判断するためのもの
モデル3	・思考結果である表現が、解釈系により異なった内容に解釈されないことを判断するためのもの

と感じる人間の「勘」が含まれ、実務経験を積むことによって構築される。

モデル3は表現された内容が正しいものかどうかを判断する前提となる適正な表現形態を実現（表現形態の推敲）するために必要とされるもので、表現法に関する規約から構成されるものである。過去に作成された「曖昧性のない、理解し易い表現」の事例から抽出され構築されるものである。

(4) モデル3と推敲の重要性

開発現場では、上流工程やコード化工程のレビューにおいて、表現に曖昧な所や理解し難い所があっても、レビュー参加者は独自の判断で表現内容を解釈する。このため表現内容の不具合が見逃され、最終的には試験の工程段階での試験により成果物の正当性が確認されている。この結果、作業の手もどりが発生したり、試験漏れで運用開始後、重大事故を発生することがある。

開発現場では、記述標準や規約に従って仕様表現することになっているが、適用が徹底されているとは言えない。

また、以下の理由により表現形態の推敲もなされていない。

①作成者は自分の考えをレビュー参加者に伝えるためではなく、自分が読んで分かる形式に表現しておけば良いと考える傾向があること。

②客観的に表現形態の不具合を指摘する体系的な手法やツールが揃っていないこと

前者については、レビューを、作成者が表現形態を推敲する段階と、推敲された結果を用いて関係者が、表現内容（表現が意味する内容）の正当性を検証する段階の2つに分け、推敲作業を独立する。後者についてはモデル3を基盤とした推敲の手法とツールの整備が必要である。

(5) モデル3構成の視点

仕様表現は文章や図形、表が組合わされた構造

A consideration about intelligence of software development system

Atushi Takeuchi Hiroshi Fujimoto  
DEPARTMENT OF COMPUTER SCIENCE,  
FACULTY OF ENGINEERING, NIHON UNIVERSITY  
KORIYAMA, FUKUSHIMA 963, JAPAN

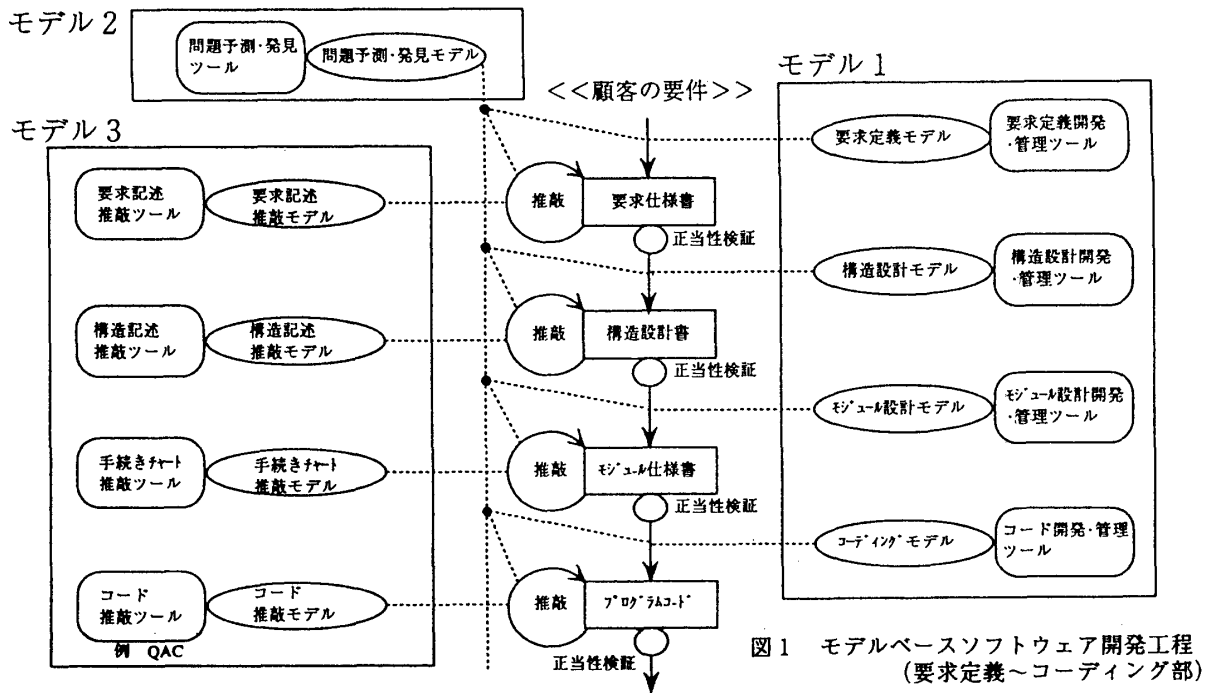


図1 モデルベースソフトウェア開発工程 (要求定義～コーディング部)

体であることから、次ぎの2つの面から表現形態を推敲する。

- ①個々の文章や図形，表の表現法の特徴
- ②文章や図形，表の組合せ形態の特徴

したがってモデル3は、①技法で決められた記述ルールや文法に従って表現されているが、解釈系が異なったとき、表現内容が異なって解釈される可能性があったり、理解し難い表現形態を検出するための判定基準を与えるもの(表現法モデル)と、②表現の規模や複雑さが大きなものは理解が難しくなるとの観点から、表現の規模や複雑さの適性を判定するためのもの(構造モデル)が必要となる。

3. モデルベースのソフトウェア開発過程の概念

ソフトウェア開発の作業はPDCA活動の繰返しであり、以下各ステップ毎に以下のモデルが使用される。

- ①P：開発や管理に適用する技法やツール，事例を選択 [モデル1]
- ②D：選択した技法やツール，事例を使用 [モデル1]  
状況の変化を監視 [モデル2]
- ③C：成果物の表現形態を推敲 [モデル3]  
成果物の表現内容の正当性を検証 [モデル1]  
状況の変化を監視 [モデル2]
- ④A：状況の変化に対する対策 [モデル2]  
モデルベースのソフトウェア開発過程を図1に示す。

4. 適用事例

モデルを構築し活用を支援する機能を備えた

開発システムの実現が望まれるが、C言語記述プログラムのコード化作業についてのモデル3とその支援ツール(QAC)の事例を紹介する。QACはC言語で記述されたプログラムのANCI規約の違反やコーディング規約違反、MaCabeの複雑性、Myersのインターバル等のマトリクスの値を測定するツールである(東陽テクニカ社)。QACが判定に使用している基準のうち、ANCI規約、コーディング規約がモデル3の表現法モデルに、マトリクスがモデル3の構造モデルに対応する。東陽テクニカ社より、以下の適用結果の事例が示されている。曖昧性のない、理解し易いコードを作成することの必要性、そのための推敲、モデル3の重要性が認識できる。

- ①ISO9001認証を受けた会社で作成したサンプルプログラム(180K行)でも18行に1つのコーディング規約違反が発見された<sup>[1]</sup>
- ②測定した関数の10%のものが200以上のパスカンの複雑性を有していた<sup>[1]</sup>
- ③運用中のプログラムで1000行当たり65個の危険なコーディングが発見された

5. おわりに

モデルベースのソフトウェア開発システムを確立するために、今後上流工程を対象としたモデル3の構築と支援システムの実現を進める。また、モデル1およびモデル2については、エージェントを中心とした実現法を検討する。

参考文献

- [1] Les Hatton: テストデバック時間を1/3にするテクノロジー-セシ資料, 東陽テクニカ, 12.1995