

Title	形式仕様記述を用いたテキストファイルの図式化の研究
Author(s)	手塚, 隆之
Citation	
Issue Date	2008-03
Type	Thesis or Dissertation
Text version	author
URL	<a href="http://hdl.handle.net/10119/4299">http://hdl.handle.net/10119/4299</a>
Rights	
Description	Supervisor:東条 敏 教授, 情報科学研究科, 修士

# 形式仕様記述を用いた テキストファイルの図式化の研究

手塚 隆之 (610703)

北陸先端科学技術大学院大学 情報科学研究科

2008年2月10日

キーワード: UML, ソフトウェア工学, オブジェクト指向, 日本語係り受け解析.

近年, 組み込み機器のソフトウェア開発ではプログラムステップ数が増加の一途をたどっている. 組み込み製品でも特に高機能化, ネットワーク化が必要な製品については, 特に顕著である. 代表的なものとして携帯電話, 自動車, MFP (マルチ・ファンクション・プリンタ) などがあるが, これらは一つの製品の中にあるプログラムステップ数は100万ステップを遥かに越え, 中には1000万ステップに達しようとしているものもある. このように開発規模が大規模化している中, 納期は縮小傾向にあり, しかしながら複数機種 of 並行開発も行っているという非常に厳しい状況にある.

現在のソフトウェア工学の仕様記述法としてもっとも汎用的なスタイルはオブジェクト指向の手法であると考えられる. 組み込みの世界でも同じ傾向を示しており, 従来の開発技術では対処できないこの状況に対応するため, オブジェクト指向開発技術が必要になる訳で, それはモデリング主体の開発を意味している. ところが技術者を育てるためには時間とコストが必要になる訳で, 特にオブジェクト指向開発可能な技術者不足が深刻化している.

このような背景から, ソフトウェア開発工程の上流部分の開発を支援でき, 生産性をアップさせることができるツール類の必要性を感じている. また開発要員不足からオブジェクト指向開発に不慣れな人員が開発の現場に投入されることも起こっており, 分析・設計品質を一定レベルまで確保する仕組みも必要になってきている. 特に分散開発, 海外リソースの活用などが盛んで, 共通フォーマットによるドキュメントの共有も重要になってきている. そこで, 本研究ではソフトウェアに対する要求文書 (すなわちテキスト文) から UML のクラス図とシーケンス図を導出するものを試作することを検討した.

本研究では入力として使うテキスト文として料理レシピの調理手順を用いた. 料理の作成手順はプログラムの実行手順と類似性が高く, UML で言うところのユースケースに近いものがあるからである. 料理レシピ文に書かれている文字列を解析し, その構造解析結果を形式言語の一種であるプログラミング言語に中間的に置き換えて, 図式表現であ

る UML へと変換を試みた。静的な構造をあらわすクラス図，動的な構造を表すシーケンス図を使って表示することを提案したが，料理のレシピ文を使った検証では，ある一定の成果を挙げられたと考える。その根拠としては，オブジェクト分析の経験のない分析者であっても，本アプリケーションを操作することで，オブジェクトモデリングの初期の分析作業が行えることが分かったことが挙げられる。オブジェクト指向に不慣れな人がクラス抽出すると危険な兆候を示すことが多い。例えば「クラス名が動詞句である」とか「メソッド名が説明的である」とかいったことである。しかしながら試作したアプリケーションを使用すればそのような心配はなくなるので，これは今回開発したアプリケーションの有用性と可能性を示すことができるものと考えられる。

本研究では 4 つのアイデアを導入している。

- メソッドに対して，テキスト文に現れた順番に番号を付けた変数名を引数とした。
- クラス名として認識した名詞に対し，連続で現れる名詞をクラス属性と定義した。
- 特殊な調理動作を示す補足説明をクラス図に反映させるための手段として Java ソースコードのメソッドコメントを利用した。
- UML モデリングツールをアプリケーション側からキーイベントによって操作する試みを行った。

一つ目のメソッド引数の連番については，シーケンス図生成の際に使用したが，過去の研究例にないシンプルな方法でシーケンス図を生成することができた。ただし，提案したアイデアであるテキスト文に現れた順番に番号を付けるやり方は，実行される手順を考慮した記述となっている料理レシピ文だからこそ使えるやり方とも言える。実際のソフトウェア開発の現場では，仕様を書く人間は思いつきで書いている面もあり，また仕様追加も頻繁におこなわれることからプログラムの実行手順を意識したドキュメントとなっているものは稀である。

二つ目のクラス属性については，特にアイデアが生かされてはいないが，強いて言うなら食材や調味料のグループを可視化するのに役立ったと言えるのではないかと考える。

三つ目のクラス図に特記事項を反映させるために，Java ソースプログラムのメソッドコメントを用いるやり方は，本研究で使用した JUDE 以外のモデリングツールにも使えるやり方であるか，検証しなければならない。広く使える仕組みであれば，ソフトウェア開発する側からみると大変便利な機能であるといえる。通常は複数のドキュメントを参照しなければならないことが，モデリングツール上に自動的に記載されていることになるからである。

四つ目のアイデアは研究の本質から外れるが，外部アプリケーションをキーイベントを使って操作するというやり方は，操作の自動化と誤操作防止に役立つもので，改めて認識することができた。