形式仕様記述言語を用いた要求仕様書の形式化と検証仕様書の獲得手法に関する研究

吹田有行 (710061)

情報科学研究科北陸先端科学技術大学院大学 情報科学研究科

2010 年 2 月 9 日

キーワード: VDM++ , OSEK/VDX , 形式化, 検証仕様書.

OSEK/VDX 仕様書からは様々な実装が作成される。様々な OSEK/VDX の実装が仕様書に準拠した動作をしているかを確認しなければならない。仕様書は、実装の動作をタスクや資源などの観点から資源とタスクなどの関係や制約を記述している。一方、実装は状態の変化による動作である。実装と仕様書は書かれている記述の観点に相違がある為準拠しているか確認するのは以下の点より困難である。

1. 仕様書は自然言語で書かれている為曖昧である
2. 仕様書の記述では実装の動作を直接比較できることが出来ない

1 の問題点は自然言語で書かれた仕様書の解釈は一意ではない点である。従って、曖昧である仕様書から実装が仕様に準拠しているか確認することは困難である。2 の問題点は、仕様書は実装の動作をある観点から資源とタスクなどの関係や制約について記述されており、実装と同じ観点から比べることができない。その問題点を解決する為に以下の要素を持つ仕様書が求められている。

1. 形式化された仕様書
2. 実装と同じ観点を持った仕様書

1 は、自然言語で書かれている為曖昧である仕様書の問題点に対する解決方法である。形式化された仕様書は曖昧さが無くなるので実装と比較元が明確になる。本研究では形式化の際 VDM++を使用する。 2 は、仕様書の記述では実装の動作を直接比較できることが出来ない問題点に対する解決方法である。実装と同じ観点を持つとは、実装と同じ振る舞いを持った動作することである。実装と同じ振る舞いを持った動作する仕様書ならば同じ観点で比較することが出来るようになる。この実装と同じ観点で記述された仕様書を検証仕様書と定義する。実装が仕様に準拠していることを確認する為に仕様書から検証仕様

書を獲得する事が求められている。本研究では仕様書から検証仕様書を獲得する為の方法を提案する。しかし、仕様書を検証仕様書に変換するのは以下の問題点があげられる

1. 情報が分散している
2. 齟齬が起きる可能性がある

１の問題点は仕様書にはサービスコールによるタスクの状態遷移や制約、資源とタスクの関係や制約関する情報が分散して記述されているため、仕様書を検証仕様書に変換するのは困難である。検証仕様書を作成する時、状態遷移や制約、資源とタスクの関係や制約を整理する方法が無いこと問題である。2 は仕様書と検証仕様書を変換する際に齟齬が出る可能性があるのは以下の２つのタイミングの時である。

- 仕様書を形式化する時
- 仕様書の様々な観点を組み合わせる時

齟齬が出る仕様書を形式化の例としてタスクの状態の形式化がある。VDM++で記述される関数は意味を厳密にする分、制約が厳しく自然言語の曖昧な表現の為齟齬がある変換を行ってしまう可能性がある。仕様書の様々な観点を組み合わせて振る舞いを作る。しかし仕様書の振る舞いを表現するのは膨大な仕様書の情報が必要であり漏れてしまう可能性がある。仕様書を検証仕様書に変換する問題点から以下の手順で仕様書から検証証書を獲得する。

1. 分散している情報をそのまま他の情報を参照しないで形式化を行う
2. 分散した情報から振る舞いに変換する。
3. 仕様書と作成した検証仕様書を確認

本研究では実際に OSEK/VDX の仕様書を事例に提案手法を適用した。手順１の目的は以下である。

- 仕様書の形式化
- 仕様書の情報に事前・事後条件をつける
- 対応関係表の作成

分散している情報を一度に形式化するのは困難である。本研究では分散した仕様書の情報をそのまま形式化する。これにより各情報を確認しながら仕様書を作成せずに分散したままで形式化することで、情報が分散している問題点を解決する。しかし、他の情報を参照無ければ形式化できない記述がある。その場合は仕様書を形式化を行わない。形式化した際に事前・事後条件の情報を明確にして整理する準備を行う。仕様書を形式化したの

で、仕様書とVDM記述のギャップを埋める為に対応関係表をつくる。対応関係表より自然言語からVDM記述への変換が明確になり、変換が本当に正当か議論すること可能になる。本研究ではこの手順で作成された仕様書を独立VDM記述と呼ぶ。独立ＶＤM記述は対応関係表が認められれば形式化された仕様書と物である。2の手順の目的は以下である。

- 形式化できない所を形式化
- 形式化された情報の整理
- 他の情報を統合

独立ＶＤM記述で他の情報を参照しなければ形式化できない箇所がある。他の章を参照しながら形式化を行う。この形式化は独立ＶＤM記述と同じ方法で形式化を行う。全ての情報の形式化が終わったので、他の形式化された情報を事前・事後条件によって整理する。事前・事後条件が同じ情報は同じ動作を違う観点からみて記述されている可能性が高い。そのような記述は検証仕様書をつくる上で密接な関係である可能性が高い。またある情報Aの事前条件と同じ事後条件を持つ情報Bがあるとする。これはBのある観点からみた動作が終了した後にAのある観点からみた動作が始まる可能性がある。本研究の事例ではOSEK/VDXのタスクの状態に注目して整理を行った。この様に事前事後条件に着目することで分散している仕様書の情報の整理を行う。最後に整理した情報から統合していく。この手順で出来た仕様書を統合ＶＤM記述と呼ぶ。本研究ではサービスコールに着目して統合ＶＤM記述を作成した。最後に仕様書と齟齬が無いか確認する。対応関係表が認められれば、仕様書と同じ独立ＶＤM記述と同じである。そこでVDMtoolのインタプリタを使用して齟齬があるかを確かめる。齟齬の確認の仕方として独立ＶＤM記述と統合ＶＤM記述に対応関係が認められる引数をそれぞれインタプリタを使用して実行する。実行した独立ＶＤM記述と統合ＶＤM記述の返り値に対応関係が認めらられれば独立ＶＤM記述の情報を統合ＶＤM記述は満たしている事がわかる。