社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－ＳＥ－160 (3)
2008－EMB－ 9 (3)
2008／6／19

COSMIC法による機能規模測定の信頼性評価

野中 誠[†1] 長野伸一[†2]
荒木盛次[†3] 菊本正紀[†4]

組込みシステム開発のソフトウェア要求プロセスを想定し，COSMIC法を用いたソフトウェア機能規模の測定実験を行った。2名の熟練測定者が同一のソフトウェア要求仕様に基づいて機能規模を測定した結果，機能規模のうち10～20%は異なるデータ移動を識別していた。その原因として，ユーザ機能要求の視点の違い，連携コンポーネントとのインタフェースの曖昧さ，状態遷移のガード条件の扱い，連携コンポーネント同士の制御手順の不明確さが挙げられた。これらの原因を要求プロセスの初期に解消することで，機能規模測定の信頼性向上に貢献すると考えられる。

A Reliability Evaluation of Functional Size Measurement by using the COSMIC Method

Makoto Nonaka,[†1] Shin'ichi Nagano,[†2] Moritsugu Araki[†3] and Masanori Kikumoto[†4]

An experiment was performed on measurement of the functional size of software by using the COSMIC method on the assumption of a software requirements process in embedded system development. Two skilled raters independently measured the functional size of a software system based on an identical software requirements specification, and identified 10 to 20% different data movements in the measured software functional size. Four major reasons of the difference have demonstrated through the analysis; the difference of viewpoint on functional user requirements, ambiguity of interfaces of related external components, consideration of guard conditions on state transitions, and ambiguity of procedures for controlling external components. Bridging these gaps at earlier stages of a software requirements process will increase the reliability of functional size measurement.

1. はじめに

組込みソフトウェア開発プロジェクトの工数見積り，開発計画，進捗管理などの基準として，COSMIC法☆[3]で測定されたソフトウェア機能規模の利用が期待されている[16]。ソフトウェア機能規模とは，ユーザ機能要求☆☆を定量化して得られる測定量であり[9]，その測定に必要な情報は，本質的に，ソフトウェア要求プロセス内☆☆☆ですべて与えられる，または定義される。したがって，要求プロセス終了時には，開発対象ソフトウェアの機能規模は一意な値に定まるはずである。

しかし実際には，ソフトウェアの機能規模測定(FSM: Functional Size Measurement)を実施した結果は，常に一意な値に定まるわけではない。同一のユーザ機能要求に対して，同一のFSM手法を適用した場合でも，測定者個人[1),12),13)]や組織における測定標準の違い[13)]によって，測定結果に差異が生じる。また，同一の測定者であっても，ソフトウェア要求仕様およびFSM手法の測定ルールの解釈によって一貫した測定結果が得られない場合もある。

このように，FSMには，真の値と測定結果が異なるという正確性の問題と，測定を行うたびに結果が異

†1 東洋大学 経営学部
Faculty of Business Administration, Toyo University
†2 日本電信電話 (株)
Nippon Telegraph and Telephone Corp.
†3 ソフトメトリックスジン (有)
Soft Metrics Jin Inc.
†4 日本ノーベル (株)
Japan Norvel Corp.

☆ The Common Software Measurement International Consortium Method

☆☆ ユーザニーズを満たすためにソフトウェアが果たさねばならないユーザの操作や手順[9]。品質要求や技術要求は含まれない。

☆☆☆ プロセスは時間的な区切りを意味しないことから[8]，ここでは，要求プロセスが開発プロジェクトの初期段階ですべて終了するということを意味していない。

なるという再現性の問題が存在する[11]。本稿では，正確性と再現性を総称して FSM 手法の信頼性と呼ぶ。真の値の推定値を得るために，複数人による測定結果の平均を用いることを推奨する提言もあるが[14]，本質的な解決からはほど遠い提言である。FSM 手法の信頼性を高めるために，測定結果に差異が生じる原因を明らかにし，これを取り除く努力をすべきである。

代表的な FSM 手法である IFPUG-FPA 法☆の信頼性が低下する原因として，開発経験や測定経験といった測定者のスキル要因が指摘されている[13]。また，測定ルールの曖昧性も信頼性低下の原因として指摘されている[12]。さらに，要求プロセスにおいて測定に必要な情報が得られなかった場合も，FSM 手法の信頼性は低下する。しかし，組込みシステム開発におけるソフトウェア要求プロセスにおいて，COSMIC 法の信頼性を低下させる原因は先行研究において明らかになっていない。

本研究の目的は，組込みシステム開発におけるソフトウェア要求プロセスにおいて，COSMIC 法の信頼性を低下させる原因を明らかにし，信頼性の低下を防ぐ方策を提言することである。この目的を達するために，著者らは，高齢者在宅監視システムという架空の組込みシステム開発を対象として，そのソフトウェア要求プロセスにおいて COSMIC 法を用いたソフトウェア機能規模の測定実験を行った。測定を行ったのは COSMIC 法の熟練者 2 名であり，先行研究にあった測定者のスキル要因による影響を排除する努力をしている。

本稿の構成は次の通りである。2 章では，本実験の理解に必要な前提知識と実験概要を述べる。3 章では，測定結果，差異が生じた部分とその原因，および信頼性低下を防ぐ方策について述べる。4 章で本研究全体に関する考察を述べ，5 章で結論を述べる。

2. 準　　備

本章では，まず COSMIC 法の概要を紹介する。続いて，想定したソフトウェア要求プロセス，測定対象システムの概要，および測定の準備手順について説明する。

2.1 COSMIC 法

COSMIC 法は，ユーザまたは連携コンポーネントとのデータグループの入力 (E)/出力 (X)，記憶領域とのデータグループの読み込み (R)/書き込み (W) という 4 種類のデータ移動 (**図 1**) に着目して機能規模を測定する FSM 手法である。COSMIC 法は，国際規格[10] としても採用されている。COSMIC 法では，ユーザ機能要求を機能プロセスの集合として表現し，各機能プロセスにおいて識別されたデータ移動の数を合計して機能規模を得る。本稿の執筆時点で最新の COSMIC 法である COSMIC method v3.0 では，機能規模の単位として CFP を用いる。

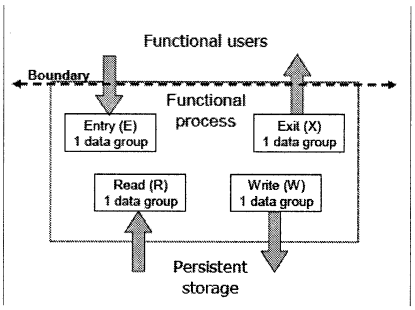

図 1　COSMIC 法における 4 種類のデータ移動[3)]

データグループとは，データ属性の集合であり，ユーザ機能要求の観点からその単位が識別される。また，機能プロセスとは，少なくとも 2 つのデータ移動を含む機能単位であり，トリガイベントによって実行される。ここで，単一のデータグループに対して，これを移動させる複数のデータ移動がそれぞれ異なる機能プロセスに含まれていれば，それぞれの機能規模を別個に測定することに留意する必要がある。例えば，記憶装置が保持するシステム設定情報というデータグループを，2 つの異なる機能プロセスが独立に読み込む場合は，それらのデータ移動は別々に測定する。

2.2 想定したソフトウェア要求プロセス

組込みシステム開発のソフトウェア要求プロセスは，組織，開発対象，方法論によって多種多様である。組込みソフトウェア開発プロセス標準が示されていたり[7]，一般的な要求プロセスとして要求の抽出・分析・仕様化・妥当性確認といったステップが示されていたりする[2]。また，非機能要求の明確化も重要である。しかし，本研究での関心事はソフトウェア機能規模であることから，機能性に焦点を絞った要求プロセスを想定する。ここでは，複雑さを回避するために，以下に示す比較的シンプルな要求プロセスを想定した。**図 2** に，要求プロセスのフローを図示する。

ステップ 1:　システムの機能概要を明確化する。ユーザ機能要求のうち，主に正常系の処理について，自

☆ International Function Point Users Group - Function Point Analysis Method

然言語で記述する。また，システムブロック図を作成し，ソフトウェアとその周辺のアクタ (ハードウェア構成要素も含む) との関係性を示す。

ステップ 2: ユースケース図を作成し，ソフトウェアが実現する機能をモデル化する。例外処理について，特別なドメイン知識がなくても挙げられるものは列挙し，その内容をユースケース記述に書く。

ステップ 3: 状態遷移表を作成し，ソフトウェアの状態と入力イベントのすべての組み合わせについて，ソフトウェアの振る舞いを定義する。作成された状態遷移表に基づいて，視覚化のために，状態遷移図を作成する。

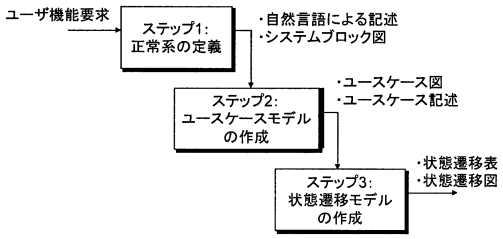

図 2 想定した要求プロセスのフロー

なお，一般的な要求プロセスでは，顧客ニーズを満たすための機能性について幅広く検討するステップが含まれる。このステップでは，機能性を高めたり，あるいは不必要な機能を減らすなどの検討がなされる。しかし，本研究では COSMIC 法の機能規模の信頼性に着目していることから，初期のユーザ機能要求はすでに固定化されている状態を想定した。要求プロセスの各ステップを経ることにより，固定化されているが文書化されていない要求が，段階的に仕様化されていくというプロセスを想定している。

2.3 測定対象のソフトウェア

測定対象の組込みソフトウェアとして，高齢者専用集合住宅の在宅監視システムに含まれる制御ソフトウェアを取り上げた。**図 3** にこのシステムのブロック図，**図 4** にユースケース図，**図 5** に状態遷移表を示す☆。このシステムは，高齢者の在室状況を，鍵の施錠センサー，マイコン内蔵水道メーターのセンサーによって監視するシステムである。高齢者が在室時に警備員呼び出しブザーを押下したり，水道が一定時間以上利用されない状態が続くと，制御ソフトウェアから通信機器にメッセージを送って警備会社へと連絡する。高齢者の在/不在の情報は，生活情報監視盤に表示される。生活援助員は，この情報を参照して，定められた業務手順に従って高齢者に連絡をとるなどして高齢者の安否を気遣う。

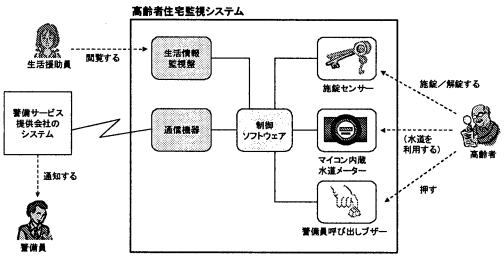

図 3 高齢者在宅監視システムのブロック図

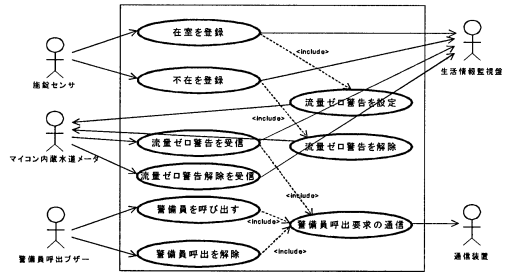

図 4 高齢者在宅監視システムのユースケース図

状態 / イベント	在室			不在
	正常	水道12時間未使用	警備員呼出中	
中から施錠	-	-	-	在室(正常)
中から解錠	-	-	-	N/A
外から施錠	不在	不在	不在	-
外から解錠	-	-	-	-
流量ゼロ警告	在室(水道未使用)	-	-	-
流量ゼロ警告解除	-	在室(正常)	-	N/A
呼び出しブザー押下	在室(警備員呼出中)	在室(警備員呼出中)	-	N/A
警備員呼び出し解除	-	-	在室(正常)	-

図 5 高齢者在宅監視システムの状態遷移表

このシステム仕様は，実際の事例を参考にして著者が創作したものである。元々，要求仕様レビューの練習問題として作成したこともあり，規模も小さく，複

☆ この仕様では妥当性の観点で問題があるが，妥当性評価は FSM の範囲外であるため，このままの仕様に対して測定する。

雑な処理のないシンプルなシステム仕様である☆。**表 1** に，各ステップで得られる仕様別の，規模メトリクスと測定値を示す。

表 1 測定対象ソフトウェアの規模

ステップ	メトリクス	測定値
1	仕様書スライド枚数	6
2	ユースケース数	6
	include されるユースケース数	3
	アクタ数	5
3	状態数	4
	入力イベント数	8

2.4 準備手順

想定したソフトウェア要求プロセスは，筆者らが定義した。測定対象ソフトウェアの仕様は，著者らのうちで測定に関与しない 1 名が定義した。ただし，ステップ 2 のユースケースモデルは，測定を行った 1 名が作成し，これを共有した。これらのプロセスおよび仕様に従って，著者のうち 2 名が測定者となり，このソフトウェアの機能規模を COSMIC 法を用いて測定した。

3. 測定結果

3.1 測定結果の概要

表 2 に，要求プロセスのステップ別に，2 名の測定者が共通して識別したデータ移動の数と，各測定者が独自に識別したデータ移動の数を示す。各測定者が各ステップで測定した機能規模は，共通部分と，各測定者が固有に識別したデータ移動の数を合計した値で得られる。例えば，ステップ 3 では両測定者とも 30CFP という測定結果を得たが，そのうちの 6CFP はそれぞれ異なるデータ移動を識別していたことを表している。

表 2 測定結果：共通部分と，測定者固有のデータ移動識別数

ステップ	共通部分	測定者 A	測定者 B
1	13	12	0
2	24	3	3
3	24	6	6

ステップ 2 において，測定された機能規模に対する，測定者間で差異があった機能規模の比率は 11%であった。ステップ 3 において，同比率は 20%であった。測定対象ソフトウェアの機能規模は小さいが，それにも関わらず，10〜20%の誤差が生じている。なお，ステップ 1 の差異は他とは状況が異なるので，ここでは評価しない。

3.2 差異の詳細：ステップ 1

ステップ 1 での測定結果は，測定者 A は 25CFP，測定者 B は 13CFP であった。ステップ 1 で共通して測定された機能規模は，ステップ 3 での共通して測定された機能規模の約 54%であった。

表 2 に示した通り，両者の測定結果が大きく異なっている。2.2 節で述べた通り，想定した要求プロセスでは，ステップ 1 では正常系のユーザ機能要求を自然言語で書いた仕様を与えた。しかし，測定者 A は経験的知識により，例外処理に伴って生じるであろうデータ移動を補って測定していた。このように，測定者 A は与えられた要求仕様を純粋に測定したわけではなく，機能規模の見積りを行っていた。一方，測定者 B は，ステップ 1 で与えられた情報にのみ基づいて，正常系のデータ移動のみを測定した。ステップ 1 において大きな差異が生じた理由はこのためである。

一方で，測定者 B が識別した正常系のデータ移動は，測定者 A もすべて等しく識別していた。このことから，本例題システムの正常系データ移動については，測定者によって等しく識別されていた。

このように，ステップ 1 における正常系処理を中心とした概要レベルの記述では，与えられた情報のみから測定可能な部分においては差異が生じなかった。

3.3 差異の詳細：ステップ 2

ステップ 2 では，両測定者とも測定結果は 27CFP であった。しかし，そのうちの 3 つのデータ移動については，測定者によって認識が異なっていた。この認識の違いは，いずれも，測定対象ソフトウェアと連携コンポーネント間での通信プロトコルの曖昧さに起因している。曖昧さのために通信プロトコルの解釈に差異が生じ，異なるデータ移動が識別された。

測定者 A は，警備会社への通信用機器に対するデータ移動「警備員呼び出し要求の送信」が実施されるにあたり，これに先だって，呼び出し要求を発している高齢者を示すユーザ ID を通信機器に出力するプロトコルであると認識した。そのため，「警備員呼び出し要求の送信」で移動されるデータグループと，ユーザ ID に関するデータグループを分けて識別した。

一方，測定者 B は，通信機器に対するデータ移動「警備員呼び出し要求の出力」の実施に先だって，まず通信機器の状態を読み込み，通信可能であるかどうかを調べた上で警備員呼び出し要求を出力するというプロトコルであると認識した。この読み込み処理は，

☆ 本システムの仕様の概略は web で公開している。`http://www.se.mng.toyo.ac.jp/jp/public/zaitaku_v1.pdf`

測定者 A が識別していないデータ移動である。また，警備員呼び出し要求とユーザ ID は，通信メッセージという 1 つの論理的なデータグループに含まれると考え，測定者 A のように出力処理を分けなかった。

COSMIC 法では，データグループをユーザ機能要求の観点から，意味のある粒度で切り分ける。通信システムにおけるユーザ機能要求を考えれば，一度の通信で送受信されるデータの集合を，意味のあるデータグループの粒度として捉えることは妥当のようにも思える。一方，測定者 A は，アプリケーション領域の観点から意味のある粒度でデータグループを識別している。このように，ユーザ機能要求をどの視点で捉えたのかによって，データグループの認識に差異が生じた。

3.4 差異の詳細：ステップ 3

ステップ 3 では，両測定者とも測定結果の合計は 30CFP であったが，そのうちの 6CFP のデータ移動については認識が異なっていた。ただし，そのうち 3CFP については，ステップ 2 での認識のズレがステップ 3 に持ち込まれたものであるため，ステップ 3 で新たに生じた認識のズレは 3 つのデータ移動であった。認識のズレは，(1) 測定対象ソフトウェアと連携コンポーネントとのインタフェースの曖昧さ，(2) 状態遷移のガード条件の解釈，(3) 連携コンポーネント同士の制御方法に起因していた。

まず，(1) 連携コンポーネントとのインタフェースについて説明する。高齢者の状態が在室から不在に遷移させる際に，生活情報監視盤に「水道未使用」が表示されていたとする。このとき，生活情報監視盤からデータを読み込む必要があるのか，あるいは，水道未使用かどうかという情報は測定対象ソフトウェアの内部で保持しておき，生活情報監視盤からはデータを読み込むことなしに更新情報を送るのか，という違いである。測定者 A は読み込み不要と考えたが，測定者 B は読み込みが必要と考え，データ移動の認識に差が生じた。

次に，(2) 状態遷移のガード条件の解釈について説明する。このシステムは，呼出ブザー押下時および水道未使用警告が発せられたときに警備員を呼び出す。仕様では，**図 6** に示したように，水道未使用状態において水道を利用したとしても，警備員呼び出し中であれば正常状態へと遷移せずに警備員呼び出し中へと遷移する。ここで，ガード条件で示された「警備員呼び出し中」を，データの読み込みとして表現するかどうかで認識が異なった。測定者 A は，測定対象ソフトウェアの内部で保有する情報とみなし，このデータ移動を識別しなかった。一方，測定者 B は，これを連携コンポーネントから読み込むべき情報として識別した。

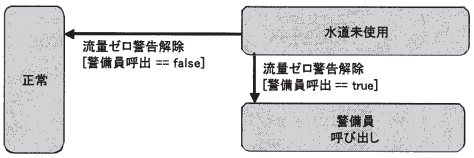

図 6 「水道未使用」と「警備員呼び出し」の状態遷移

最後に，(3) 連携コンポーネント同士の制御方法について説明する。マイコン内蔵水道メーターと生活情報監視盤という 2 つのコンポーネントについて，先にどちらのコンポーネントからどの情報を取得するのかという制御手順の認識の差が生じた。与えた仕様には，各コンポーネントとソフトウェアとのプロトコル仕様は記述してあったが，連携するコンポーネント同士の制御手順は書かれていなかった。この情報の欠落のために，測定結果が異なった。

4. 考　　察

以下に，前章で示した測定者間の認識の差異について，その原因や対策について考察する。

4.1 ユーザ機能要求視点の違い

ステップ 2 では，ユーザ機能要求視点を通信システムのレベルとするか，アプリケーション領域のレベルとするのかによって，データグループの捉え方に差異が生じた。いずれの視点も，分析者同士で共通認識がなされていればよい問題であると考えることはできる。ただし，筆者らは，アプリケーション領域の視点で統一すべきと考える。なぜなら，通信システムの視点で，データグループの粒度を通信データの構造に基づいて決めるのは，技術的要求に基づいた FSM になる可能性がある。FSM に関する国際規格では，技術的要求や品質要求から独立した，機能性に着目して FSM を実施することを求めている[9]。そのため，アプリケーション領域の観点で測定することが望ましいと考える。

ただし，アプリケーション領域の観点で FSM を実施するとなると，通信データなどに規程のデータ構造に含まれる要素を，アプリケーション領域の観点で捉え直す作業が必要になる。本稿の主旨とは異なる問題指摘だが，この変換は自動化が難しく，FSM 適用のハードルを高めることになるかもしれない。

4.2 連携コンポーネントとのインタフェース

ステップ 3 で生じた測定差異の一つが，測定対象ソフトウェアと連携する外部コンポーネントとのインタ

フェース仕様の曖昧さである。これは，あらかじめ明確に定義しておけば回避できる問題である。一般的に言われていることではあるが，インタフェース仕様を早い段階で確定することは，要求の明確化に結びつくとともに，機能規模の測定値がばらつく要因の除去にも貢献する。

4.3 状態遷移のガード条件の解釈

ステップ3で生じた測定差異の一つである。これは，例えばガード条件に指定された情報はデータ移動として測定するなど，機械的な測定ルールを設けることはできない。なぜなら，ガード条件に指定された情報が測定対象ソフトウェアの内部と外部のどちらに存在するのかは，測定対象ソフトウェアのスコープをどのように捉えるかによって異なるためである。

とはいえ，この問題も前項と同様に，早い段階から明確に定義しておけば回避できることである。仕様に含まれる情報については，測定対象ソフトウェアの内部と外部のどちらに置かれるのかを，早期に明確化することが必要である。

4.4 連携コンポーネント同士の制御

ステップ3で生じた測定差異の一つである。組込みソフトウェア開発に限ったことではないが，組込みソフトウェアを通じてさまざまな機器を相互に連携させて，統合システムとしての機能性を実現する場合が多い。ここで問題となるのが，統合システムとして見たときに，連携コンポーネント同士の協調のさせ方について，誰が意思決定できるかということである。この意思決定には，統合システムとして見たときのアプリケーション領域と連携コンポーネントに関する正しい知識が必要である。統合システムに関するアプリケーション領域の知識は，必ずしも顧客が保有しているわけではない。また，統合システムとして正しい振る舞いとなるために，複数の利害関係者から情報を得てシステム設計することが求められる。このような詰めを早期に行うことが，FSM の信頼性の改善に結びつくといえる。

以上に挙げてきたことは，要求定義の分野でこれまで指摘されてきたことに共通する。要求の精度を高めることで，FSM の信頼性を向上させることができるが，特に以上の点を早期に明確化することで，FSM の測定値を早く収束させられる。

4.5 include されるユースケースの扱い

本稿では，ステップ2においてユースケース図から機能規模を測定した。測定結果に差異は影響しなかったものの，複数のユースケースから同一のユースケースが include されている場合に，include 先ユースケースの扱いについて測定者間で議論となった。具体的には，include 元のユースケースを機能プロセスとして捉えた場合に，include 先ユースケースに含まれるデータ移動を，include 元のユースケース別に重複測定すべきか，一度測定されたデータ移動はそれ以降は測定対象から外すべきかということである。結論として，COSMIC 法の考え方に従い，重複測定することにしたが，明確な指針がなかったため判断に迷った。

この問題は，測定対象ソフトウェアを複数レイヤで考えた場合には，より複雑になる可能性がある。COSMIC 法では，測定対象ソフトウェアの内部にレイヤの考え方を導入し，レイヤ別に FSM を実施する方法を提供している。このとき，include 元ユースケースと include 先ユースケースが同一レイヤに含まれれば，ユースケース間のデータ移動は考慮しなくてよい。しかし，別レイヤになった場合，ユースケース間のデータ移動は識別すべき対象になる。このとき，複数の include 元ユースケースと一つの include 先ユースケースとの関係において，重複測定すべきかどうかは，現時点では明確な指針を筆者らは持っていない。具体的な事例を通じて，妥当な方法を探る必要がある。

5. 関 連 研 究

FSM 手法の信頼性や，COSMIC 法に関する議論は，これまでにもいくつか研究が報告されている。以下に，それらのうちで本研究と関連のあるものを紹介する。

Kemerer らは，FPA 法[4] の測定結果にばらつきが生じる原因を分析し，バックアップファイル，メニュー，外部ファイルの扱い方がばらつきの主要な源泉であることを示している[12]。Low らは，FPA 法を用いて，同一ソフトウェア仕様を複数の組織，複数の分析者に測定させた結果を示している[13]。この文献によると，ファンクションポイントの組織平均値が 26.0 から 85.5 までばらついており，最大では 159 という測定値を得た分析者もいた。また，分析者を熟練レベル別に分けると，熟練者グループと非熟練者グループでは平均と標準偏差が異なり，非熟練者グループはばらつきが大きいことが報告されている。このように，FSM は個人の熟練度にも大きく依存するため，測定ルールを明確化し，測定マニュアルや測定ツールを整備することが求められる。

COSMIC 法に関して，特定のアプリケーション領域に対して測定ルールを明確化する研究が行われている。野中らは，対話型ソフトウェアの画面仕様書を対象に，COSMIC 法に基づいた特定の測定ルールを設

けることで，測定の支援を行っている[15]。画面仕様とそれに対応して作成されたソースコード行数との関係を分析した結果，外れ値などを除けば高い相関にあることを示している。このような測定ルールの詳細化は，FSM 手法の信頼性を高めることに貢献するが，自動化の支援については言及しておらず，以前として人的要因による測定誤りが生じる可能性がある。

一方で，COSMIC 法に基づいた自動計測支援や測定ツールに関する研究も行われている。Diab らは，ROOM を用いた COSMIC 法の形式化に関する研究[5]や，Rational Rose RealTime のモデルから自動的に COSMIC 法の機能規模を測定する研究[6] などを行っている。こうした自動計測支援は有用だが，特定の環境や，特定の仕様記述に依存してしまう。また，4.3 節で示した通り，状態遷移図のガード条件に指示された情報が測定対象レイヤの外部または内部のどちらにあるのかによって測定方法が異なる。自動計測のためにはこのような情報を付記する必要があり，その手間が増えることによって自動計測のメリットが弱まる可能性もある。

冒頭にも述べたように，COSMIC 法を組込みソフトウェア開発に適用した場合に，どのような原因で測定結果に差異が生じるのかは，筆者らの知る限り十分な研究が実施されているとは言い難い。本稿では 2 名という少ない測定者による測定試行の結果を報告したが，ここで得られた知見をもとに，機能規模測定の信頼性低下を招く要因を分析し，信頼性を向上させるための提言に結びつけたいと考えている。

6. おわりに

組込みソフトウェア開発における要求プロセスにおいて，高齢者在宅監視システムという仮想システムを対象に，2 名の測定者が COSMIC 法により機能規模を測定する実験を行った。その結果，仕様の認識の違いにより，測定結果が 10〜20%異なるという結果が得られた。測定結果に差異が生じた原因は，主に，ユーザ機能要求の視点の違い，連携コンポーネントとのインタフェースの曖昧さ，状態遷移のガード条件の扱い，連携コンポーネント同士の制御手順の不明確さであった。これらの多くは要求プロセスにおいて早期に明確化すべき課題ではあるが，言い方を変えれば，これらの問題を早期に解決できるように要求プロセスを進めることで，機能規模の測定値が早い段階で安定すると考えられる。

本稿で報告した内容は，少人数の測定者による，小規模システムに対する試行的な取り組みであり，十分に一般化できる知見が得られたわけではない。今後の課題として，現実的な規模の，現実に生じる典型的な問題を含んだ組込みシステムを例題に，適切な実験計画に基づいた測定実験を実施することが挙げられる。

参考文献

1) Abrahao, S., Poels, G. and Pastor, O.: Assessing the reproducibility and accuracy of functional size measurement methods through experimentation, *Proceedings. 2004 International Symposium on Empirical Software Engineering*, pp. 189–198 (2004).
2) Abran, A., Moore, J. W., Bourque, P. and Dupuis, R.(eds.): *Guide to the Software Engineering Body of Knowledge (SWEBOK)*, IEEE Computer Society (2004).
3) Abran, A., Desharinais, J.-M., Oligny, S., St-Pierre, D. and Symons, C.: The COSMIC Functional Size Measurement Method version 3.0, Measurement Manual, `http://www.gelog.etsmtl.ca/cosmic-ffp/updates/COSMIC_method_V3.zip` (2007).
4) Albrecht, A. J. and Gaffney, J. E.: Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Transactions on Software Engineering*, Vol. SE-9, No. 6, pp. 639–648 (1983).
5) Diab, H., Frappier, M. and St. Denis, R.: Formalizing COSMIC-FFP using ROOM, *Computer Systems and Applications, ACS/IEEE International Conference on. 2001* (Frappier, M.(ed.)), pp. 312–318 (2001).
6) Diab, H., Koukane, F., Frappier, M. and St-Denis, R.: [mu]cROSE: automated measurement of COSMIC-FFP for Rational Rose RealTime, *Information and Software Technology*, Vol. 47, No. 3, pp. 151–166 (2005).
7) IPA/SEC: 組込みソフトウェア向け開発プロセスガイド (改訂版), 翔泳社 (2007).
8) ISO/IEC: ISO/IEC 12207:1995, Information technology - Software life cycle processes (JIS X 0160-1996: ソフトウェアライフサイクルプロセス) (1995).
9) ISO/IEC: ISO/IEC 14143-1:1998, Information technology - Software measurement - Functional size measurement - Part1: Definition of concepts (JIS X 0135-1:1999, ソフトウェア測定－機能規模測定－第 1 部：概念の定義) (1998).
10) ISO/IEC: ISO/IEC 19761:2003, Software engineering - COSMIC-FFP - A functional size measurement method (JIS X 0143:2006,

COSMIC-FFP 法－機能規模測定法) (2003).
11) ISO/IEC: ISO/IEC TR 14143-3:2003, Information technology - Software measurement - Functional size measurement - Part3: Verification of functional size measurement methods (2003).
12) Kemerer, C. F. and Porter, B. S.: Improving the reliability of function point measurement: an empirical study, *IEEE Transactions on Software Engineering*, Vol.18, No.11, pp.1011–1024 (1992).
13) Low, G. C. and Jeffery, D. R.: Function points in the estimation and evaluation of the software process, *IEEE Transactions on Software Engineering*, Vol. 16, No. 1, pp. 64–71 (1990).
14) Symons, C. R.: Function point analysis: difficulties and improvements, *IEEE Transactions on Software Engineering*, Vol. 14, No. 1, pp. 2–11 (1988).
15) 野中誠, 角頼章広, ブカーリイサム, 東基衛: 画面仕様書に基づく対話型ソフトウェアの複雑度重みつき機能規模の測定技法, 情報処理学会論文誌, Vol. 43, No. 12, pp. 3993–4004 (2002).
16) 眞瀬健一, 渡邉保夫, 長野伸一, 綿引隆一: 交換機の開発コストへの COSMIC-FFP 適用, 情報処理学会研究報告, Vol. 2002-SE-137, No. 35, pp. 1–7 (2002).