

やわらかいネットワークにおけるユーザ要求の獲得 に関する一考察

唐橋拓史[†] 郷健太郎[†] Goutam Chakraborty[†] 菅原研次^{††} 白鳥則郎[†]

[†] 東北大学電気通信研究所 / 情報科学研究科

^{††} 千葉工業大学情報工学科

あらまし 情報ネットワークの利用者増にともない、その上で提供されるサービスに対する利用者の要求も多種・多様化している。さらに利用者個人の要求も、時間の経過や作業形態の変化により、刻々と変化している。このような利用者要求の多種・多様化や変化に柔軟に対処することを目的に、やわらかいネットワークが研究されている。本稿では、このやわらかいネットワークにおける利用者要求獲得支援システムを提案する。このシステムは外部表現、内部表現、仕様表現、プログラム、要求獲得支援機構、変換機構から構成されている。ここで要求獲得支援機構における Evolution Mechanism(EM)が本質的に重要な役割を果たしている。このシステムの応用例としてHSC エディタの機能の追加について述べている。さらに仕様表現を用いて、やわらかさの定量的評価についても考察している。

A Study on Flexible Acquisition of User Requirements on Flexible Networks

T.Karahashi[†], K.Go[†], G.Chakraborty[†], K.Sugawara^{††}, and N.Shiratori[†]

[†] Research Institute of Electrical Communication
/ Graduate School of Information Sciences,
Tohoku University

^{††} Department of Computer Science,
Chiba Institute of Technology

Abstract Due to the increase in number of users in information networks, various user requirements are also increasing and the requirement for individual user is changing as the time passes. In order to deal with this situation, we have been researching on Flexible Network. In this article, we propose the User Requirement Acquisition Support System. The system is constructed with Requirement Description, System Specification, Program, Requirement Acquisition Support Mechanism, and Converter. The most important part in the system is the Evolution Mechanism in Requirement Acquisition Support Mechanism. We present function additions to HSC Editor as an application example of the system. Furthermore, we consider the quantitative evaluation of flexibility using System Specification.

1 はじめに

近年、情報ネットワークのユーザが増加するに従い、ネットワーク上で提供されるサービスに対してさまざまな要求が提示されるようになってきている。従来はこの要求を基にプログラマが新しいサービスを実装して要求を満たしてきたが、これでは多彩な要求に素早く答えることは困難である。そこでユーザ要求を即座にサービスとして実現する支援環境が必要となる。このような問題をやわらかいネットワークの研究の中で検討してきた。[1]-[4]。

本稿ではやわらかいシステムにおいて、文献[1]で導入した「Evolution Mechanism(EM)」に基づいた要求獲得システムを提案する。このシステムはユーザ要求表現をユーザフレンドリに支援するための自律的な機構から構成されている。

2 要求獲得システム

本稿で提案する要求獲得システムは、EM の表現を目的とした第1段階としての支援システムである。

要求獲得システムを図1のように構成する。

(1) 要求発生段階

ユーザがシステムに対する要求のイメージを持つ段階である。この時点ではユーザの要求はあいまいで、矛盾を含んでいることもある。

(2) 外部表現

ユーザ要求の入力表現。ウィンドウ、アイコンなどを用いてユーザにとって記述・理解しやすいような手順とインターフェースによって構成される。

(3) 要求獲得支援機構

ユーザ要求獲得の支援効果的に行なうための機構を提供し、EM 及び外部表現と内部表現の間の関連を管理する。

(4) 内部表現

ユーザ要求のシステム内部での中間表現。

(5) 変換機構1

ユーザの要求表現を仕様表現(有限状態機械)へと変換する。また、有限状態機械の検証で矛盾や情報の欠落が発見された場合、その内容を要求表現へとフィードバックする。

(6) 仕様表現

要求表現に基づいて、要求を満たす機能を有限状態機械として表現したもの。要求の誤りなどを検出することに利用する。

(7) 変換機構2

仕様表現からプログラムへの変換アルゴリズム。

(8) プログラム

ユーザの要求を実装したもの。

3 要求獲得支援機構

3.1 諸定義

(1) 機能: ユーザがシステムに対して求める機能。ユーザ要求は機能の集合で定義される。

(2) シーン: ユーザがアプリケーションを使っている場面をイメージとして表現した外部表現

(3) オブジェクト: ユーザがアプリケーションを使っている時に画面上に見えているウィンドウやアイコンなどを抽象的に表現した内部表現

(4) リレーション: 2つ以上のオブジェクトの間の関係を抽象的に表現した内部表現

(5) アクション: ユーザの操作、およびシステムの動作を抽象的に表現した内部表現

3.2 外部表現

外部表現の枠組を図2のように定義する。

オブジェクト、リレーション、アクションはアイコンの形で表現する。

また、それぞれのシーンはオブジェクトのアイコンを配置し、それらの関係をリレーションのアイコンを用いて定義することで構成する。同様にアクションを含むシーンはこれにアクションのアイコンを配置することで構成する。アクションを含むシーンはユーザがアプリケーションを操作した状況や、アプリケーションがなんらかの動作をしたときの状況を表現している。

次にユーザから要求を獲得するときの流れを説明する(図4)。

STEP1: 初期画面(図3-(1))

ユーザがこのシステムを起動すると初期画面が表示される。画面左下には Information ウィンドウが表示され、システムがユーザへ支持する内容などを表示する。ユーザが「要求入力」ボタンを押すと STEP2へ進む。

STEP2: 機能名の入力(図3-(2))

ポップアップウィンドウが表示され、ユーザに機能名の入力を促す。

STEP3: シーンの生成(図3-(3))

画面左には幾つかのオブジェクトを表わすアイコンが表示され、ユーザはこれを画面上に配置してシー

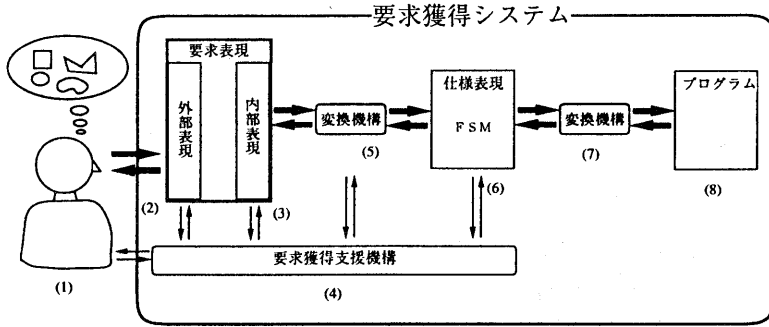


図 1: 要求獲得システムの構成

ンを作成する。

STEP4: シーン間の遷移の定義 (図 3-(4))

機能が使われる前と後の 2 つのシーンを選択する。このとき、選択した 2 つのシーンが中央のウィンドウに表示される。

STEP5: 操作の記述 (図 3-(5))

画面左には幾つかの操作を表わすアイコンが表示され、これを選択したあと、操作対象となるシーン上のアイコンを選択することで操作の内容を記述する。

STEP6: 動作の記述 (図 3-(6)) STEP5 で書いた操作に対するシステムの動作の内容を記述する。画面左には幾つかのシステムの動作を表わすアイコンが表示され、これを選択したあと、動作の対象となるシーン上のアイコンを選択することで動作の内容を記述する。

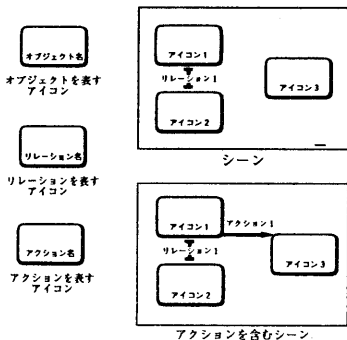


図 2: 外部表現の定義

3.3 要求獲得支援機構と Evolution Mechanism

要求獲得支援機構の構成を図 3 に示す。要求獲得支援機構は EM を中心として構成する。EM は要求の変化 ΔR (例えば機能の追加など) を検出、理解し、図 3 のように外部表現、内部表現、有限状態機械および変換情報の知識ベースを利用して、ユーザとインタラクティブに ΔR を実現する。

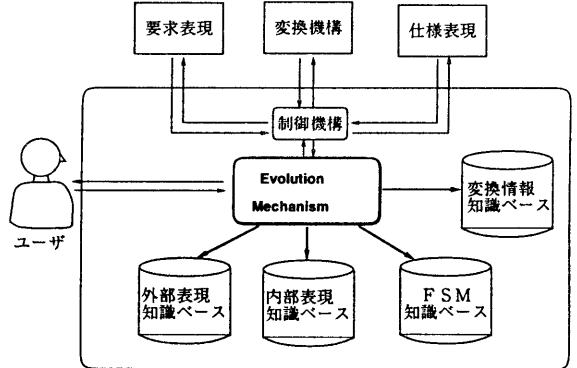


図 3: 要求獲得支援機構

4 HSC エディタへの応用

本節では 3 節で定義した要求獲得支援を HSC エディタへの応用例を示す。HSC[5] はシーケンス図に階層化とモジュール化の概念を導入した要求記述言語で、HSC エディタはその記述を支援するエディタである。紙面の都合上、機能拡張の例のみを示す。

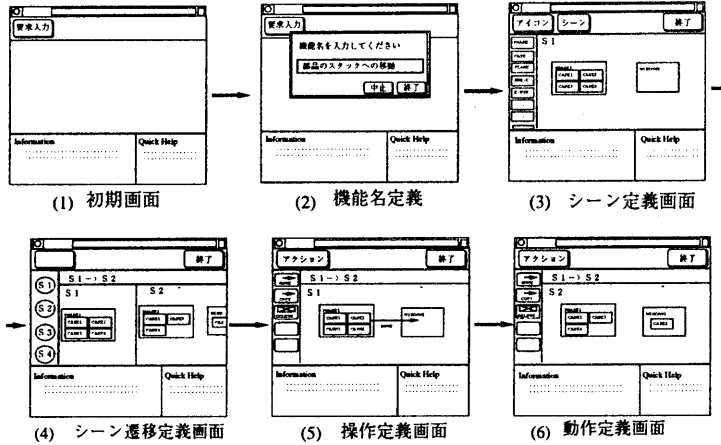


図 4: 要求獲得の流れ

4.1 要求例

ユーザが出す要求 R として次のような自然言語で表現された機能 $\{f_1, f_2\}$ を考える。

[要求 R (拡張機能)]

HSC 仕様の部品であるフェーズ、ケースを編集集中にその仕様の一部を退避できるような領域(仮にスタックと呼ぶ)を作りたい。

- f_1 : HSC 仕様の中にある部品を選んでスタックに持っていくとその部品がスタックに移動する。
- f_2 : スタックにある部品を選んで HSC 仕様の上を持っていくと、その部分に部品が挿入される。

4.2 準備

HSC エディタでは、図 5 のようなオブジェクト、リレーションとアクションが用意されている。

オブジェクト	FLAME, PHASE, CASE, NODE, MESSAGE FLAME-EDIT-WINDOW, WINDOW, etc.
リレーション	PARENT, CHILD, etc.
アクション	MOVE, COPY, DELETE, etc.

図 5: HSC エディタにおけるオブジェクト、リレーション、アクション

4.3 外部表現

4.1 で述べた要求 R をユーザが記述した例のうちで f_1 について、図 6 に示す。

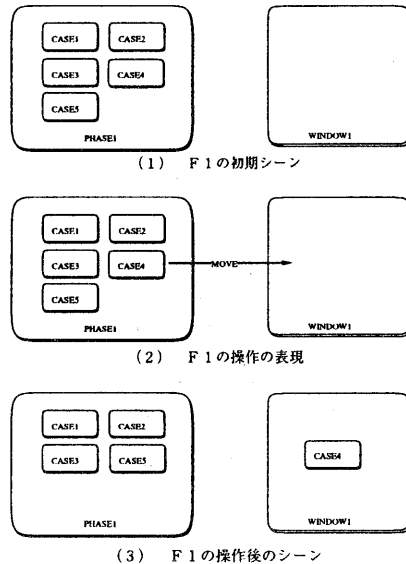


図 6: 外部表現

5 要求の相違

設計者は、獲得された要求をもとにこれらを満足するよう、すでに存在しているシステムとの差分を新たに設計しなければならない。

この作業に効果的な支援を与えるためには、すでに存在しているシステムと要求との差分を定量化し、新規にどの程度の要求を実現しなければならないかを示せばよい。そこで本稿では、要求の内部表現である FSM 上で、2つの FSM の差を定量化する。

2つの FSM の差の定量化には種々の手法が考えられるが、ここでは、トレース集合を使ってそれらの差を定義する。

定義 1 FSM S の長さ k のトレースの集合 $Tr_k(S)$ を次のように定義する。

$$Tr_k(S) = \{\sigma \in \Sigma^+ \mid \exists q \in S, q_0 \xrightarrow{\sigma} q, |\sigma| \leq k\}$$

FSM がループを含んでいる場合、トレースの集合の要素数は可算無限個になってしまう。そのためここでは、トレース集合そのものの差で定義するのではなく、トレース集合全体の大きさと差集合の大きさの比をとり 0 から 1 の値に正規化する。

2つの FSM S_1, S_2 におけるトレースの長さ k までの相違量 $dif_k(S_1, S_2)$ を次のように定義する。

定義 2 2つの FSM, S_1, S_2 の長さ k までのトレースに関する相違量 $dif_k(S_1, S_2)$ を次のように定義する。

$$dif_k(S_1, S_2) = \frac{|Tr_k(S_1) \oplus Tr_k(S_2)|}{|Tr_k(S_1) \cup Tr_k(S_2)|}$$

ここで \oplus は集合の対称差を表す。

トレースの長さ k を ∞ にしたときの $dif_k(S_1, S_2)$ の値で、 S_1 と S_2 の最終的な相違量を定義できる。

定義 3 2つの FSM, S_1, S_2 の相違量 $dif(S_1, S_2)$ を次のように定義する。

$$dif(S_1, S_2) = \lim_{k \rightarrow \infty} \frac{|Tr_k(S_1) \oplus Tr_k(S_2)|}{|Tr_k(S_1) \cup Tr_k(S_2)|}$$

次に FSM の相違量の例を与える。

図 7 にループを持たない 2つの FSM の例を示す。これらの相違量は、表 1 に示すように $k = 2$ で $\frac{1}{2}$ の値となる。表中で $Tr(S_1)$ と $Tr(S_2)$ の列は、それぞれ S_1 と S_2 の長さ k のトレースを示している。

図 8 にループを持った 2つの FSM の例を示す。表 2 に示すように、これらの相違量はトレースが長くなるごとに 1 に近づき、 $k = \infty$ で 1 の値をとる。

図 9 にループを持った 2つの FSM の例を示す。これらの相違量は、表 3 に示すように $k = \infty$ で $\frac{1}{2}$ になる。

図 7: ループを持たない FSM(例 1).

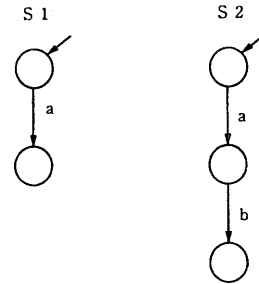
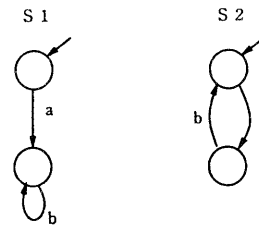


表 1: トレースの長さに対する相違量(例 1).

k	$Tr(S_1)$	$Tr(S_2)$	$dif(S_1, S_2)$
1	a	a	$\frac{0}{1} = 0$
2		ab	$\frac{1}{2}$

図 8: ループを持った FSM(例 2)



6 まとめ

本稿では、EM に基づく要求獲得システムの構成を提案し、その応用例を示した。また、すでに存在しているシステムと要求との差分を定量化する手法を示した。

今後の課題として、EM 及び各知識ベースの構築、変換アルゴリズム提案などが残されている。

参考文献

- [1] N. Shiratori, K. Sugawara, T. Kinoshita and

表 2: トレースの長さに対する相違量 (例 2). $\text{dif}(S_1, S_2)$ の一般項は $n \geq 2$ の値を示している. $n = 1$ のときの値は 0.

k	$Tr(S_1)$	$Tr(S_2)$	$\text{dif}(S_1, S_2)$
1	a	a	$\frac{0}{1} = 0$
2	ab	ab	$\frac{0}{2} = 0$
3	abb	aba	$\frac{2}{4}$
4	$abbb$	$abab$	$\frac{4}{6}$
5	$abbbb$	$ababa$	$\frac{6}{8}$
\vdots			\vdots
n	ab^{n-1}	\dots	$\frac{2(n-2)}{2n}$
\vdots			\vdots
∞			1

図 9: ループを持った FSM (例 3).

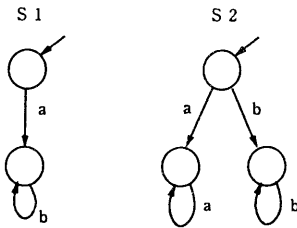


表 3: トレースの長さに対する相違量 (例 3).

k	$Tr(S_1)$	$Tr(S_2)$	$\text{dif}(S_1, S_2)$
1	a	a, b	$\frac{1}{2}$
2	aa	aa, bb	$\frac{2}{4}$
3	aaa	aaa, bbb	$\frac{3}{6}$
4	$aaaa$	$aaaa, bbbb$	$\frac{4}{8}$
5	a^5	a^5, b^5	$\frac{5}{10}$
\vdots			\vdots
n	a^n	a^n, b^n	$\frac{n}{2n}$
\vdots			\vdots
∞			$\frac{1}{2}$

G. Chakraborty, "Flexible Networks: Basic Concepts and Architecture", *IEICE Transactions on Communications*, Vol.E77-B, No.11, pp.1287-1294, 1994.

- [2] N. Shiratori, K. Sugawara, T. Kinoshita and G. Chakraborty, "Flexible Systems: A Step Towards New Generation Networks", *Proceedings of the 9th International Conference on Information Networking*, pp.477-482, 1994.
- [3] K. Sugawara, T. Kinoshita, G. Chakraborty and N. Shiratori, "Agent-Oriented Architecture for Flexible Networks", *ISADS95 (International Symposium on Autonomous Decentralized Systems)*, Phoenix, 1995.
- [4] M. Moser, S. Sugiura, S. D. Lee, K. Sugawara and N. Shiratori, "An Agent Framework for Flexible Networking", In *Proceedings of the FLAIRS-95 (Florida Artificial Intelligence Research Symposium) Workshop*, Miami, 1995.
- [5] 黄 錦法, 白鳥 則郎, "シーケンス図に基づく通信システム仕様記述法 HSC とその支援環境", *情報処理学会論文誌*, Vol.34, No.6, pp.1302-1311, 1993.