

インタフェースの情報を用いたシーケンス図作成支援システム

柘植 優一† 紫合 治‡

† 東京電機大学情報環境学研究科 ‡ 東京電機大学情報環境学部

組込みシステムの設計において、システムの振る舞いを記述することは非常に重要であるが、一般に使われるUMLを基準とするモデリングツールでは、その振る舞いを記述することは容易ではない。また、システムの設計のシナリオ分析の段階では信号のやり取りをシーケンス図によって表すことも多い。本稿は、システムの外部装置との信号のやり取りに焦点を当てたシーケンス図を、プロトコルという信号の種類と出現順序を示したインタフェースの情報を用いて、系統的に構築する方式の提案とその支援ツールの考察をまとめたものである。

Sequence Diagram Design System using Interface Specification

Yuichi Tsuge† Osamu Shigo‡

† Graduate School of Information Environment, Tokyo Denki University

‡ School of Information Environment, Tokyo Denki University

In a design of a system, it is very important to describe the behavior of the system. However, it is not easy to describe the complex behavior with existing modeling tools based on UML. Also, in the system analysis using scenarios, sequence diagram is widely applicable to express the exchange of signals among components. This paper describes a method to systematically build the sequence diagrams that express the signal exchanges between the system and its environment devices. The method uses the interface protocol specification which specifies the valid order of signals appeared through the interface port. The design support system for the method is also described.

1. はじめに

近年、ユビキタスシステムの発展に伴い、組込みシステムの重要性が高まってきている。組込み分野の分析ではシナリオ分析[1]、信号入出力例等の解析[2]にシーケンス図が広く利用されている。シーケンス図の記述をサポートするための従来の支援システムにはMicrosoft Office Visio[3]やRational Rose[4]などがあるが、これらのツールはその記述において、構造などシNTAX的なものは支援するが、信号名、順序などSEMANTECS的なものは自由度が高く支援が不十分であると言える。このためシーケンス図は一般にドキュメント目的が中心となっていた。よってこれらを補う方

式とその支援システムが望まれる。

本稿はシステムのインタフェースの仕様をプロトコルという入出力信号の種類、出現順序を定めた規則を用いて、システムの外部装置との信号のやり取りを示すシーケンス図を系統的に構築する方式を提案する。また提案方式を支援することを目的としたシステムについても検討・考察を行なった。

2. モデル

提案方式のシステムを設計する際に用いるモデルはシステムの構造を表す構成図、システムの外部装置とやり取りする信号の種類と出現順序を表すプロトコル、システムの振る舞いを示すシーケンス図の3つを

挙げることができる。

2.1. 構成図

システムの構成はカプセル、ポート、コネクタといった要素で表される。この構成図の要素は、リアルタイムシステム向けのモデリング技法である ROOM (Real-time Object Oriented Model) [5][6]に準拠する。カプセルはシステムの部品を表し、ポートはカプセルとカプセルの間の信号の送受信を行なうために、カプセルが信号をやり取りする他のカプセル 1 つにつき 1 つだけ設置され、ポートとポートはコネクタを介して繋がっている。このコネクタ上に信号は行き交う。ポートはプロトコルという信号の送受信を規定するインタフェースの情報を持ち、カプセルへの入力信号、出力信号の窓口となっている。図 1 に簡単な電話交換システムの構成図の例を挙げる。

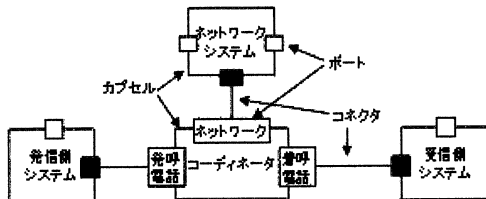


図 1 構成図の例

2.2. プロトコル

プロトコルはインタフェースの情報であり、具体的には信号の入出力の種類と出現順序を規定するものである。例として、図 1 中央に位置するコーディネータのカプセルの持つポートの発呼電話プロトコル記述を図 2 に示す。図の左側のリストはそのポートのプロトコルが取り得る入力信号と出力信号の集合を示している。見易さのため、入力信号は信号名の前に『?』, 出力信号は信号名の前に『!』を付け記される。また、右側の図は信号の出現順序とタイミングを規定したものである。小さめの黒丸は開始点であり、「オンフック」「ダイヤル中」などの大きめの楕円は状態を示す。状態と状態を繋ぐ矢印は状態間の遷移を示し、入力信号

を用いた入力イベントによる入力遷移と、出力信号を用いた出力イベントによる出力遷移から成る。開始点からは信号を持たない唯一つの特別な遷移によって状態へと繋がる。この状態がプロトコルの初期状態となる。図 2 から例を挙げると「相手待ち中」の状態では、入力イベントとして『?終了』の信号を受け取ったならば「オンフック」の状態に遷移し、出力イベントとして『!相手終了』の信号を発生させたのならば「通話中」の状態に遷移する。

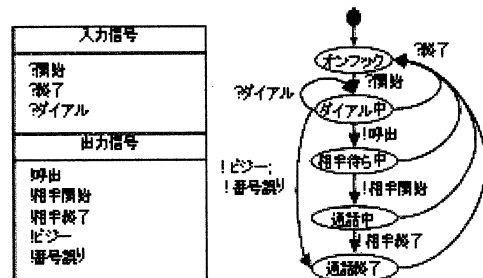


図 2 プロトコルの例

2.3. カプセルのシーケンス図

構成図におけるカプセルは、その振る舞いを示すシーケンス図を持つ。シーケンス図は動作の例の一つであり、その振る舞いを完全に示すためには複数のシーケンス図が必要となる。一般に使われるシーケンス図は複数カプセル間の信号のやり取りを記述したものであるが、ここでは一つのカプセルに着目した動作例として、カプセルのシーケンス図を検討する。このシーケンス図に対して、それがもつカプセルのことを対象カプセルと呼称したとき、このシーケンス図はシステムの外部装置との信号のやり取りに焦点を当てているため、オブジェクトは対象カプセルとそのカプセルが持つポートのみであり、対象カプセル以外の他のカプセルのオブジェクトなどはシーケンス図に記述しない。図 3 にカプセルのシーケンス図の例を載せる。この図は図 1 の「コーディネータ」を対象カプセルとしたときのシーケンス図である。図のようにシーケンス図はポートまたはカプセルを示すオブジェクト、ライフラ

イン、オブジェクト間の信号のやり取りを示すメッセージだけのシンプルなものとなっており、モデルの簡単化のため、UML2.0[7]にあるような復号フラグメント、状態不等式といった要素はない。また、使用されるオブジェクトはスタティックに存在しているものとする。これは組込みシステムにおいてオブジェクトをダイナミックに生成することはそれほど重要ではないためである[8]。

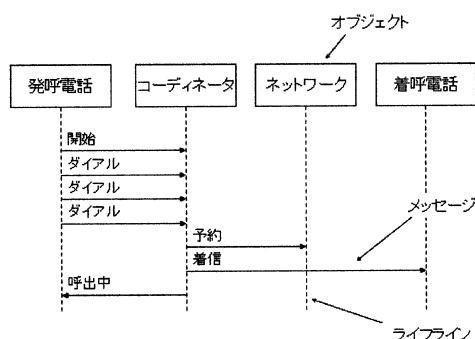


図 3 シーケンス図の例

3. プロトコルに沿ったシーケンス図

対象カプセルの持つ各シーケンス図は各々のポートのプロトコルに矛盾しない必要がある。この節ではプロトコルに矛盾しないように、シーケンス図が満たすべき条件、また、そうしたシーケンス図の作成手順を示す。

3.1. シーケンス図の条件

シーケンス図、またシーケンス図の集合の満たすべき条件を以下に示す。

- (1) プロトコルの情報を使って作成できる。すなわち、あるポートの入出力信号列はそのポートのプロトコルによって生成できること。
- (2) すべての可能な入力信号が現れている。すなわち、シーケンス図で対象カプセルが入力信号を待っている状態を「入力待ち点」と呼ぶ

と、入力待ち点において、その状態で取り得る入力信号が、そのカプセルのいずれかのシーケンス図に現れること。

上の条件の(1)によって、シーケンス図がポートのプロトコルの規定を満たしていることを保障する（必要条件）。また、(2)によって、シーケンス図の集合はポートのプロトコルで規定された可能な入力すべてのケースを網羅していることを保障する（十分条件）。

3.2. シーケンス図の作成

プロトコルに沿ったシーケンス図の作成手順を以下に示す。

- ①. あるタイミングでのすべてのポートの状態の組をそのタイミングのポート状態と呼んだとき、現ポート状態を、すべてのポートのプロトコルの初期状態としたものにする。または、既存のシーケンス図に現れる入力待ち点のポート状態とする。
- ②. 各々のポートのプロトコルの現状態から取り得る入力信号を一つだけ選びシーケンス図に加え、その信号を持つポートのプロトコルの現状態を更新する。
- ③. 必要ならば、各々ポートのプロトコルの現状態から取り得る出力信号を一つだけ選びシーケンス図に加え、その信号を持つポートのプロトコルの現状態を更新する。
- ④. ③の操作を 0 回以上繰り返す。
- ⑤. ②～④を必要なだけ繰り返す。

以上の手順で(1)の条件に沿ったカプセルのシーケンス図が作成できる。

3.3. シーケンス図の補填

3.2 で述べた手順で作成したカプセルのシーケンス図の集合がすべての可能な入力を網羅する方法を以下に示す。

- ①. シーケンス図のある入力待ち点 P においてその状態で取り得る入力信号 S が、いずれかのシーケンス図の同状態の入力待ち点で現れて

いるかをチェックする。

- ②. 入力信号 S が現れていないならば，対象カプセルのもつシーケンス図の集合に新たなシーケンス図を追加し，そのシーケンス図の現ポート状態に入力待ち点 P でのポート状態をコピーする。
- ③. その新たなシーケンス図に入力信号 S を追加し，その信号を持つポートのプロトコルの現状態を更新する。
- ④. その後，3.2 のシーケンス図作成の手順でシーケンス図を完成させる。
- ⑤. ①～④の操作を対象カプセルの持つすべてのシーケンス図のすべての入力待ち点において適応する。

以上の手順で対象カプセルのシーケンス図の集合は(2)の条件に沿ったものとなる。

4. 支援ツール

我々は提案した方式を支援するシステムの検討・考察を行なった。支援システムを使った対象システムの設計は，構成図の作成，プロトコルの作成，シーケンス図の作成の3つの段階に亘り行なう。

4.1. 構成図作成

まず，設計するシステムを表すカプセルを作る。新しいカプセルを作成すると描画領域にカプセルとなる大きな四角が描画される。このカプセルにポートを必要だけ追加する。カプセル，ポートの名前は作成された際デフォルト名が付けられているので，それぞれ選択し，名前の変更を行うことができる。また，ポートはインタフェースの情報としてプロトコルを設置できる。図 4 に構成図の設計画面を示す。

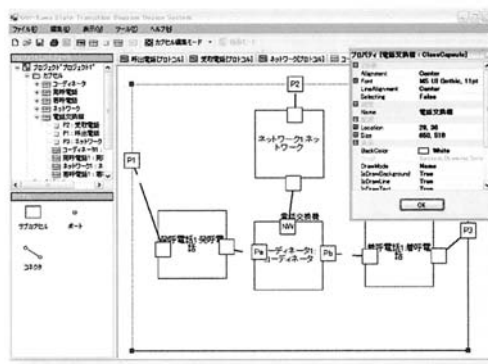


図 4 構成図の設計画面

4.2. プロトコル作成

プロトコルを新規に作成すると描画領域に開始点と初期状態，それらを繋ぐ初期遷移が配置される。はじめにプロトコルの状態を必要だけ描画領域に設置する。次に状態間の遷移を適切な状態と状態を繋ぐように設置する。遷移は入力信号ならば「?」が，出力信号ならば「!」が信号名の前に表示される。入力信号はシーケンス図ではそのプロトコルを設置したポートから対象カプセルへのメッセージになり，対して出力信号は対象カプセルからそのプロトコルを設置したポートへのメッセージとなる。状態と遷移は選択することでその名前を変更することができる。図 5 にプロトコルの設計画面を示す。

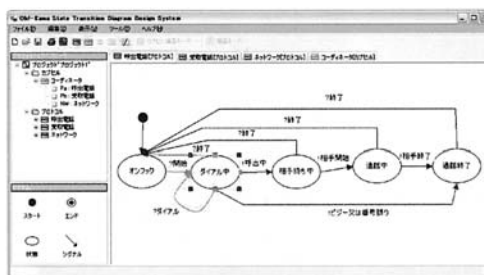


図 5 プロトコルの設計画面

4.3. シーケンス図作成

新規作成すると対象のカプセルに対するシーケンス図を作成する描画領域に、対象カプセルとその設置したポートを表すオブジェクトとそのライフラインが描画される。次にそのポートを表すオブジェクトに繋がるライフラインを選択するとそのポートのプロトコルの現状態から取り得る信号を選び、メッセージとして設置することができる。このように支援システムが現状態から取り得る信号のリストを自動で選出し設計者はそれを選択するだけで済むので、そのポートの持つプロトコルの現状態から取りえない信号を設置してしまうといった誤りを回避することができる。図 6 にシーケンス図の設計画面を示す。

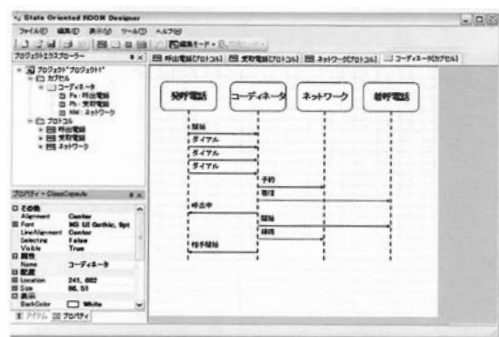


図 6 シーケンス図の設計画面

4.4. シーケンス図の補填

入力信号の不備のチェックのボタンを押すと、現在作成されているシーケンス図の集合に対してポートのプロトコルで規定された可能な入力のすべてのケースを網羅しているかチェックされる。入力すべてのケースが網羅されていないならば、どのようなポート状態のとき、どの入力信号が選択されていないかメッセージボックスによって通知されるので、設計者はそれ情報を元にシーケンス図の追加・修正を行なっていけばよい。

5. 例

支援システムを用いて具体的なシステムの設計を行なってみる。ここでは簡単な電子レンジを例に設計を行なう。

5.1. 対象システムの仕様

設計する電子レンジの仕様に付いて述べる。電子レンジはドアが一つ付いており、ドアを開け、中に温めるものを入れる。また、電子レンジにはあたためボタンが一つだけ付いていて、ドアが閉まっているときのみボタンを押すと一定時間ドアの中のものにマイクロ波を当て温める。ドアが開いているときに温めボタンを押しても温め動作を行なわない。また、温め中にドアを開けるとマイクロ波を当てるのを止め、温め動作を中止する。中止した後にドアを閉めてあたためボタンを再び押すと、また同じように一定時間温め動作をするものとする。

5.2. 対象システムの設計

対象システムとなる電子レンジの設計は、あたためボタンやマイクロ波をまとめて制御する「制御部」を対象に行なう。

5.2.1. 構造図の作成

カプセルを1つ作成する。作成されたカプセルの名前を制御部に変更する。この電子レンジの制御部にはどのようなポートが設置されているかを考える。仕様からあたためボタンに対するポート、マイクロ波に対するポート、ドアに対するポート、また、あたため動作の施行時間を計るタイマーに対するポートが必要ことがわかる。制御部に4つポートを設置し、それぞれ名前を「P あたためボタン」、「P マイクロ波」、「P ドア」、「P タイマー」とする。図 7 に作成した構造図を示す。

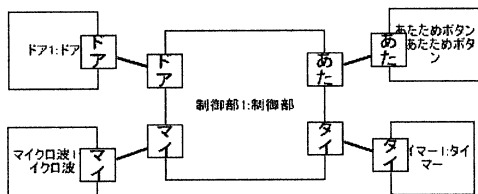


図 7 作成した構造図

5.2.2. プロトコルの作成

次に作成した構造図のポートに設置するプロトコルを仕様に沿うように作成する。はじめにドアのポートに対応するプロトコルを作成する。アイテム欄の状態を選択し、それを描画領域に合計 2 個配置し、名前をそれぞれ「閉」、「開」と付ける。また、それぞれの状態への遷移を配置する。閉から開への遷移を「開く」、開から閉への遷移を「閉じる」と名前を付け替える。同様に、残りの P あたためボタン、P マイクロ波、P タイマーを仕様に沿うように作成する。作成したプロトコルを対象カプセルの持つ対応するそれぞれのポートに設置する。作成した 4 種類のプロトコルを図 8 に示す。

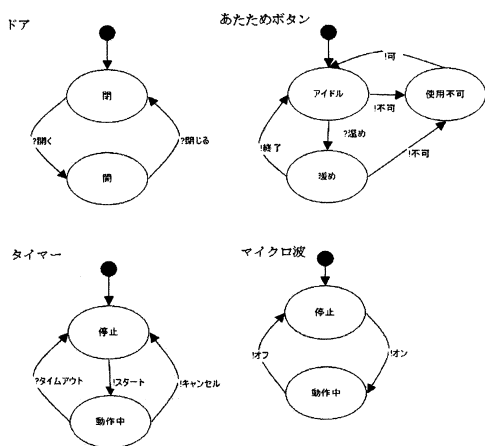


図 8 作成したプロトコル

5.2.3. シーケンス図の作成

次に電子レンジの制御部の振る舞いを示すシーケンス図を作成する。まず、あたためボタンを押し、あたため終了の時間をタイマーが知らせ温め動作が終わるまでの振る舞いを示すシーケンス図を作成する。シーケンス図作成画面にし、シーケンス図を新規作成するとオブジェクトとしてカプセルの制御部とポートの P ドア、P あたためボタン、P タイマー、P マイクロ波、またそれぞれのオブジェクトのライフラインが描画領域に配置される。あたためボタンのオブジェクトから伸びるライフラインを選択するとそのポートの持つプロトコルの現状態から取り得る信号のリストが出るので、その中の「?温め」の信号を選択すると、P あたためボタンのポートから対象カプセルへのメッセージとして追加される。次に、P タイマーを選択し「!スタート」の信号を、またその後、P マイクロ波を選択し「!オン」の信号をそれぞれ追加する。次に、温め動作の施行時間の終了をタイマーが制御部に知らせるため、タイマーのオブジェクトから伸びるライフラインを選択し、出た信号リストの中の「?タイムアウト」の信号を選びシーケンス図にメッセージを追加する。そして、その入力に対する出力も同様に追加する。これでこのシーケンス図の作成は終わる。また、最初の状態からドアを開き、その後ドアを閉じたときの制御部の振る舞いを示すシーケンス図も同様に作成する。作成した 2 つのシーケンス図を図 9 に示す。

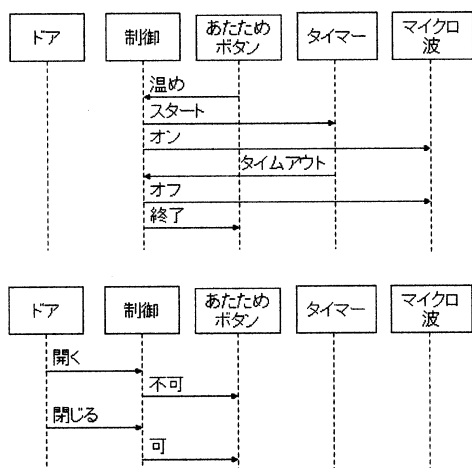


図 9 作成したシーケンス図

5.2.4. 入力信号不備のチェック

シーケンス図の不備のチェックのボタンを押し、作成したシーケンス図に入力信号の不備がないかをチェックする。チェックすると図 9 の上のシーケンス図のタイマーからの?タイムアウトのメッセージが出るところであるが、このとき他にもドアから?開くのメッセージを取り得ることが示される。よってこのメッセージを補う 3 つの目的のシーケンス図を作成する。このシーケンス図も 5.2.3 と同様に作成していく。作成したシーケンス図を図 10 に示す。この後、再び入力信号の不備のチェックを行っても入力信号の不備があることは示されないで、この電子レンジの振る舞いを示すシーケンス図は以上に挙げた 3 種類で十分であることが分かる。

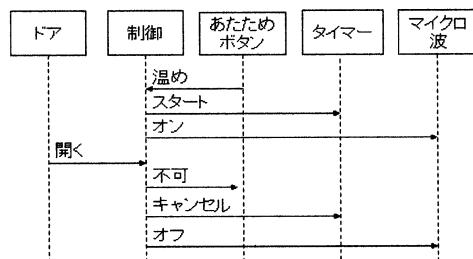


図 10 足りない入力信号を補ったシーケンス図

6. 考察とまとめ

インタフェースの情報を用いてシーケンス図を系統的に作成する方式を提案した。インタフェースの情報はプロトコルという外部装置とやり取りする入出力信号の種類と出現順序を規定するものを導入した。これにより外部装置との信号のやり取りに焦点を当てたシーケンス図をプロトコルの情報を用い、利用できるイベントを制限して系統的に作成できる。また、提案方式のサポートを目的とする支援システムの検討・考察を行なった。

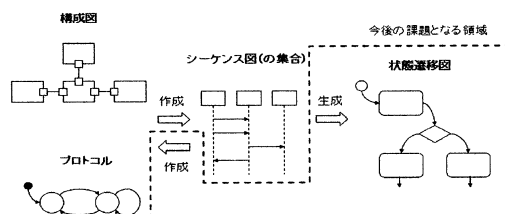


図 11 設計の流れ

今後の課題として、複人数によって提案方式を用いた実際のシステムの設計・評価が考えられる。

また、システムの設計を、提案方式を用いて設計を行なったとき、プロトコルの中でも状態数や遷移数が割合少なく比較的簡単なものは他のシステムの設計で

も完全に同じものや、また状態数、信号の入力・出力のタイミングは同じであるが、信号名のみ違うといったプロトコルを見掛けたりする。よって、そういったプロトコルはライブラリとして保存しておき、システムの設計時にそのライブラリから適当なプロトコルを読み込み、そのまま使用、または修正ができるようにすれば、設計における時間の短縮が期待できる。

提案方式におけるシステムの設計の流れはシステムの構成図、プロトコルを作成し、それを用いてシーケンス図を作成していくといったものであった。システムの設計の流れを図 11 に示す。図のように逆の流れとして、シーケンス図から構成図、プロトコルを作成していくというものを考えている。また、システムの振る舞いを示す形としてシーケンス図の他に状態遷移図が使われることは良くある。システムの振る舞いとして、シーケンス図は動作例であり、状態遷移図は一般則であるといえる。提案方式ではシーケンス図を系統的に構築するものであったが、さらに、作成したシーケンス図の集合から状態遷移図を生成するという流れが考えられる[9][10]。

ーム社、2006。

- [8] 成沢文雄, 納谷英光, 横山孝典 : 組み込み制御システムのためのオブジェクト指向コード生成ツール, 情報処理学会論文誌 vol.46, 2005
- [9] Jon Whittle and Johann Schumann : Generating Statechart Design From Scenarios, White paper, IBM, 1998.
- [10] 紫合治 : 環境の仕様を用いたシナリオから状態マシンの生成, FOSE04, 2004.

参考文献

- [1] 大西淳、郷健太郎 : 要求工学、共立出版、2002.
- [2] 谷口考治他 : Java プログラムの実行履歴に基づくシーケンス図の作成、FOSE04, 2004.
- [3] Microsoft : Microsoft Office Visio,
<http://www.microsoft.com/japan/office/visio/prodinfo/default.msp>
- [4] IBM : Rational Rose,
<http://www-06.ibm.com/jp/software/rational/products/design/rosetech/>
- [5] Bran Selic and James Rumbaugh : Using UML for Modeling Complex Real-time Systems,
- [6] Bran Selic, et al : Real-time Object Oriented Modeling, John Wiley and Sons, Inc., 1994.
- [7] Object Management Group : UML2.0 仕様書 2.1 対応、オ