# Model Synthesis from Imprecise Specifications

Bill Mitchell[1], Robert Thomson[2], Paul Bristow[2]

[1] Department of Computing, University of Surrey, Guilford,
Surrey GU2 7XH, UK
[2] Motorola UK
Research Lab, Viables Industrial Estate, Hampshire RG22 4DP, UK

**Abstract.** The paper defines a formal semantics for MSC scenarios that is a weakening of the state semantics from [6], whilst permitting some additional semantics in the spirit of Live Sequence Charts (LSCs) [4]. The semantics here differs from that of LSCs in that mandatory behaviour is defined dynamically within the domain of possible scenarios. This permits a semantics which uses domain knowledge to define when compositions of imprecise requirements are valid. This has been implemented by Motorola UK Research Labs, and is being used in a pilot study for a new telecommunications mobile 3G handset.

## 1 Annotated Events

Industrial MSC [7] scenarios have rather imprecise compositional semantics. The paper describes a weakening of standard model synthesis semantics ([1], [3], [6]) that permits valid composition of imprecise scenario specifications. This work has been applied to industrial requirements specifications in Motorola case studies.

Consider the leftmost MSC in figure 1, which is a requirements scenario for a wireless mobile handset. This describes how a WAP 'Browser' process downloads a Java application iteratively from the 'Air Interface' process until it receives the 'EOF' message, or it detects that the file is corrupted.

The extended hexagonals are MSC condition symbols that describe which operational phase is active at any time. We will refer to them as phase symbols from now on. Phase symbol labels will be identified with propositional boolean formulae in the paper.

An MSC defines a partial order semantics on the order that system events can be observed to occur. A message $m$ is translated into a send event $!m$ and a receive event $?m$.

**Definition 1** *Define $T(P)$ to be the set of traces generated by the process $P$ in an MSC $M$.*

An event $x$ in an MSC $M$ occurs in the scope of phase symbol $u$ if the first phase symbol prior to $x$ within the process it belongs to is $u$. Let $\mathcal{P}$ be the set of phase symbols associated with an MSC $M$, and let $\psi$ be a map that defines the set of phase names for each symbol. I.e. $\psi : \mathcal{P} \longrightarrow 2^{\mathsf{Ph}}$, where $\mathsf{Ph}$ is the set of phase names.

Where $E$ is the set of events for an MSC $M$, let $\phi : E \longrightarrow \mathsf{Ph}$ be the function that maps each event $e$ to the set of phases it belongs to, that is the set $\psi(u)$, where $e$ is in the scope of $u$.

Mitchell B., Thomson R. and Bristow P. (2004).
Model Synthesis from Imprecise Specifications.
In *Proceedings of the 2nd International Workshop on Verification and Validation of Enterprise Information Systems*, pages 20-25
DOI: 10.5220/0002669500200025

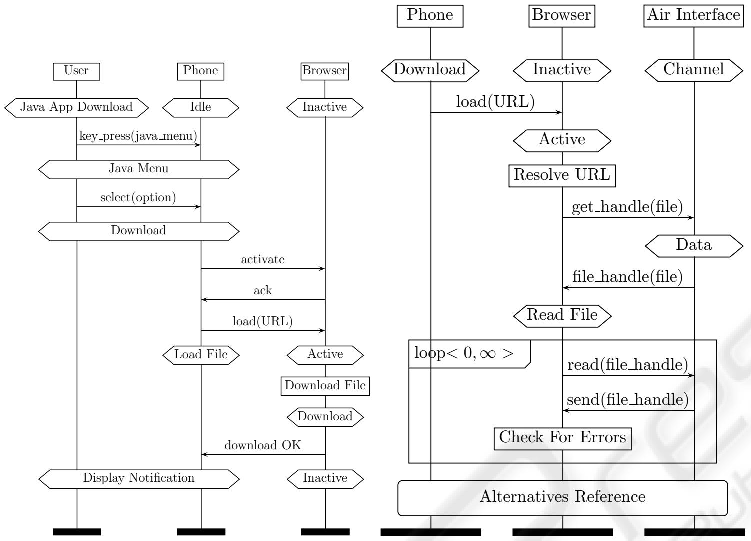

**Fig. 1.** MSC Requirement Scenarios

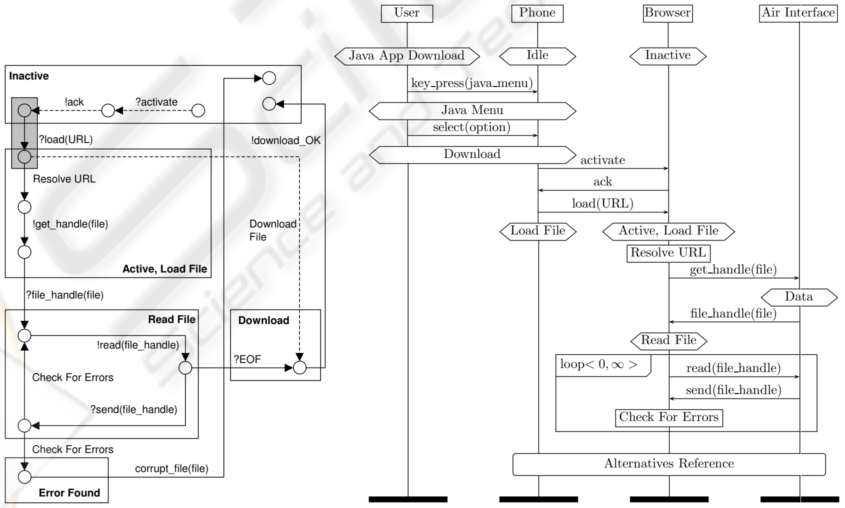

**Fig. 2.** Phase Automaton, and Overlap Scenario

**Definition 2** *Define the phase traces for a process $P$ in an MSC scenario $M$ to be sequences of triples:*

$$(S_0, e_0, S_1)\,(S_1, e_1, S_2) \cdots (S_n,\, e_n,\, S_{n+1})$$

*where $e_0,\, \ldots,\, e_n$ is an event trace of $P$, $S_i \subseteq \mathsf{Ph}$, $\phi(e_i) = S_i$, and $S_{n+1}$ is the last phase for process $P$ in the scenario $M$.*

Each triple in a phase trace is referred to as an *annotated event*.

## 2 Dynamic Constraints

A temporal model $T$ consists of a directed graph $G$, with vertex labelling $\nu : G_V \longrightarrow 2^{\mathsf{Ph}}$, edge labelling $\varepsilon : G_E \longrightarrow E$, and some vertex $i$ that represents the initial moment. Temporal formulae are defined as usual:

- $T, v \vDash \langle e \rangle \phi$ iff there is an edge $(v, w) \in G_E$ such that $\varepsilon(v, w) = e$, and $T, w \vDash \phi$
- $T, v \vDash [e]\phi$ iff for every edge $(v, w) \in G_E$ where $\varepsilon(v, w) = e, T, w \vDash \phi$
- $T, v \vDash \Box\phi$ iff $T, v \vDash \phi$ and $T, w \vDash \Box\phi$ for every edge $(v, w) \in G_E$
- $T, v \vDash \Diamond\phi$ iff there is some vertex $w$ reachable from $v$ such that $T, w \vDash \phi$

The satisfiability of ordinary boolean formulae is defined as usual. Formula $\phi$ is satisfied in $T$ when $T, i \vDash \phi$. $\phi$ is valid when it is satisfied in every model, when we write $\vdash \phi$.

**Definition 3** *For a set $S \subseteq \mathsf{Ph}$, define $\bigwedge S = \bigwedge_{x \in S} x$. For a phase trace $t = (S, e, S') \cdot t'$, define its temporal semantics as*

$$\|t\| = \bigwedge S \wedge \langle e \rangle (\bigwedge S' \wedge \|t'\|)$$

*A context $\mathcal{C}$ is any temporal formulae over $\mathcal{P}$ and $E$.*

A temporal context controls how phases are related across the requirements scenarios.

**Definition 4** *For context $\mathcal{C}$ we define phase trace $t$ to match phase trace $t'$ when*

$$\vdash \mathcal{C} \Rightarrow (\|t\| \Rightarrow \Diamond \|t'\|)$$

Intuitively $t$ matches $t'$ if after some initial delay, $t'$ becomes the same as $t$ within the context defined by $\mathcal{C}$.

**Definition 5** *Let $a = (S, e, S')$ be an annotated event. When $\nvdash \mathcal{C} \Rightarrow (\bigwedge S \Rightarrow \bigwedge S')$ define $a$ to be a phase transition event.*

*Define a phase transition trace to be a trace of annotated events terminating with a phase transition event.*

Let $t_1$ be the phase transition trace of the phase trace $t_0$ consisting of $t_1 = (\{\mathsf{Inactive}\}, ?\mathsf{activate}, \{\mathsf{Inactive}\})\ (\{\mathsf{Inactive}\}, !\mathsf{ack}, \{\mathsf{Inactive}\})\ (\{\mathsf{Inactive}\}, ?\mathsf{load(URL)}, \{\mathsf{Active}\})$
In the rightmost MSC of figure 1 the initial annotated event of process 'Browser' is $t_2 = (\ \{\mathsf{Inactive}\}, ?\mathsf{load(URL)}, \{\mathsf{Load\ File}\})$.

From this we can prove $\vdash \Box([\mathsf{load(URL)}](\mathsf{Active} \Rightarrow \text{`}\mathsf{LoadFile}\text{'})) \Rightarrow (\|\mathsf{t_1}\| \Rightarrow \Diamond\|\mathsf{t_2}\|)$.

## 3 Phase transition simulation

$$
\begin{aligned}
P \mid Q &= P \lhd Q + P \rhd Q \\
a \cdot P \lhd b \cdot Q &= a \cdot P {\lhd\!|}\, b \cdot Q \;\text{ if }\; a \sqsupset_{\mathcal{C}} b \\
a \cdot P \lhd b \cdot Q &= a \cdot (P \lhd b \cdot Q) \;\text{ if }\; a \not\sqsupset_{\mathcal{C}} b \\
P \rhd Q &= Q \lhd P \\
0 \lhd Q &= 0 \\
a \cdot P {\lhd\!|}\, b \cdot Q &= (a \cup b) \cdot (P {\lhd\!|}\, Q) \;\text{ if }\; a \sqsupset_{\mathcal{C}} b \;\text{ and }\; \neg\eta_{\mathcal{C}}(a) \\
a \cdot P {\lhd\!|}\, b \cdot Q &= (a \cup b) \cdot (P \parallel Q) \;\text{ if }\; a \sqsupset_{\mathcal{C}} b \;\text{ and }\; \eta_{\mathcal{C}}(a) \\
a \cdot P {\lhd\!|}\, b \cdot Q &= a \cdot P + b \cdot Q \;\text{ if }\; a \not\sqsupset_{\mathcal{C}} b \;\text{ and }\; \eta_{\mathcal{C}}(a) \\
a \cdot P {\lhd\!|}\, b \cdot Q &= a \cdot P \;\text{ if }\; a \not\sqsupset_{\mathcal{C}} b \;\text{ and }\; \neg\eta_{\mathcal{C}}(a) \\
0 {\lhd\!|}\, Q &= 0 \\
a \cdot P \parallel b \cdot Q &= (a \cup b) \cdot (P \parallel Q) \;\text{ if }\; a \sqsupset_{\mathcal{C}} b \\
P \parallel Q &= Q \parallel P \\
0 \parallel Q &= Q \\
a \cdot P \parallel b \cdot Q &= a \cdot P + b \cdot Q \;\text{ if }\; a \not\sqsupset_{\mathcal{C}} b \;\text{ and }\; b \not\sqsupset_{\mathcal{C}} a
\end{aligned}
$$

**Fig. 3.** Phase Transition Process Algebra

For annotated events $a = (S, e, S')$ and $b = (U, g, U')$ define $a \sqsupset_{\mathcal{C}} b$ when $e = g$, $\vdash \mathcal{C} \Rightarrow (\bigwedge U \Rightarrow \bigwedge S)$ and $\vdash \mathcal{C} \Rightarrow (\bigwedge U' \Rightarrow \bigwedge S')$. Define $P$ to simulate process $Q$ within context $\mathcal{C}$, written as $P \sqsupset_{\mathcal{C}} Q$, if $\forall a$ such that $Q \xrightarrow{a} Q'$ there is some $a'$ where $P \xrightarrow{a'} P'$ such that $a' \sqsupset_{\mathcal{C}} a$ and $P' \sqsupset_{\mathcal{C}} Q'$.

For annotated events $a_i$ and phase trace $t = a_0 \cdot a_1 \cdots a_{n-1}$, let $P \xrightarrow{t} P'$ denote that there are processes $P_i$, for $0 \leq i \leq n$, such that $P_i \xrightarrow{a_i} P_{i+1}$, $P_0 = P$ and $P_n = P'$.

**Definition 6** *Define $P$ to simulate the phase transitions of process $Q$ within context $\mathcal{C}$, written as $P \trianglerighteq_{\mathcal{C}} Q$, when the following holds. For all phase transition traces $t$ such that $Q \xrightarrow{t} Q'$, and for all phase traces $\tau$ that match $t$, whenever there is a process $P'$ such that $P \xrightarrow{\tau} P'$ then $P' \sqsupset_{\mathcal{C}} Q'$.*

**Definition 7** *Let $\{M_i \mid 0 \leq i \leq n\}$ be a set of scenarios, let $Q_i$ be a process from $M_i$ for each $i$. That is each $Q_i$ defines exactly the observed behavior of one process in scenario $M_i$.*

*We define process $P$ to be the phase transition representation of processes $Q_i$ when $P \trianglerighteq_{\mathcal{C}} Q_i$ for each $i$. Define the* overlaps *of $P$ to be those phase transition traces of $P$ that are not contained in any of the $Q_i$.*

## 4 Phase Transition Processes

In figure 3 we briefly describe a process algebra that defines how to synthesise a phase transition representation from a set of processes described by the requirements scenarios.

Let $\mathcal{A}$ be the set of annotated events. Let $\eta_{\mathcal{C}} : \mathcal{A} \longrightarrow \mathbb{B}$ be a boolean valued function that defines when an annotated event is a phase transition. That is $\eta_{\mathcal{C}}(S, e, S') = \mathtt{t}$ when $\nvdash \mathcal{C} \Rightarrow (\bigwedge S \Rightarrow \bigwedge S')$. For annotated events $a = (S, e, S')$ and $b = (U, e, U')$ define $a \cup b = (S \cup U, e, S' \cup U')$.

**Proposition 8** *Given a set Q of processes $Q_i$ from requirements scenarios $M_i$ for $0 \leq i \leq n$, then*

$$P = Q_0 \mid Q_1 \mid \cdots \mid Q_n$$

*is a phase transition representation of Q. Where $\mid$ is defined by the axioms of figure 3.*

*If $P'$ is another phase transition representation of $Q$, then $P' \sqsupset_{\mathcal{C}} P$. That is $P$ is canonical up to simulation equivalence. Define $P$ to be the phase transition process for $Q$.*

Figure 2 describes one of the overlaps given by the phase transition process of the ‘Browser’ processes in figure 1.

**Definition 9** *A phase automaton consists of a set of events $E$, states $\mathcal{P}$ and transitions from $\mathcal{P} \times E \times \mathcal{P}$. A phase automaton also has a function $\psi : \mathcal{P} \longrightarrow 2^{\mathsf{Ph}}$.*

Given a process that has annotated events for actions, we can translate it into a phase automaton consisting of the following state transitions. Each action transition $P \xrightarrow{a} P'$, where $a = (S, e, S')$, defines a state transition $u \xrightarrow{e} u'$ for each $u \in \psi^{-1}(S)$, and $u' \in \psi^{-1}(S')$.

**Proposition 10** *The phase automaton of a phase transition process is always finite.*

Figure 2 is the phase transition process of the two ‘Browser’ processes defined in figures 1 and 1. Those states that belong to the same phase are grouped together in a box labelled with the phase name. The dotted arrows represent the part of the process behavior that is exclusive to figure 1. The solid arrows are the behavior that is defined by figure 1. The grey box denotes where phase trace $t_1$ matches $t_2$. This match defines where the two ‘Browser’ processes from figures 1 and 1 are joined together.

In general the phase transition process $P$ is built by joining together specification scenario processes wherever there is a match between phase transition traces. The process algebra of figure 3 captures this idea formally.

For an annotated event $a = (S, e, S')$ let $*a = e$. Let $P$ be a process that has annotated events as actions. Let $A$ be a state machine that accepts some subset of $E^*$. Define $A \sqsupset P$, if for all $P \xrightarrow{a} P'$ there is some $A \xrightarrow{*a} A'$ such that $A' \sqsupset P'$. That is when reduced to a process over plain events $P$ can be simulated by $A$ in the usual sense.

**Proposition 11** *Let $P$ be the phase transition representation of a set of processes $Q_i$ from MSC scenarios $M_i$, where the temporal context $\mathcal{C}$ is a tautology.*

*Let $A$ be the state chart of the $Q_i$ processes defined according to the semantics of [6] where each set of phase names attached to a phase symbol from the $M_i$ is mapped to a unique state name.*

*Then*

$$A \sqsupset P$$

### 4.1 Conclusions

The research reported in this paper is a consequence of case studies of Motorola requirements scenarios. These highlighted that standard scenario modeling techniques needed to be extended in order to be legitimately applied to MSC scenarios that are not precise in their compositional semantics. The work reported here has been incorporated into the *ptk* tool suite [2], has been validated against a suite of industrial requirements specifications, and is being applied in a pilot study for a new mobile handset for Motorola.

## References

1. R. Alur, K. Etessami, M. Yannakakis, Inference of Message Sequence Charts, Proceedings 22nd International Conference on Software Engineering, pp 304-313, 2000.
2. P. Baker, P. Bristow, C. Jervis, D. King, B. Mitchell, Automatic Generation of Conformance Tests From Message Sequence Charts, Proceedings of 3rd SAM Workshop 2002, Telecommunications and Beyond: The Broader Applicability of MSC and SDL, pp 170-198, LNCS 2599.
3. Yves Bontemps, Pierre-Yves Schobbens, Synthesis of Open Reactive Systems from Scenario-Based Specifications, Third International Conference on Application of Concurrency to System Design (ACSD'03)
4. Werner Dam, David Harel, LSCs: Breathing life into message sequence charts, Formal Methods in System Design, 19(1):45-80, 2001.
5. Johann Schumann, Jon Whittle, Generating Statechart Designs From Scenarios, Proceedings of the 22nd international conference on on Software engineering, 2000.
6. Sebastian Uchitel, Jeff Kramer, Jeff Magee, Synthesis of Behavioral Models from Scenarios, IEEE Transactions on Software Engineering, vol. 29, no. 2, February 2003
7. Z.120 (11/99)ITU-T Recommendation - Message Sequence Chart (MSC)