Similarity for Reuse of Specifications in Communication Software Development

U. Yamamoto† E. S. Lee‡ N. Shiratori†

†Research Institute of Electrical Communication
Tohoku University
Sendai, JAPAN

‡Dept. of Information Engineering
Sung-Kyun-Kwan University
Seoul, KOREA

Abstract

The reuse of existing software is one of the most effective way for software development. We have focused on the specification process with FDTs (Formal Description Techniques), and have proposed a new concept of similarity based on LTSs (Labelled Transition Systems) as a criterion to reuse specifications. However, it's definition has some problems, such that (a) it can't be applied to an LTS with some loops, and (b) the definition of similarity between actions is not clearly expressed.

In this paper, we remove these problems in order for our approach to be widely applicable to practical use. For first problem, we extend the definition of similarity to be able to apply to an LTSs with some loops. For second one, we consider that the similarity of actions is defined based on not only the name of actions, but also the attribute of actions and the way of occurrence of actions as the temporal ordering.

1 Introduction

Reuse of software products has been researched as the breakthrough to improve the productivity and quality of software development, because we can easily obtain the target product by slightly modifying the existing ones. We focus on the specification process with FDTs (Formal Description Techniques), by which the designer can describe the specifications formally and rigorously, and the specifications written in FDTs can be automatically verified, validated and implemented[1, 2, 3]. FDTs such as SDL[4], Estelle[5] and LOTOS[6] have been developed and researched as the international standard.

To reuse the existing specifications, we need some criterion to retrieve the most appropriate specification. We have already proposed a concept of similarity between specifications as the criterion to retrieve the similar specifications[7]. In [7], we have judged the similarity between specifications using LTSs (Labelled Transition Systems). LTSs are state transition graphs in which nodes represent states and edges represent state changes. The main advantage is that this approach can be commonly used among most FDTs because most FDTs have the transition graphs as a representation way of the dynamical meanings of specifications. We have formally defined the similarity of LTSs and the degree of similarity to judge which of LTSs is similar to the requirement specification given from the designer.

However, it still has some serious constraints as follows:

- We assumed that every LTS has no loops, which means a sequence of actions whose execution from a state leads arrival of the same state. However, most LTSs generally have some loops, so our similarity isn't applicable in most case of the reuse of specifications.
- Although the similarity of specifications has been formally defined in a some degree, it was still incomplete due to the ambiguity in the similarity of actions. We assumed that the information for the similarity of actions have already been stored in the knowledge base, or is directly given from the designer. This provides a serious constraint for the practical use.

In this paper, we discuss these constraints and give some solutions in order for our approach to be widely applicable to practical reuse of specifications. For the first issue, we extend the definition of similarity to be applicable in the case that LTSs have some loops. For the second issue, we give three criteria, (1) the name of actions, (2) the attribute of actions and (3) the way of occurrence of actions as the temporal ordering, to determine the similarity of actions. And we show a retrieval method of similar specifications using these similarities.

The outline of this paper is as follows: Section 2 shows a concept of similarity between LTSs which we have extended the similarity proposed in [7], and define the similarity of actions, to overcome some problems in the previous definition. In Section 3, we show the retrieval method of similar specifications with our concept of similarity which is defined in Section 2. In Section 4, we show a simple evaluation of our approach and a specification environment based on the reuse of

0-8186-7216-1/95 $04.00 © 1995 IEEE

specifications which now we are implementing. Finally, Section 5 contains some concluding remarks.

2 Similarity of Specifications

In this section, we discuss a concept of similarity between two specifications.

2.1 Labelled Transition System

Throughout this paper, it is assumed that we have an *alphabet* $\mathcal{A}$, a finite set of symbols. Its element is called an *action*. This corresponds to a primitive event of a system and this is assumed to be externally observable and controllable. An internal action which can't be externally observed is represented as τ and let $Act = \mathcal{A} \cup \{\tau\}$. Act^* denotes the set of all sequences of actions and we write ε as the empty sequence. $\widehat{r}$ is the resulting sequence of actions r where τ is removed.

An LTS is a state transition graph in which nodes represent states and edges represent state changes. This is formally defined as follows:

Definition 1 (Labelled Transition System)
A *labelled transition system* L is a quadruple $\langle S, A, T, s_0 \rangle$, where

1) S is a nonempty set of *states*,

2) A is a subset of Act,

3) $T \subseteq S \times A \times S$ is a *transition relation* and

4) $s_0 \in S$ is the *initial state* of L.

The transition relation defines the dynamical change of states as actions may be performed. For $(s, a, s') \in T$, we normally write $s \xrightarrow{a} s'$ and this may be interpreted as "in the state s an action a can be performed and after the action the state moves to s'". We also write τ and ϵ as an internal action and an empty sequence of actions, respectively.

If $r = a_1 a_2 \ldots a_n \in Act^*$, then

- $s \xrightarrow{r} s'$ stands for $s \xrightarrow{a_1}\xrightarrow{a_2} \ldots \xrightarrow{a_n} s'$.
- $s \overset{r}{\Rightarrow} s'$ stands for $s \xrightarrow{\tau^*}\xrightarrow{a_1}\xrightarrow{\tau^*}\xrightarrow{a_2}\xrightarrow{\tau^*} \ldots \xrightarrow{\tau^*}\xrightarrow{a_n}\xrightarrow{\tau^*} s'$.

Note that $s \xrightarrow{\varepsilon} s$ for all states s.

Now, we define a simulation and a bisimulation as a relation between LTSs[8]. Intuitively, if LTS L_1 is simulated by LTS L_2, the behavior of L_1 is completely included in L_2. So, a simulation helps verification of the consistency in the stepwise refinement of specifications. On the other hand, if L_1 is bisimulated by L_2, their observable behaviors are completely same.

Definition 2 (Simulation)
Let $L_1 = \langle S_1, A_1, T_1, s_{10} \rangle$ and $L_2 = \langle S_2, A_2, T_2, s_{20} \rangle$ be LTSs. A binary relation $R \subseteq S_1 \times S_2$ is said to be a *simulation* (from L_1 to L_2) if $(s_1, s_2) \in R$ implies that, for all $a \in Act$,

if $s_1 \xrightarrow{a} s_1'$ for some s_1', then $s_2 \overset{\widehat{a}}{\Rightarrow} s_2'$ and $(s_1', s_2') \in R$ for some s_2'.

L_1 is *simulated* by L_2 if $(s_{10}, s_{20}) \in R$ for some simulation R.

Define a relation $\sqsubseteq$ by

$$\sqsubseteq = \cup\{R \subseteq S_1 \times S_2 \mid R : \text{a simulation}\}.$$

Then, $\sqsubseteq$ is the largest simulation. It is clear from the definition that L_1 is simulated by L_2 iff $s_{10} \sqsubseteq s_{20}$.

Definition 3 (Bisimulation)
Let $L_1 = \langle S_1, A_1, T_1, s_{10} \rangle$ and $L_2 = \langle S_2, A_2, T_2, s_{20} \rangle$ be LTSs. A binary relation $R \subseteq S_1 \times S_2$ is said to be a *bisimulation* if $(s_1, s_2) \in R$ implies that, for all $a \in Act$,

1) if $s_1 \xrightarrow{a} s_1'$ for some s_1', then $s_2 \overset{\widehat{a}}{\Rightarrow} s_2'$ and $(s_1', s_2') \in R$ for some s_2'.

2) if $s_2 \xrightarrow{a} s_2'$ for some s_2', then $s_1 \overset{\widehat{a}}{\Rightarrow} s_1'$ and $(s_1', s_2') \in R$ for some s_1'.

L_1 is bisimulated by L_2 if $(s_{10}, s_{20}) \in R$ for some bisimulation R.

In the same way as the simulation, define a relation $\approx$ by

$$\approx = \cup\{R \subseteq S_1 \times S_2 \mid R : \text{a bisimulation}\}.$$

Then, $\approx$ is the largest bisimulation. It is clear from the definition that L_1 is bisimulated by L_2 iff $s_{10} \approx s_{20}$.

2.2 Similarity of LTSs

To reuse the existing specifications, we need some criterion to retrieve the most appropriate specifications. In this section, we propose a concept of similarity between specifications as one of the criteria. We have already proposed a concept of similarity between two LTSs[7]. However, it has a constraint that every LTS has no loops. However, most LTSs generally have some loops, so our similarity isn't applicable in most case of the reuse.

Here, we extend our concept of similarity between LTSs to solve above problems.

2.2.1 Basic Concept

As mentioned above, there exist some relationships between LTSs such as bisimulation and simulation. However, they can't be used as criterion of similarity between LTSs in reuse process. For example, we consider LTSs in Fig. 2. From the point of human's view, it can be said that they are similar (or have common parts) each other. For the LTS of Fig. 2(a), however, it cannot have any relationship which has been defined, because it is clear that the LTS 2(a) and 2(b)(or 2(c)) have different observable behaviors and therefore bisimulation isn't the relation between them. On the other hand, the behavior of the LTS 2(a) is not included in the LTS 2(b), since the action d occurs after occuring the action c in 2(a), while cannot do so in 2(b). Thus, the LTSs 2(a) is not simulated by

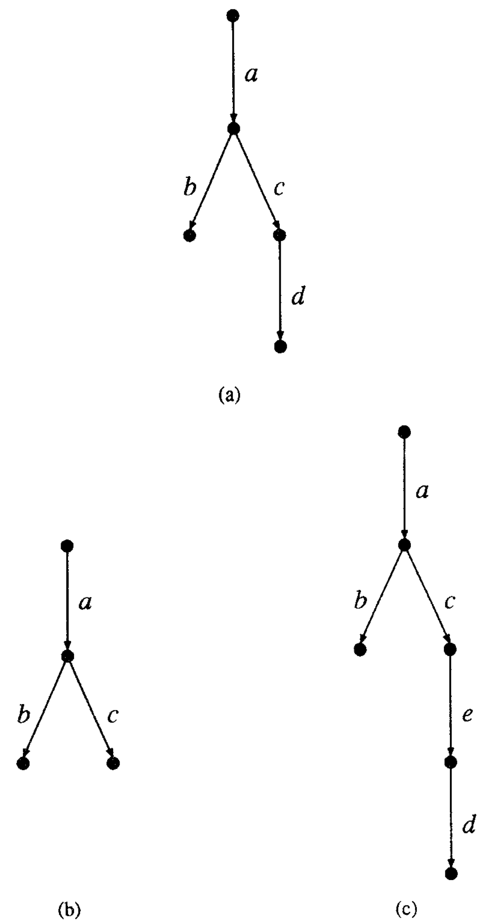

Figure 1: Concept of Similarity

the LTS 2(b), namely they don't have the simulation relationship for the LTS 2(a).

For this reason, we must define a new concept of relationship to be able to judge that the LTS 2(a) is similar to 2(b)(and 2(c)). Our concept of similarity between LTSs is defined as the meaning that they have some common pattern of occurence of actions if some actions are ignored. For example, In Fig 2, LTS (a) and (b) have a common part with the actions a, b and c, since each behavior of them includes that action a occurs first and the occurrence of action b or c follows it. On the other hand, LTS (a) and (c) also have a common part for the action a, b, c and d by ignoring action e. From our concept, more two LTSs have a common part, it is said that more similar they are. Later, we define our cocept formally and show some characteristics.

2.2.2 Definition and Property

The definition of similarity is based on transition relations of LTSs. To define the similarity, we use some notations for LTSs in Table 1.

notation	meaning
$A(t)$	if $t = s \xrightarrow{a} s'$, then $A(t) = a$
$Bef(t)$	if $t = s \xrightarrow{a} s'$, then $Bef(t) = s$
$Aft(t)$	if $t = s \xrightarrow{a} s'$, then $Aft(t) = s'$
$t \xrightarrow{r} t'$	if $t = s \xrightarrow{a} s_1$ and $t' = s_2 \xrightarrow{a'} s_3$, then $s \xrightarrow{a} s_1 \xrightarrow{r} s_2 \xrightarrow{a'} s_3$
$s \overset{T}{\leadsto} t$	for some $t_1, \cdots t_n \in T$, $s = Bef(t_1)$ and $t_1 \xrightarrow{\epsilon} \cdots \xrightarrow{\epsilon} t_n \xrightarrow{\epsilon} t$

Table 1: Notations for LTSs

We begin with definitions of the relations between two transitions, called as *sequential relation* and *parallel relation* as follows:

Definition 4 (Relations between Transitions) Let T be a transition relation and $t, t' \in T$.

1) For some T' such that $T \cap T' = \emptyset$, if $Aft(t) \overset{T'}{\leadsto} t'$, then we refer to this relation as *sequential relation* and write $t \rhd_T t'$.

2) For some state $s \in S$ and T', T'' such that $T \cap T' = T \cap T'' = \emptyset$, if $s \overset{T'}{\leadsto} t$ and $s \overset{T''}{\leadsto} t'$, then we refer to this relation as *parallel relation* and write $t \diamond_T t'$.

The above two relations indicates the ordering of two transitions in the same transition relation. *Sequential* relation means that one transition occurs following by the other transition in the same transition relation, while some transitions may occur between them. *Parallel* relation means that two transitions alternatively occur in the same transition relation.

Next, We introduce the similarity of actions, represented as a set λ of pairs of actions. However, we don't mention how it is given now, but later.

Now, we show a definition of the similarity of transition relations, which decides the similarity between two LTSs (specifications).

Definition 5 (Similarity of Transition Relations) Let T_1 and T_2 be transition relations. A binary relation $\mathcal{Q} \subseteq T_1 \times T_2$ is said to be a *similarity of transition relations* (from L_1 to L_2) if the following conditions are satisfied:
For all $(t_1, t_2) \in \mathcal{Q}$,

1) If $A(t_1) \neq \tau$, then $(A(t_1), A(t_2)) \in \lambda$. Otherwise, $A(t_1) = A(t_2) = \tau$ or $A(t_2) = \varepsilon$.

2) If $t_1 \rhd_{P_1(\mathcal{Q})} t'_1$ for some $t'_1 \in P_1(\mathcal{Q})$, then $t_2 \rhd_{P_2(\mathcal{Q})} t'_2$ and $(t'_1, t'_2) \in \mathcal{Q}$ for some transition $t'_2 \in T'_2$.

3) If $t_1 \diamond_{P_1(\mathcal{Q})} t'_1$ for some $t'_1 \in P_1(\mathcal{Q})$, then $t_2 \diamond_{P_2(\mathcal{Q})} t'_2$ and $(t'_1, t'_2) \in \mathcal{Q}$ for some transition $t'_2 \in T'_2$ or $(t'_1, t_2) \in \mathcal{Q}$.

4) If $s_{10} \overset{T_1 - P_1(\mathcal{Q})}{\rightsquigarrow} t_1$, then $s_{20} \overset{T_2 - P_2(\mathcal{Q})}{\rightsquigarrow} t_2$.

where,

- $P_1(\mathcal{Q}) = \{t \in T_1 \mid \text{for some } t' \in T_2, (t, t') \in \mathcal{Q}\}$
- $P_2(\mathcal{Q}) = \{t' \in T_2 \mid \text{for some } t \in T_1, (t, t') \in \mathcal{Q}\}$

[Example]
Let us consider the two LTSs given in Fig. 2(a). We assume the following sets:

- $\mathcal{Q}_1 = \{(s_0 \xrightarrow{a} s_1, s'_0 \xrightarrow{a} s'_1), (s_1 \xrightarrow{b} s_2, s'_2 \xrightarrow{b} s'_4),$
  $(s_3 \xrightarrow{d} s_0, s'_3 \xrightarrow{d} s'_5)\}$
- $\mathcal{Q}_2 = \{(s_0 \xrightarrow{a} s_1, s'_0 \xrightarrow{a} s'_1), (s_1 \xrightarrow{b} s_2, s'_2 \xrightarrow{b} s'_4),$
  $(s_1 \xrightarrow{c} s_3, s'_1 \xrightarrow{c} s'_2)\}$

For $\mathcal{Q}_1$ (Fig. 2(b)), actions of corresponding transitions are the same, and if there is any relationship between transitions in $P_1(\mathcal{Q}_1)$, then there exists the same relationship between the corresponding transitions in $P_2(\mathcal{Q}_1)$. For example,

$$(s_0 \xrightarrow{a} s_1) \triangleright_{P_1(\mathcal{Q}_1)} (s_1 \xrightarrow{b} s_3)$$
$$\Rightarrow (s'_0 \xrightarrow{a} s'_1) \triangleright_{P_2(\mathcal{Q}_1)} (s'_1 \xrightarrow{c} s'_3).$$

These satisfy the conditions of Def. 5, and if any pair of transitions is added to $\mathcal{Q}_1$, the conditions of Def. 5 aren't satisfied. Hence, $\mathcal{Q}_1$ is a similarity of transition relations. However, $\mathcal{Q}_2$ (Fig. 2(c)) isn't a similarity of transition relations because $(s_0 \xrightarrow{a} s_1) \triangleright_{P_1(\mathcal{Q})} (s_1 \xrightarrow{b} s_2)$, but not $(s'_0 \xrightarrow{a} s'_1) \triangleright_{P_2(\mathcal{Q})} (s'_2 \xrightarrow{b} s'_4)$.

Definition 6 Let $\mathcal{Q}$ be a similarity of transition relations from L_1 to L_2,

1) $\mathcal{Q}$ *covers* L_1 if $P_1(\mathcal{Q}) = T_1$.

2) $\mathcal{Q}$ *covers* L_2 if $P_2(\mathcal{Q}) = T_2$.

Theorem 1 Let L_1 and L_2 be LTSs. If there exists a simulation $R \subseteq S_1 \times S_2$, then there exists a similarity of transition relations $\mathcal{Q} \subseteq T_1 \times T_2$ which covers L_1.

Proof:
We assume that R is a maximal simulation from L_1 to L_2. Define a maximal binary relation $\mathcal{Q} \subseteq T_1 \times T_2$ by:

$$\mathcal{Q} = \{(t_1, t_2) \mid (Aft(t_1), Aft(t_2)), (Bef(t_1), Bef(t_2)) \in R \text{ and } A(t_1) \neq \varepsilon\}.$$

Note that this $\mathcal{Q}$ satisfies the first condition of a similarity of transition relations, namely the condition about action.

Now, we will show that R is a similarity of transition relations from L_1 to L_2. Let $(t_1, t_2) \in \mathcal{Q}$. Suppose $t'_1 \in P_1(\mathcal{Q})$.

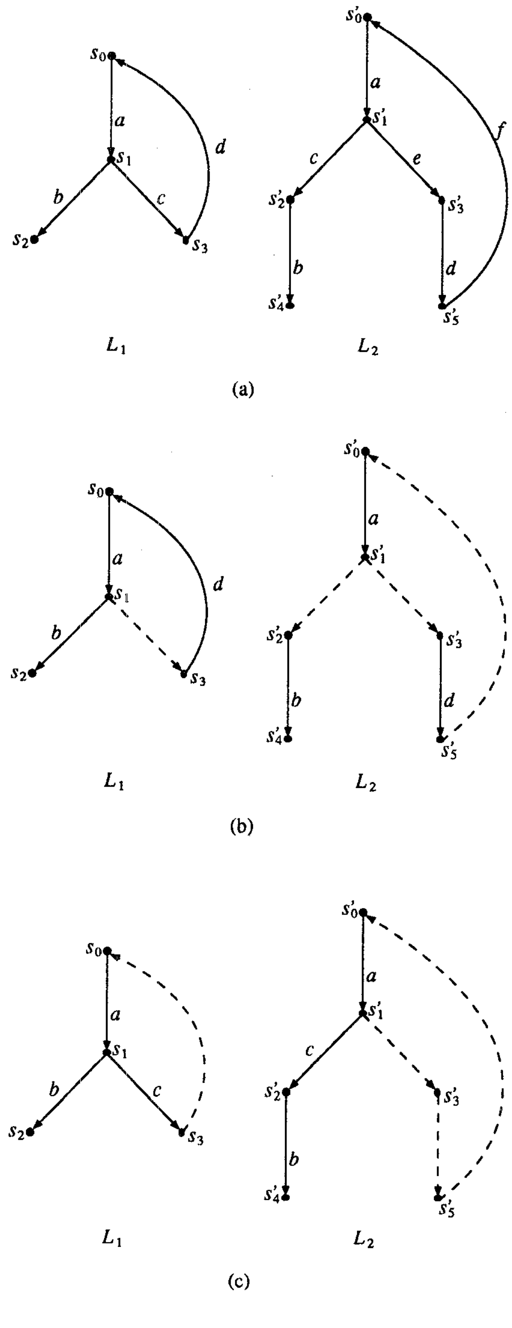

Figure 2: Example of Similarity

Case 1: If $t_1 \triangleright_{P_1(\mathcal{Q})} t'_1$, then there exists $t'_2 \in P_2(\mathcal{Q})$ such that $(t'_1, t'_2) \in \mathcal{Q}$ and $t_2 \triangleright_{P_2(\mathcal{Q})} t'_2$, because $Aft(t_1) = Bef(t'_1)$ and if there exists any transition from $Aft(t_1)$ by action a, there exists some transition $Aft(t_2) \stackrel{\widehat{a}}{\Rightarrow} s'_2$ by the simulation R.

Case 2: If $t_1 \diamond_{P_1(\mathcal{Q})} t'_1$, then there exists $t'_2 \in P_2(\mathcal{Q})$ such that $(t'_1, t'_2) \in \mathcal{Q}$ and $t_2 \diamond_{P_2(\mathcal{Q})} t'_2$, because $Bef(t_1) = Bef(t'_1)$ and if there exists any transition from $Bef(t_1)(Bef(t_2))$ by action a, there exists some transition $Bef(t_2) \stackrel{\widehat{a}}{\Rightarrow} s'_2$ by the simulation R.

Hence, $\mathcal{Q}$ is clearly a similarity of transitions from L_1 to L_2 covering L_1. □

Theorem 2 Let L_1 and L_2 be LTSs. Suppose there exists a similarity of transition relations $\mathcal{Q} \subseteq T_1 \times T_2$ which covers L_1. Let $L'_2 = \langle S_2, A_2, T'_2, s_{20} \rangle$ be an LTS obtained from $L_2 = \langle S_2, A_2, T_2, s_{20} \rangle$ by replacing a transition $t_2 = s_2 \xrightarrow{a} s'_2 \in T_2$ such that $(t_1, t_2) \notin \mathcal{Q}$ for all $t_1 \in T_1$ by $s_2 \xrightarrow{\tau} s'_2$. Then, L_1 is simulated by L'_2.

Proof:
Define a binary relation $R \subseteq S_1 \times S_2$ by

$$R = \{(Aft(t_1), Aft(t_2) \mid (t_1, t_2) \in \mathcal{Q}\} \cup \{(s_{10}, s_{20})\}.$$

Now, we will show that R is a simulation from L_1 to L'_2. Let $(s_1, s_2) \in R$. Suppose $t'_1 = s_1 \xrightarrow{b} s'_1$ for some $b \in Act$ and for some s'_1. From the assumption, there exists $t'_2 = s''_2 \xrightarrow{b'} s'_2$ such that $(t'_1, t'_2) \in \mathcal{Q}$, where $b = b'$ or $b = \tau$ and $b' = \varepsilon$. Thus, $(s'_1, s'_2) \in R$, and $\xrightarrow{b'} = \stackrel{\widehat{b}}{\Rightarrow}$.

Case 1: If there exist transitions t_1 and t_2 such that $Aft(t_1) = s_1$ and $Aft(t_2) = s_2$, then $t_1 \triangleright_{P_1(\mathcal{Q})} t'_1$ and $t_2 \triangleright_{p_2(\mathcal{Q})} t'_2$ because of the definition of similarity. Therefore, $s_2 \xrightarrow{r} s''_2 \xrightarrow{b'} s'_2$ for some $r \in Act^*$. Since any action with a transition t such that $t \notin P_2(\mathcal{Q})$ are replaced by τ from assumption, $r = \tau^*$. Hence, $s_2 \stackrel{b'}{\Rightarrow} s'_2$.

Case 2: If there doesn't exist transitions t_1 and t_2 such that $Aft(t_1) = s_1$ and $Aft(t_2) = s_2$, then s_1 and s_2 are the initial states of L_1 and L'_2, respectively. By reachability condition, $s_2 \xrightarrow{r} s''_2 \xrightarrow{b'} s'_2$ for some $r \in Act^*$. There must exist at least one transition t'_2, where $s_2 \stackrel{T'}{\rightsquigarrow} t'_2$ for some T' such that $T' \cap P_2(\mathcal{Q}) = \emptyset$. Because, if there exists any transition t''_2 such that $s_2 \stackrel{T'}{\rightsquigarrow} t''_2$ and $t''_2 \triangleright_{P_2(\mathcal{Q})} t'_2$, then there exists t''_1 such that $(t''_1, t''_2) \in \mathcal{Q}$, but $t'_1 \diamond_{P_1(\mathcal{Q})} t''_2$. Therefore, $s_2 \xrightarrow{r} s''_2 \xrightarrow{b'} s'_2$ for some $r \in Act^*$. Since any action with a transition t such that $t \notin P_2(\mathcal{Q})$ are replaced by τ from assumption, $r = \tau^*$. Hence, $s_2 \stackrel{b'}{\Rightarrow} s'_2$.

Since $\xrightarrow{b'} = \stackrel{\widehat{b}}{\Rightarrow}$, $s_2 \stackrel{\widehat{b'}}{\Rightarrow} s'_2$. Hence, for all $(s_1, s_2) \in R$, if $s_1 \xrightarrow{a} s'_1$, then $s_2 \stackrel{\widehat{a}}{\Rightarrow} s'_2$ and $(s'_1, s'_2) \in R$ for some s'_2. This clearly shows that R is a simulation. □

Corollary 1 Let L_1 and L_2 be LTSs. L_1 is bisimulated by L_2 if there exists a similarity of transition relations $\mathcal{Q}$ such that

1) $\mathcal{Q}$ covers L_1;

2) $\mathcal{Q}$ covers L_2. □

Theorem 3 Let L_1 and L_2 be LTSs. Suppose there exists a similarity of transition relations $\mathcal{Q} \subseteq T_1 \times T_2$. Let $L'_1 = \langle S_1, A_1, T'_1, s_{10} \rangle$ be an LTS obtained from $L_1 = \langle S_1, A_1, T_1, s_{10} \rangle$ by replacing a transition $t_1 = s_1 \xrightarrow{a} s'_1 \in T_1$ such that $(t_1, t_2) \notin \mathcal{Q}$ for all $t_2 \in T_2$ by $s_1 \xrightarrow{\tau} s'_1$, and L'_2 be an LTS obtained from L_2 in the similar way. Then, L'_1 is simulated by L'_2.

Proof:
Define a binary relation $R \subseteq S_1 \times S_2$ by

$$R = \{(Aft(t_1), Aft(t_2) \mid (t_1, t_2) \in \mathcal{Q}\} \cup \{(s_{10}, s_{20})\}.$$

Now, we will show that R is a simulation from L'_1 to L'_2. Let $(s_1, s_2) \in R$. Suppose $t'_1 = s_{13} \xrightarrow{b} s_{12}$ such that $s_1 \xrightarrow{\tau^*} s_{13} \xrightarrow{b} s_{12}$ for some $b \in Act$ and for some s_{12} and s_{13}. From assumption, there exists $t'_2 = s_{23} \xrightarrow{b'} s_{22}$ such that $(t'_1, t'_2) \in \mathcal{Q}$, where $b = b'$ or $b = \tau$ and $b' = \varepsilon$. Thus, $(s_{12}, s_{22}) \in R$, and $\xrightarrow{b'} = \stackrel{\widehat{b}}{\Rightarrow}$.

The rest of the proof is similar to one of Theorem 2. □

This theorem says if there is a similarity from L_1 to L_2, a part or all of behavior of L_1 is embedded into L_2 in a graph theoretic way.

Finally, we define the degree of similarity.

Definition 7 (Degree of Similarity) Let L_1 and L_2 be LTSs. Given a function $\sigma : A_1 \times A_2 \to [0, 1]$, the *degree of similarity* $\rho(L_1, L_2)$ between L_1 and L_2 is defined by:

$$\rho(L_1, L_2) = \frac{\max\{\|\mathcal{Q}\|\}}{|T_1|}$$

where,

- $\mathcal{Q} \subseteq T_1 \times T_2$: similarity of transition relations and $((t_1, t_2), (t_1, t'_2) \in \mathcal{Q} \Rightarrow t_2 = t'_2)$;
- $|T_1|$ represents the number of transitions in T_1;
- $\|\mathcal{Q}\| = \sum\{\sigma(A(t_1), A(t_2)) \mid (t_1, t_2) \in \mathcal{Q}\}$.

The degree of similarity shows the criterion how an LTS contains the behavior of the other LTS. It is used to judge which of LTSs is similar to the requirements. In this definition, we consider the maximal value of

$\|\mathcal{Q}\|$, because there may be some similarities of transition relations between two LTSs. For example, let us consider two LTSs in Fig 2(a). Then, it is clear that we have two similarities of transition relations between them. One is the similarities for the actions a and c, and another is for the actions a, b and d.

2.3 Similarity of Actions

To use the similarity mentioned in Section 2.2, we have to consider the similarity of actions, λ. There are some approaches how it is given as follows:

a) The similarity of actions is given from the designer when he investigates the similarity between LTSs to reuse one of stored specifications.

b) It is initially stored in the knowledge base, and/or incrementally stored by reusing specifications.

c) It is given from two LTSs based on some criteria relative to actions, such as the name of action and temporal ordering of occurrence of actions.

Generally, a) and b) have some serious problems.

In approach a), the designer has to completely know about actions which occur in LTSs. While the name of action may denote the functionality of it, the designer must still have a heavy burden relative to it.

In approach b), it is difficult to initially give such knowledge to the support system, since the engineer who gives such knowledge must find all possible pairs of similar actions which can be used in reuse process. But, incremental storage of such knowledge can be used as the personal knowledge.

Here, we take the approach c). We employ the following criteria to construct the similarity of actions.

- name of action
- attribute of action
- the way of occurrence of action

As above mentioned, the name of action may denote the functionality of it, so we think comparison about the name is simple, but effective. However, it has a problem that the names of two actions may be different among the designers, or among the domain of application, even though they have the same functionality.

Therefore, we employ the other criteria, attribute of action and the way of occurrence of action. For the first criterion, we assume that an action is generally classified as either sending or receiving action to apply the domain of communication network especially, and we use this to judge the similarity of actions. For the second criterion, some actions in one LTS may not be corresponded to any action in another LTS. However, we consider that it can be said that an action in one LTS is similar to the action in another LTS if the patterns of their occurrence are the same way, and this is also used as one of the criterion.

We consider that the similarity of actions is determined while the degree of similarity between two LTSs is calculated. In the next section, we show the retrieval method of similar specifications using the concept of similarity.

3 Retrieval Mechanism of Similar Specifications

3.1 Retrieval Method

Now, we show the retrieval method of similar specifications, which consists of two steps:

STEP-1) For the LTS given as the user requirement, the degree of similarity between it and each existing LTS in the knowledge base is calculated.

STEP-2) In the high order of the degree of similarity, the LTSs are shown to the user.

Here, we show STEP-1 more detailed. Let L_1 be the LTS as the user requirement and L_2 be the LTS calculated the degree of similarity. Then, the step is as follows:

1-i) If there exist the actions with the same name in L_1 and L_2, then they can be corresponded each other and the pair of them can be included in the similarity of actions λ. At the first construction of the similarity of transition relations $\mathcal{Q}$, the similarity of actions λ is only used.

1-ii) The other uncorresponded action in L_1 can be corresponded to the action in L_2 when the following conditions are satisfied:

a) the attributes of them are the same, and

b) the pattern of occurrence of them, based on $\mathcal{Q}$, are the same.

The pair of them can be included in λ, and the similarity of transition relations is updated using λ.

1-iii) Using the constructed similarities of transition relations, the degree of similarity is calculated.

3.2 Example

For example, we consider two LTSs shown in Fig. 3, and we assume that L_1 is the LTS as the user requirement and L_2 is the existing LTS in the knowledge base. the symbol '?' prefixed to the actions represents "input" or "receiving", and the symbol '!' represents "output" or "sending". Let λ and $\mathcal{Q}$ be the empty sets. Now we show the calculation of the degree of similarity between L_1 and L_2.

There exists action d in L_1 and L_2, so the pair (d, d) of actions can be included in λ. The pair of transitions with action d as dashed line (1) in Fig. 3 is included in $\mathcal{Q}$. Next, the correspondence of actions with different names may be found. One correspondence is the pair (a, x), since the patterns of occurrence of them are the same, even if considering $\mathcal{Q}$. Then, (a, x) is included in λ and the pairs of transitions with action a and x as dashed line (2) in Fig. 3 is included in $\mathcal{Q}$. Moreover, the transitions with the action c and e in L_1 are

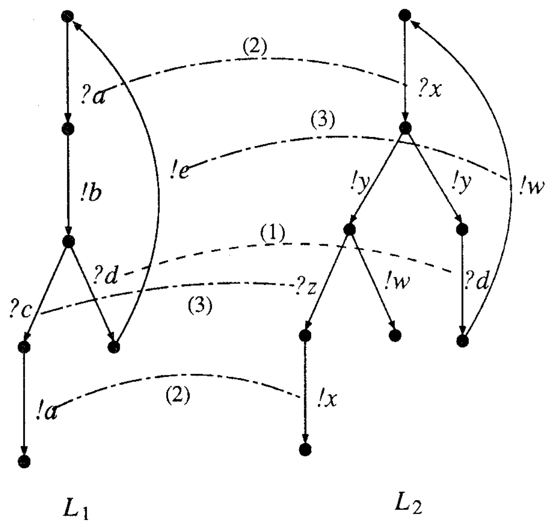

Figure 3: Example of Calculation

corresponded to the ones with the action z and w as line (3) in Fig. 2, respectively, based on the attributes of actions and the pattern of occurrence and $\mathcal{Q}$. In this way, we can obtain a similarity of transition relations $\mathcal{Q}$ between L_1 and L_2. In the shown process, the actions a, c, d and e are corresponded to the actions in L_2, but there exist the other correspondence about the actions in L_1. For example, the actions a, b, d and e in L_2 can be corresponded to the actions x, y, d and w, respectively. Using some similarities of transition relations between L_1 and L_2, the degree of similarity is calculated as follows:

$$\rho(L_1, L_2) = \frac{\text{the number of corresponded transitions in } L_1}{\text{the number of transitions in } L_1} = 5/6 = 0.867$$

4 Evaluation

In this subsection, we evaluate the similarity and the retrieval method shown above. First, we compare our similarity with some relations between LTSs which exist as mentioned in Section 2. In Table 2, We show the comparison of our similarity with them as a criterion to judge the specifications to be reused.

a) Independency of Application
To find the relation of bisimulation or simulation between two LTSs, each corresponding transitions has to have the same action by the definitions. This leads that two compared LTSs need to belong to the same domain of application, and causes the limitation of reuse.

In contrast, our similarity has more weak condition for actions that the transitions with the same action in one LTS are corresponded with the ones in another LTS which have the same action that may be different from that action. Hence, our similarity is applicable to reuse specifications over the application domains.

b) Easiness of Matching
As mentioned in Section 3, simulation means that the behavior of one LTS is included in another LTS, and bisimulation means that observable behaviors of two LTSs are completely same. Namely, given the incomplete requirement of the target system, there are few LTSs which are judged as the similar one, because because the target system isn't usually included in and same to the existing one.

On the other hand, since our similarity have more weak conditions for the corresponding transitions than the existing relations, our similarity makes it more easy to correspond two LTSs.

c) Complexity
Certainly, the complexity of the calculation using our similarity is larger than the one using the other existing relations, but it doesn't matter if the computer with the appropriate power is available.

5 Conclusion

In this paper, We discussed about the reuse of specifications in communication software development. We pointed out some problems for our concept of similarity in [7] and proposed one of the approaches to overcome it. In this approach, the definition of similarity is extended in order for our concept of similarity to be applicable to the LTS with some loops, and the similarity of actions depends on not only the name of each action, but also the attributes of actions and the pattern of occurrence of actions. Therefore, it enables the designer to retrieve the similar LTSs without the similarity of actions, and the similarity of actions is found as a result of the retrieval. By reusing the existing specifications retrieved using our similarity, it is expected that the designer can obtain the similar specifications effectively, and the productivity and quality in specification process will be largely improved.

We have some future works, such as implementation and evaluation of the specification environment to investigate the effectiveness of it. Now we are implementing our concept as a part of the specification environment SERA[9] on X window system. Figure 3 shows an example of specification process using the specification environment SERA. In this figure, the designer first describes the specification using LTS as his initial requirement. Using this LTS, the similar LTSs are retrieved from the knowledge base by calculating the degree of similarity, and shown to the designer. If the designer finds the most suitable LTS among them, he can select it, rename the names of actions.

	Bisimulation	Simulation	Similarity
a) Independency of Application Domain	×	×	○
b) Easiness of Matching	×	△	○
c) Complexity	○	△	△

Table 2: Comparison for Relations of LTSs

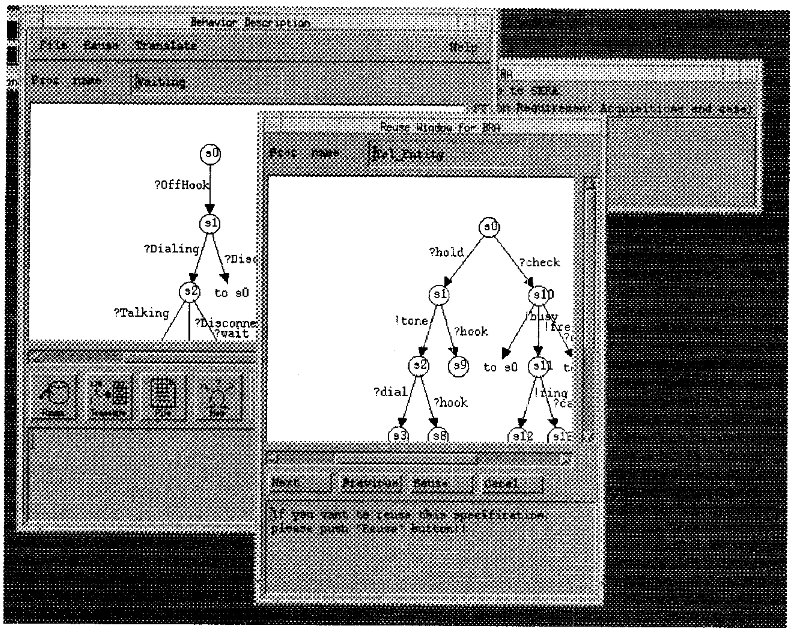

Figure 4: Example of Specification Environment

References

[1] N. Shiratori, H. Kaminaga, K. Takahashi and S. Noguchi, "A Verification Method for LOTOS Specification and Its Application," In *Protocol Specification, Testing and Verification, IX*, North-Holland, pp. 59 – 70, 1990.

[2] J. A. Manas and T. D. Miguel-More, "From LOTOS to C," In *Proc. of FORTE 88*, pp. 79–84, 1988.

[3] P. V. Eijk, "LOTOS Tools based on the Cornell Synthesizer Generator," In *Protocol Specification, Testing and Verification IX*, North-Holland, pp. 275–286, 1990.

[4] CCITT, "Specification and Description Language," Recommendation Z.100, 1989.

[5] ISO, "Information Processing Systems — Open Systems Interconnection — Estelle — A formal description technique based on an extended state transition model," ISO 9074, 1989.

[6] ISO, "Information Processing Systems — Open Systems Interconnection — LOTOS — A formal description technique based on the temporal ordering of observational behaviour," ISO 8807, 1989.

[7] U. Yamamoto, A. Togashi and N. Shiratori, "A Support Method for Specification Process based on LTSs," *IEICE Trans. Fundamentals*, Vol. E77-A, No. 10, pp. 1656-1662, 1994.

[8] R. Milner, *Communication and Concurrency*, Prentice Hall, 1989.

[9] E.S. Lee, U. Yamamoto and N. Shiratori, "Requirement Acquisition and Cases based Specification Environment, SERA," In *Proceedings of The 9th International Conference on Information Networking(ICOIN-9)*, pp279-284, 1994.