Methodologies for the Description of System Requirements and the Derivation of Specifications (Extended Abstract)

Atsushi Togashi*　　Nobuyuki Usui, Kukhwan Song, Norio Shiratori†

Dep. of Comp. Sci., Shizuoka Univ.　　{RIEC, GSIS}, Tohoku Univ.

Abstract

Methodologies for the description of system requirements and the synthesis of formal specifications from user requirements are presented. We will specifically deal with the issues (1) mathematical treatment of system requirements and their relationship with formal specifications represented as state transition systems, (2) sound and complete systems, i.e. standard systems, (3) derivation of standard systems from user requirements, and (4) some discussions on partial logical Petri Nets and Production systems.

1 Introduction

For a complex and sophisticated system, operational descriptions might be too tedious to handle for rapid prototyping and analysis of a system's behavior. In such cases, it is more convenient to express the system on a higher level, somehow in a functional manner. This approach yields formal specifications that emphasize the system's general behavioral properties rather than its operational details. Moreover, it has a practical significance if the desired description can be derived or synthesized, in a systematic way from the user requirements on system functions.

This paper proposes new methodologies for the description of system requirements and the synthesis of formal specifications from user requirements. The formal specifications can be taken as models of the system requirements. More generally, the main objective is to be able to derive an implementable or operational system description from a given high-level description on system functions. The proposed methodology can be fully automated, hence may/can improve both productivity and quality of system development. We have implemented a support system based on our approach and applied several practical system designs such as a telephone service, a communication protocol, a CATV system, etc.

In the literature of communicating systems, Formal Description Techniques (FDT), e.g. SDL [3], Estelle [1] and LOTOS [4], have been proposed as high-level specification languages. The conventional state machine oriented approaches such as SDL and Estelle and algebraic approach such as LOTOS are suitable for the purpose of description and investigation of the total behavior of systems. But, these approaches might be not suitable for rapid prototyping and flexible software development. Because we must enumerate and/or determine all system behaviors from an early stage of system design. Our objective is to give theoretical foundations and proposal of a flexible approach on the synthesis of formal specifications from user requirements written in an early stage of system design.

From objectives, our work has some connection with an STR (State Transition Rule) method, which is a specification method based

*Department of Computer Science, Shizuoka University, 5-1, 3 Johoku, Hamamatsu 432, Japan; Phone: +81-53-478-1463, Fax: +81-53-475-4595; togashi@cs.inf.shizuoka.ac.jp

†{Research Institute of Electrical Communication, Graduate School of Information Sciences} Tohoku University, 2-1-1, Katahira, Aoba, Sendai 980, Japan; Phone: +81-22-217-5454, Fax: +81-22-263-9848

on a production system proposed by Hirakawa and Takenaka in [8]. But, the methodology proposed here differs from their approach mainly in theoretical discussions such as sound & completeness and formal treatment, rather than practical methodology for description and use. Another related work is a synthesis of communicating processes from temporal logic specification by Manna and Wolper in [?]. Their approach is based on tableau-like method and completely different form ours from technical point of view. Besides those works, no other related works could be found in the literature.

The outline of this paper is as follows: In section 2 after giving preliminaries, we deal in detail with the issue of requirements and formal specifications. In section 3, we discuss the key notions, soundness and completeness. Section 4 provides an equivalent transformation on requirements with the result of determinacy for the resulting transition systems. Section 5 gives an automatic transformation technique from user requirements to formal specifications followed by the discussions in section 6 and the conclusion in section 7.

2 Requirements and Formal Specifications

Let $\mathcal{P}$ be a set of *atomic propositions*. Each atomic proposition describes a specific property of the intended system under the target of design. A *partial interpretation* I is a partial function $I : \mathcal{P} \rightarrow \{\textbf{true}, \textbf{false}\}$, where **true** and **false** are the truth values of propositions. If the truth value of a proposition f under I is defined to be **true** then we say that I *satisfies* f, denoted by $I \models f$. $I \not\models f$ denotes that the truth value of f is defined to be **false** and we say I *does not satisfy* f. Theses can be defined inductively as follows:

(1) $I \models A$ ($I \not\models A$) if I is defined on A and $I(A) = \textbf{true}$ ($I(A) = \textbf{false}$), where $A \in \mathcal{P}$.
(2) $I \models \neg f$ ($I \not\models \neg f$) if $I \not\models f$ ($I \models f$).
(3) $I \models f \wedge g$ ($I \not\models f \wedge g$) if $I \models f$ and $I \models g$ ($I \not\models f$ or $I \not\models g$).
(4) $I \models f \vee g$ ($I \not\models f \vee g$) if $I \models f$ or $I \models g$ ($I \not\models f$ and $I \not\models g$).

For propositions f and g, $f \Rightarrow g$ denotes the assertion that for any partial interpretation I, $I \models f$ implies $I \models g$, i.e. $\forall I.\, I \models f \supset I \models g$.

Definition 2.1 Let f and g be propositions.

(1) f is *consistent* if $I \models f$ for some partial interpretation I.
(2) f is *inconsistent* if f is not consistent.
(3) f is *dependent* on g if either $g \Rightarrow f$ (in positive) or $g \Rightarrow \neg f$ (in negative).
(4) f is *independent* of g if f is not dependent on g. □

Let γ, γ' be consistent conjunctions of literals. It is clear from the definition that $\gamma \Rightarrow \gamma'$ iff $L(\gamma) \supset L(\gamma')$, where $L(\gamma)$ denotes the set of literals appearing in γ. This implies the following proposition.

Proposition 2.1 *Let γ be a consistent conjunction of literals. An atomic proposition A is independent of γ iff A does not appear in γ at all neither in positive nor negative. The negative literal $\neg A$ is independent of γ iff A is independent of γ.* □

A system can be essentially specified by its fundamental functions and their related constraints for execution. To be more precise, a system function may be invoked by a specific input provided that its pre-condition to be satisfied before execution can hold in the current state. Then, the function is executed, possibly producing some appropriate output. After the execution the current state is changed into the new one. In the new state, another functions (including the same function as well) can be applicable. Taking account of this intuition of system specifications, a function requirement is formally defined in the next definition.

Definition 2.2 A *function requirement* is a tuple $\rho = \langle id, a, f_{in}, o, f_{out} \rangle$, where

(1) id is a *name* of the function;
(2) a is an *input symbol* of the function;
(3) f_{in} is a *pre-condition* of the function to be satisfied before execution, which is represented as a consistent proposition using atomic propositions in $\mathcal{P}$;

(4) o is an *output symbol* of the function;

(5) f_{out} is a *post-condition* of the function to be satisfied after execution, which is represented as a consistent conjunction of literals by atomic propositions in $\mathcal{P}$. □

For simplicity, in what follows we omit the names and the output symbols from the description of function requirements because they do not play the central roles on the theoretical treatment in this paper. A function requirement $\rho = \langle a, f_{in}, f_{out} \rangle$ is often abbreviated as $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$.

Definition 2.3 A *system requirement* is a pair $\mathcal{R} = \langle R, \gamma_0 \rangle$, where R is a set of function requirements and γ_0 is an *initial condition* represented as a consistent conjunction of literals in $\mathcal{P}$. □

In this paper, state transition systems are considered as formal specifications. In the literature, a state transition system is an underling structure of Formal Description Techniques, e.g. SDL [3], Estelle [1] and LOTOS [4], and used to give the operational semantics of concurrent processes in process calculi [9], based on the paradigm of SOS (Structural Operational Semantics) by Plotkin [11].

Definition 2.4 A *state transition system* is a quadruple $M = \langle Q, \Sigma, \rightarrow, q_0 \rangle$, where Q is a set of *states*, Σ is a set of input symbols, $\rightarrow$ is a *transition relation* defined as $\rightarrow \subset Q \times \Sigma \times Q$, and q_0 is an *initial state*. □

The transition relation defines the dynamical change of states as input symbols may be read. For $(p, a, q) \in \rightarrow$, we normally write $p \stackrel{a}{\rightarrow} q$. Thus, the transition relation can be written as $\rightarrow = \{\stackrel{a}{\rightarrow} \mid a \in \Sigma\}$. $p \stackrel{a}{\rightarrow} q$ may be interpreted as "in the state p if a is input then the state of the system moves to q". Now, we assume that for an atomic proposition A and for a state $q \in Q$ it is pre-defined whether or not A holds (is satisfied) in q if the truth value of A in q is defined. $q \models A$ indicates that the truth value of A in q is defined and A holds in q. Let define the partial interpreter associated with a state q in M, denoted by $I(q)$, in such a way that

$$I(q)(A) = \begin{cases} \textbf{true} & \text{if } q \models A \\ \textbf{false} & \text{if } q \not\models A \ (q \models \neg A) \\ \text{undef.} & \text{otherwise} \end{cases}$$

for all atomic propositions A. Let $Sat(q)$ denote the set of all literals l such that the truth value of a literal l is defined in q and $q \models l$.

Proposition 2.2 *$q \models f$ iff f is implied from $Sat(q)$, $Sat(q) \vdash f$, for all propositions f.*

Proof: The proof is by structural induction on propositions f. □

Two states p and q in M are *logically equivalent* iff $I(p) = I(q)$. A transition system M is *logically reducible* if there exist distinct logically equivalent states in M. Otherwise, the system is *logically irreducible*. To the rest of this paper, stated otherwise, a transition system means a logically irreducible system. Thus, $p = q$ iff $I(p) = I(q)$ $(Sat(p) = Sat(q))$.

3 Soundness and Completeness

Definition 3.1 A state transition $t = (p \stackrel{a}{\rightarrow} q)$ *satisfies* (is *correct w.r.t.*) a function requirement $\rho : f_{in} \stackrel{b}{\Rightarrow} f_{out}$, denoted as $t \models \rho$, if the following conditions hold:

(1) $p \models f_{in}$, $a = b$, and $q \models f_{out}$.

(2) The partial interpretations $I(p)$ and $I(q)$ are identical if atomic propositions independent of f_{out} are only concerned. □

The condition (1) means the precondition and the postcondition must hold in the current state and the next state, respectively. The condition (2) states that for an atomic proposition A independent of f_{out}, $p \models A \iff q \models A$. This means that the truth value of independent atomic propositions *w.r.t.* the postcondition remain unchanged through the transition.

Example 3.1
Consider the requirement description $\mathcal{R}_1 = \langle \{\rho_1 : A \stackrel{a}{\Rightarrow} \neg A, \rho_2 : B \stackrel{b}{\Rightarrow} A\}, A \wedge B \rangle$ and the transition system M_1 given in (a) in Figure 1. Now, consider the transition $t_1 = \langle q_0 \stackrel{a}{\rightarrow} q_1 \rangle$ and the function requirement $\rho_1 : A \stackrel{a}{\Rightarrow} \neg A$. Since $q_0 \models A$ and $q_1 \models \neg A$ the condition (1) in Definition 3.1 holds for t_1 *w.r.t.* ρ_1. The atomic proposition independent of $\neg A$ is B. Since the truth values of B in q_0, q_1 are defined

and $q_0, q_1 \models B$ the condition (2) in Definition 3.1 holds. Thus, the transition t_1 satisfies the function requirement ρ_1. In the exactly same way, we can easily check that the transitions $q_0 \xrightarrow{b} q_0$, $q_1 \xrightarrow{b} q_0$ satisfy the function requirement $\rho_2 : B \overset{b}{\Rightarrow} A$. □

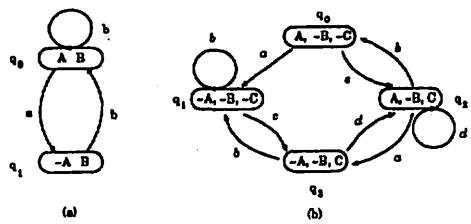

Figure 1: Transition Systems M_1 and M_2

Example 3.2 As a more involved example, let us consider the requirement

$$\mathcal{R}_2 = (\{\ \rho_1 : A \overset{a}{\Rightarrow} \neg A \wedge \neg B,\ \rho_2 : \neg A \wedge \neg B \vee A \wedge C \overset{b}{\Rightarrow} \neg C,\ \rho_3 : \neg C \overset{c}{\Rightarrow} C,\ \rho_4 : C \overset{d}{\Rightarrow} A\},\ A \wedge \neg B \wedge \neg C)$$

and the transition system M_2 given (b) in Figure 1. In the same way as in Example 3.1, it is checked that:

- the transition $q_0 \xrightarrow{a} q_1$, $q_2 \xrightarrow{a} q_3$ satisfy ρ_1;
- the transition $q_1 \xrightarrow{b} q_1$, $q_2 \xrightarrow{b} q_0$, $q_3 \xrightarrow{b} q_1$ satisfy ρ_2;
- the transition $q_0 \xrightarrow{c} q_2$, $q_1 \xrightarrow{c} q_3$ satisfy ρ_3;
- the transition $q_2 \xrightarrow{d} q_2$, $q_3 \xrightarrow{d} q_2$ satisfy ρ_4.

□

Let γ be a consistent conjunction of literals. We define a partial interpretation $I(\gamma)$ based on γ by

$$I(\gamma)(A) \begin{cases} \textbf{true} & \text{if } A \text{ appears positive in } \gamma \\ \textbf{false} & \text{if } A \text{ appears negative in } \gamma, \\ \text{undef.} & \text{otherwise} \end{cases}$$

for all atomic propositions A.

Definition 3.2 A state transition system $M = \langle Q, \Sigma, \rightarrow, q_0 \rangle$ is *sound* with respect to a requirement description $\mathcal{R} = \langle R, \gamma_0 \rangle$ if the following conditions are satisfied:

(1) $I(q_0) = I(\gamma_0)$;
(2) for any transition t in M there exists a function requirement $\rho \in R$ such that $t \models \rho$. □

The transition systems M_1 and M_2 are sound with respect to the requirement description $\mathcal{R}_1$ and $\mathcal{R}_2$, respectively.

Definition 3.3 Let $M = \langle Q, \Sigma, \rightarrow, q_0 \rangle$, $M' = \langle Q', \Sigma, \rightarrow', q_0' \rangle$ be state transition systems in common input symbols. A *homomorphism* from M into M' is a mapping $\xi : Q \rightarrow Q'$ such that

(1) $\xi(q_0) = q_0'$.
(2) if $p \xrightarrow{a} q$ in M, then $\xi(p) \xrightarrow{a} \xi(q)$ in M'.
(3) $p \models f$ implies $\xi(p) \models f$ for all states p in M and propositions f. □

The third condition in the above definition can be equivalently relaxed:

(3') $p \models l$ implies $\xi(p) \models l$ for all states p in M and for all literals l.

If a homomorphism $\xi : M \rightarrow M'$ is a bijection and the inverse function ξ^{-1} is a homomorphism from M' to M, then ξ is called an *isomorphism*. If there is an isomorphism from M to M', then M and M' are *isomorphic*.

Definition 3.4 Let M be a sound state transition system with respect to $\mathcal{R}$. M is called *complete* with respect to $\mathcal{R}$ if, there is a homomorphism $\xi : M' \rightarrow M$ for every sound state transition system M' with respect to $\mathcal{R}$. □

Definition 3.5 A sound and complete transition system with respect to $\mathcal{R}$ is called a *standard system* (*model*) of $\mathcal{R}$. □

Theorem 3.1 *Let M, M' be standard systems of $\mathcal{R}$, then M and M' are isomorphic* [?]. □

Let $M(\mathcal{R})$ denote a unique standard system of $\mathcal{R}$ up to isomorphism.

4 Transformation and Determinacy

Without loss of generality, a proposition f can be equivalently expressed as a *disjunctive normal form* $\gamma_1 \vee \cdots \vee \gamma_n$, where γ_i are conjunctions of literals. Now, consider the following transformation rules on sets of function requirements:

rule 1 $R \cup \{\gamma_1 \vee \cdots \vee \gamma_n \stackrel{a}{\Rightarrow} \gamma\} \Rightarrow R \cup \{\gamma_1 \stackrel{a}{\Rightarrow} \gamma, \ldots, \gamma_n \stackrel{a}{\Rightarrow} \gamma\}$

rule 2 $R \cup \{\gamma_1 \wedge A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma\} \Rightarrow R \cup \{\gamma_1 \wedge A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma \wedge A\}$

where neither A nor $\neg A$ appears in γ.

rule 3 $R \cup \{\gamma_1 \wedge \neg A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma\} \Rightarrow R \cup \{\gamma_1 \wedge \neg A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma \wedge \neg A\}$

where neither A nor $\neg A$ appears in γ.

Lemma 4.1 *We have the following results on the transformation rules:*

(1) *A transition t is correct w.r.t. a function requirement $\gamma_1 \vee \cdots \vee \gamma_n \stackrel{a}{\Rightarrow} \gamma$ iff it is correct w.r.t. some function requirement $\gamma_i \stackrel{a}{\Rightarrow} \gamma$, for some i.*

(2) *A transition t is correct w.r.t. a function requirement $\gamma_1 \wedge A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma$ iff it is correct w.r.t. the function requirement $\gamma_1 \wedge A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma \wedge A$, where neither A nor $\neg A$ appears in γ.*

(3) *A transition t is correct w.r.t. a function requirement $\gamma_1 \wedge \neg A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma$ iff it is correct w.r.t. the function requirement $\gamma_1 \wedge \neg A \wedge \gamma_2 \stackrel{a}{\Rightarrow} \gamma \wedge \neg A$, where neither A nor $\neg A$ appears in γ.* □

Let $\mathcal{R} = \langle R, \gamma_0 \rangle$ be a requirement. Let $\hat{\mathcal{R}} = \langle \hat{R}, \gamma_0 \rangle$ denote the resulting requirement by applying the above transformation rules to $\mathcal{R}$ as much as possible. We call $\hat{\mathcal{R}}$ the *canonical form* of $\mathcal{R}$.

Theorem 4.1 *Let $\mathcal{R}$ be a requirement. Suppose that state transition systems M and $\hat{M}$ are standard systems of $\mathcal{R}$ and $\hat{\mathcal{R}}$, respectively, then M and $\hat{M}$ are isomorphic* [?]. □

Example 4.1 If we apply the above transformation rules to the requirement $\mathcal{R}_2$ in Example 3.2, we obtain the following requirement $\hat{\mathcal{R}}_2$.

$$\begin{aligned}\hat{\mathcal{R}}_2 = \langle\{ \quad & \rho_1 : A \stackrel{a}{\Rightarrow} \neg A \wedge \neg B, \\ & \rho_2 : \neg A \wedge \neg B \stackrel{b}{\Rightarrow} \neg C \wedge \neg A \wedge \neg B, \\ & \rho_2 : A \wedge C \stackrel{b}{\Rightarrow} \neg C \wedge A, \\ & \rho_3 : \neg C \stackrel{c}{\Rightarrow} C, \\ & \rho_4 : C \stackrel{d}{\Rightarrow} A \wedge C\}, \\ & A \wedge \neg B \wedge \neg C\rangle\end{aligned}$$

By Theorem 4.1, both requirements have the isomorphic standard transition systems. □

Definition 4.1 Let M be a transition system. M is called *deterministic* if there are no transitions $p \stackrel{a}{\rightarrow} q_1$ and $p \stackrel{a}{\rightarrow} q_2$ for any states p, q_1, q_2 and for any input symbol a such that $q_1 \neq q_2$. □

Proposition 4.1 *Let $\mathcal{R}$ a a requirement description. If there are no functions $\rho_1 : f_1 \stackrel{a}{\Rightarrow} f_1'$, $\rho_2 : f_2 \stackrel{a}{\Rightarrow} f_2'$ with the input symbol in common such that $f_1 \wedge f_2$ is consistent, then the standard system of $\mathcal{R}$ is deterministic.*

Proof: Suppose the standard system $M(\mathcal{R})$ is nondeterministic, then there exist transitions $t_1 = \langle p \stackrel{a}{\rightarrow} q_1 \rangle$, $t_2 = \langle p \stackrel{a}{\rightarrow} q_2 \rangle$ for some states p, q_1, q_2 and for some input symbol a such that $q_1 \neq q_2$. Let $\rho_1 : f_1 \stackrel{a}{\Rightarrow} f_1'$, $\rho_2 : f_2 \stackrel{a}{\Rightarrow} f_2'$ be the functions such that $t_1 \models \rho_1$, $t_2 \models \rho_2$. Then, $p \models f_1$ and $p \models f_2$. Hence, $f_1 \wedge f_2$ is consistent. □

5 Synthesis of Specification

Our target is to derive a sound and complete state transition system M from a given requirement description $\mathcal{R} = \langle R, \gamma_0 \rangle$. Now, we state a transformation $\mathcal{T}$ from $\mathcal{R}$ into M. Let define a transition system $\mathcal{T}(\mathcal{R}) = \langle \Gamma, \Sigma, \rightarrow, q_0 \rangle$, where

(1) Γ is a consistent conjunction of literals in $\mathcal{P}$

(2) $\Sigma = \{a \mid \rho : f_{in} \stackrel{a}{\Rightarrow} f_{out} \in R\}$

(3) $\gamma \stackrel{a}{\rightarrow} \gamma'$ iff there exists a function requirement $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out} \in R$ such that

(a) $I(\gamma) \models f_{in}$.

(b) $I(\gamma') \models f_{out}$.

(c) If an atomic proposition A is independent of f_{out}, then $I(\gamma) \models A$ iff $I(\gamma') \models A$.

(4) $q_o = \gamma_0$.

The partial interpretation associated with a state γ in $\mathcal{T}(\mathcal{R})$ is defined as $I(\gamma)$. In other words, the states correspond possible partial interpretations for all atomic propositions in $\mathcal{P}$. It is trivial from the construction that $\mathcal{T}(\mathcal{R})$ is irreducible.

Theorem 5.1 *The state transition system $\mathcal{T}(\mathcal{R})$ derived from a requirement description $\mathcal{R} = \langle R, \gamma_0 \rangle$ by $\mathcal{T}$ is a standard system of $\mathcal{R}$.*

Proof: Soundness: This direction is clear from the construction of the transition system $\mathcal{T}(\mathcal{R})$.

Completeness: Let $M = \langle Q, \Sigma, \rightarrow, q_0 \rangle$ be a sound state transition system with respect to $\mathcal{R}$. Let define a mapping $\xi : Q \rightarrow \Gamma$ by $\xi(q) = \gamma$ for $q \in Q$, where γ is a consistent conjunction of literals such that $I(q) = I(\gamma)$. The mapping ξ is well defined.

Now, we will show that ξ is a homomorphism from M into $\mathcal{T}(\mathcal{R})$. It can be easily checked that $\xi(q_0) = \gamma_0$ since M is a sound transition system and the initial state q_0 in M satisfies only literals appearing in γ_0. Let $p \xrightarrow{a} q$ be any transition in M. Suppose $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ be the function requirement in R satisfied by this transition. So, we have

$$p \models f_{in} \qquad q \models f_{out}.$$

Thus,

$$\xi(p) \models f_{in} \qquad \xi(q) \models f_{out},$$

by the definition of ξ. The following statement

$$\xi(p) \models A \qquad \text{iff} \qquad \xi(q) \models A$$

for all atomic proposition A independent of f_{out}, can be implied by the statement

$$p \models A \qquad \text{iff} \qquad q \models A$$

for all atomic proposition A independent of f_{out}. Therefore, we have a transition $\xi(p) \xrightarrow{a} \xi(q)$ in $\mathcal{T}(\mathcal{R})$. By the definition of ξ, $p \models f$ implies $\xi(p) \models f$ for all proposition f. Hence, ξ is a homomorphism from M into $\mathcal{T}(\mathcal{R})$. □

6 Discussions

6.1 Partial Logical Petri Net

The derived state transition system $\mathcal{T}(\mathcal{R})$ from a requirement $\mathcal{R}$ can be proved to coincide with the reachability graph of a Partial Logical Petri Net. A Partial Logical Petri Net, where inhibited arcs are allowed in both inputs and outputs of transitions, and two kinds of tokens are provided. The Partial Logical Petri Net is an straight extension of a Logical Petri Net proposed by Song and et al [13].

Definition 6.1 (Partial Logical Petri Net)
A *Partial Logical Petri Net* is a tuple $PN = \langle P, T, I, O, M_0 \rangle$, where

(1) P is a set of *places*;

(2) T is a set of *transitions*;

(3) $I = \langle I_p, I_n \rangle$ is a pair of *input functions* I_p, $I_n : T \rightarrow 2^P$ such that $I_p(t) \cap I_n(t) = \emptyset$ for all $t \in T$;

(4) $O = \langle O_p, O_n \rangle$ is a pair of *output functions* O_p, $O_n : T \rightarrow 2^P$ such that $O_p(t) \cap O_n(t) = \emptyset$, for all $t \in T$;

(5) $M_0 : P \rightarrow \{0, 1, *\}$ is an *initial marking*. □

A Partial Logical Petri Net can be represented as a bipartite graph in the almost same way as a usual Petri Net [10]. However, in a Partial Logical Petri Net, we have the following extensions and restrictions.

- There are two kinds of arcs, called *positive arcs* and *negative arcs*. If $p \in I_p(t)$ ($p \in O_p(t)$), we make a positive arc, depicted as $\rightarrow$, from p to t (from t to p). If $p \in I_n(t)$ ($p \in O_n(t)$), we make a negative arc, depicted as —o, from p to t (from t to p).

- There are two kinds of tokens, a *positive token* • and a *negative token* ○ which represent truth constant **true** and **false**, respectively.

- Marking functions are restricted to the functions with the range $\{0, 1, *\}$, where 0, 1, and $*$ means that the associated condition with the place is "not satisfied", "satisfied", and "undefined", respectively.

The graphical representation of a Partial Logical Petri Net is given in Figure 2 (a).

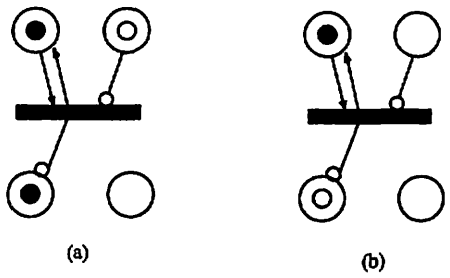

Figure 2: Partial Logical Petri Nets

In a marking M, a transitions t is *fireable* (*executable*) if the following conditions are satisfied:

(1) $M(p) = 1$ for all $p \in I_p(t)$.
(2) $M(p) = 0$ for all $p \in I_n(t)$.

If t is fireable, then t suddenly fires and the marking is changed into the marking M' defined by

$$M'(p) = \begin{cases} 0 & \text{if } p \in O_n(t) \\ 1 & \text{if } p \in O_p(t) \\ * & \text{if } p \in (I_n(t) \cup I_p(t)) \\ & \cap O_n(t)^c \cap O_p(t)^c \\ M(p) & \text{otherwise} \end{cases}$$

The transition in the net (a) in Figure 2 is fireable. After firing, the marking is changed into the one (b) in the figure.

Let $\mathcal{R} = \langle R, \gamma_0 \rangle$ be a requirement in canonical form. We can obtain a Partial Logical Petri Net $\langle P, T, I, O, M_0 \rangle$ from $\mathcal{R}$ as follows:

1. $P = \mathcal{P}$: Places correspond atomic propositions.

2. $T = R$: Transitions correspond function requirements.

3. Let $\rho : A_1 \wedge \cdots \wedge A_n \wedge \neg B_1 \wedge \cdots \wedge \neg B_m \stackrel{a}{\Rightarrow} C_1 \wedge \cdots \wedge C_j \wedge \neg D_1 \wedge \cdots \wedge \neg D_k$ be a function, where capital letters denote atomic propositions. Then, define input functions $I = \langle I_p, I_n \rangle$ and output functions $O = \langle O_p, O_n \rangle$ by

$$I_p(\rho) = \{A_1, \ldots, A_n\}$$
$$I_n(\rho) = \{B_1, \ldots, B_m\}$$
$$O_p(\rho) = \{C_1, \ldots, C_j\}$$
$$O_n(\rho) = \{D_1, \ldots, D_k.\}$$

4. The initial marking M_0 is defined by

$$M_0(A) = \begin{cases} 0 & \text{if } A \text{ appears negative in } \gamma_0 \\ 1 & \text{if } A \text{ appears positive in } \gamma_0 \\ * & \text{otherwise} \end{cases}$$

Example 6.1 If we apply the above transformation to the canonical form in Example 4.1 of the requirement in Example 3.2, we obtain the Partial Logical Petri Net in Figure 3. The resulting reachability graph of the net coincide with the transition system (b) in Figure 1. This can be guaranteed in general by the next proposition. □

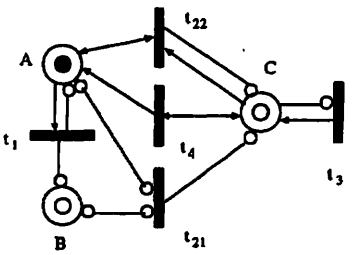

Figure 3: The transformed Partial Logical Petri Net

Proposition 6.1 *Let M be a standard system of a requirement $\mathcal{R}$. Then, the reachability graph of the Partial Logical Petri Net derived from $\hat{\mathcal{R}}$ is isomorphic to M.* □

6.2 Branching Time Temporal Logic

A function requirement $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ can be expressed as a proposition $\Box(f_{in} \supset \langle a \rangle f_{out})$ in an extended branching time temporal logic.

6.3 Production System

The derived transition system $\mathcal{T}(\mathcal{R})$ can be characterized by Production Systems as well. To be more precise, if $\mathcal{R}$ is a requirement in canonical form, then each function requirement

$\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ can be regarded as a production rule $f_{in} \rightarrow f_{out}$. Then, we have the following result.

Proposition 6.2 *Let $\mathcal{R}$ be a requirement in canonical form. If we take a function requirement $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ as a production rule $f_{in} \rightarrow f_{out}$, then the state transition system of the resulting production system is isomorphic to the standard transition system $T(\mathcal{R})$.* □

7 Conclusion

A formal methodology for the description of system requirements and the synthesis of formal specifications from user requirements have been presented. We have specifically dealt with the issues (1) mathematical description of function requirements and their relationship with formal specifications represented as transition systems, (2) soundness and completeness of the system, (3) derivation of state transition systems from user requirements, (4) methodologies on the topics in software design, and (5) some discussions on Partial Logical Petri Nets and Production Systems. The proposed framework provides theoretical and practical tools for system design. To conclude the paper, we state some further comments on our methodology.

Extension to Predicate Logic The underlying logic of this paper may be easily extended to first order predicate logic. For example, the function of `channel_up` in the CATV system is expressed more precisely by the function requirement

$$\texttt{channel\_up:} \quad power on \wedge \neg force \wedge \neg buzzer \wedge ch(x) \stackrel{chup}{\Longrightarrow} power on \wedge ch(x+1)$$

In the above description, the first order variable x is quantified universally.

Branching time Temporal Logic A function requirement $\rho : f_{in} \stackrel{a}{\Rightarrow} f_{out}$ can be expressed as a proposition $\Box(f_{in} \supset \langle a \rangle f_{out})$ in an extended branching time temporal logic. Furthermore, constraints written in temporal logic might be useful.

References

[1] ISO., Estelle: A Formal Description Technique based on the Extended State Transition Model, ISO 9074, 1989.

[2] ISO., *Information Processing Systems - Open System Interconnection - LOTOS - A Formal Description Technique based on the Temporal Ordering of Observational Behavior*, IS 8807, 1989.

[3] CCITT., *SDL: Specification and Description Language*, CCITT Z.100, 1988.

[4] Bolognesi, T., Brinksma, Ed., Introduction to the ISO Specification Language LOTOS, in *the Formal Description Technique LOTOS*, Elsevier Sci. Pub., pp.23–73, 1989.

[5] Emerson, E.A., Temporal and Modal Logic, *Handbook of Theoretical Computer Science*, Elsevier Science Publishers B.V., pp.995–1072, 1990.

[6] Gotzhein, R., Specifying Communication Services with Temporal Logic, *Protocol Specification, Testing and Verification*, XL, pp.295–309, 1990.

[7] van Glabbeek, R.J., *The Linear Time - Branching Time Spectrum*, Lecture Notes in Comput. Sci. 458, Springer-Verlag, 1990.

[8] Hirakawa, Y., Takenaka, T., Telecommunication Service Description using State Transition Rules, Proc. 6th Int. Work. Software Specification and Design, pp.140–147, 1991.

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

[10] Murata, T., Petri Nets: Properties, Analysis and Applications, IEEE Proc. Vol.77, No.4, pp.541–580, 1989.

[11] Plotkin, G.D., *A Structural Approach to Operational Semantics*, Computer Science Department, Aarhus University, DAIMI FN-19, 1981.

[12] Song, K., Togashi, A., Shiratori, N., Verification and refinement for system requirements, IEICE Trans. on Fundamentals of Elec., Comm. and Comp. Sci., Vol.E78-A, No.11, pp.1468–1478, 1995.

[13] Song, K., Togashi, A., Shiratori, N., A requirement description method based on propositional logic and its semantic description by state transition system, Trans. of Information Processing Society of Japan, Vol.37, No.4, pp.511–519, 1996 (in Japanese).