Schedulability analysis for real-time mobile systems

Cong Chen*,Yangyang Chen*, Jian-Min Jiang*†, Shi Zhang*, Zhong Hong*, Hongping Shu†, and Qiong Zeng†
*College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350007, China
†College of Software Engineering, ChengDu University of Information Technology, Chengdu 610103, China

***Abstract*—Autonomous driving systems are complex real-time mobile systems. To guarantee their safety and security, the mobile objects (agents) in these systems must be isolated from each other so that they do not collide with each other. Since isolation means two or more mobile objects cannot be located in the same area at the same time, a scheduling policy is required to control the movement of these mobile objects. However, traditional scheduling theories are based on task scheduling which is coarse-grained and cannot be directly used for fine-grained isolation controls. In this paper, we first propose an event-based formal model called a *time dependency structure* which is used to model and analyze real-time mobile systems. Then, an event-based schedule is defined. Finally, we analyze the schedulability of isolation—that is, checking whether a given schedule ensures the isolation relationship among mobile objects or not.**

***Index Terms*—mobility, isolation, scheduling policy, ambient.**

I. Introduction

Autonomous driving systems are complex mobile systems, which are a prominent subcategory of cyber-physical systems. In these complex mobile systems, safety and security, especially the isolation of mobile objects (agents), has become crucial issues. Isolation means two or more mobile objects cannot be located in the same area at the same time. The mobile objects in real-time mobile systems must be isolated from each other so that they do not collide with each other. Thus, we should have an effective mechanism for checking whether a given scheduling policy can ensure the isolation of mobile objects or not.

Existing scheduling theories, e.g, [2], [7], [8], [10]–[12], focus on task scheduling. Task scheduling mainly consider how to generate the optimal scheduling policies while the objective of the schedulability analysis is to verify that there are no violations of constraint conditions. However, as for the inherent complexity of scheduling, existing work is far from enough to solve these problems. Specifically, in the practical mobile systems, mobile objects and environments interact with each other, it is very difficult to separate a mobile system into independent tasks. We cannot directly use existing methods and techniques to obtain the scheduling policies for the isolation of mobile objects.

To solve the problem, it is necessary for a new scheduling policy to control the whole mobile system [1], [4]. Jiang et al. [4] have proposed more fine-grained event-based scheduling instead of task scheduling. An event is generally the occurrence of an action or activity. A task, process or complex activity may consist of multiple events [6]. A complex scheduling problem cannot be decomposed into independent tasks, but it can be divided into sub-problems of event-based scheduling. Though Jiang et al. [4] have discussed event-based scheduling, such event-based scheduling does not consider real-time scheduling controls, especially the scheduling in real-time mobile systems.

To investigate the scheduling in real-time mobile systems, we must model real-time mobile systems. We extend the dependency structure model [4], [5] and add the time modeling power to it. Such a model is called a *time dependency structure*, which can conveniently specify the timing constraints and mobility of real-time mobile systems.

In this paper, we first introduce a time dependency structure. Then, an event-based schedule is defined. Finally, we investigate the schedulability analysis of isolation—that is, checking whether a given schedule ensures the isolation relationship among mobile objects or not in a real-time mobile system.

II. Notation and Running example

We will adopt the concept similar to the ambient calculus [3], where computation happens in an ambient that is a closed and bounded place and a mobile object (agent) can enter or exit an ambient.

Here, we first give some notations. Given a set X, the notations 2^X and $|X|$ denote the power set and the size of X, respectively. **Time** $= [0, \infty)$, the set of non-negative reals, denotes the domain of time. $\mathbf{A}$ and $\mathbf{M}$ denote the sets of ambients and mobile objects, respectively. The event of a mobile object $\mathcal{M} \in \mathbf{M}$ for entering an ambient $\mathcal{A}(\mathcal{A} \in \mathbf{A})$ is denoted by $en_{\mathcal{A}}^{\mathcal{M}}$ and the event of $\mathcal{M}$ for exiting $\mathcal{A}$ is done by $ex_{\mathcal{A}}^{\mathcal{M}}$. In fact, it is enough for us to only use the two movement events (entering and exiting events) for specifying the mobility in a mobile system. For more information, please refer to our previous work [4].

We present a running example, which is a simple yet typical mobile system where a passenger *John* needs to take a bus in a road intersection area. It is assumed that all the vehicles are equipped with Navigation Satellite System (GPS or BDS) devices and have access to a digital map database, which provide them with critical information such as position, heading, speed, road and lane details. The road area is represented as a

Corresponding author: Jian-Min Jiang (jjm@fjnu.edu.cn). This work is supported by National Natural Science Foundation of China (No. 61772004) and the NSF of Fujian province (No. 2018J01777).

DOI reference number: 10.18293/SEKE2019-090

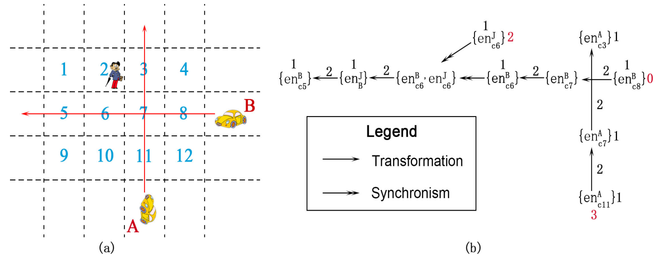

Fig. 1. A simple mobile system

grid which is divided into small cells in Figure 1.(a). Each cell in the grid is associated with a unique identifier. The buses A and B pass the intersection cell $c7$. John is located in the cell $c2$. The bus A moves along the cells $c11, c7, c3$ and the bus B does along the cells $c8, c7, c6, c5$.

John may enter the cell $c6$ and take the bus B. For simplification, we give some notations. John is denoted as J, and the event of John for entering the bus B (resp. the cell $c6$) is denoted as en^J_B (resp. en^J_{c6}). If a bus X enters a cell cx, the entering event is en^X_{cx}. To simplify modeling specification, we only consider the entering events because the event of exiting one cell in fact means the event of entering the next adjacent cell. Thus, there exist the following events: en^J_{c6}, en^J_B, $en^A_{c11}, en^A_{c7}, en^A_{c3}, en^B_{c8}, en^B_{c7}, en^B_{c6}, en^B_{c5}$.

Note that if there exist two or more vehicles in the same cell, they will collide. To avoid collision, when the buses A and B enter the cell $c7$, they must be scheduled so that they pass through the cell $c7$ in sequence. Additionally, John should enter the bus B in the cell $c6$ before B leaves.

Since a time dependency structure can represent a real-time mobile system, it is used to denote such a real-time mobile system. A real-time mobile system $\mathcal{TDS}$ may contains multiple mobile objects and ambients. For convenience, the notation $\mathcal{TDS}_x \sqsubset \mathcal{TDS}$ is used to denote that $\mathcal{TDS}_x$ is a mobile object or ambient of a real-time mobile system $\mathcal{TDS}$.

III. System Model

An event is a core concept here, which means an occurrence of an activity or action. If an event occurs, such an event is said to be *available*; otherwise it is *unavailable*. The dependency structure model [4], [5] uses an *event set* (a set of events) as a basic element. If all the events in an event set are *available*, such an event set is said to be *available*; otherwise it is said to be *unavailable*. We equip events and the relationship among events with time attributes, and introduce the *time dependency structure*.

Definition III.1 A **time dependency structure** ($\mathcal{TDS}$) is a tuple $\langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ with
–$\mathcal{E}$, a finite set of events,
–$\mathbb{I} \subseteq 2^{\mathcal{E}}$, the set of initially available event sets,
–$\mathbb{T} \subseteq 2^{\mathcal{E}} \setminus \{\emptyset\} \times 2^{\mathcal{E}} \setminus \{\emptyset\}$, the (asymmetric) *transformation* relation,
–$\mathbb{S} \subseteq 2^{\mathcal{E}}$, the *synchronism* relation such that $\forall A \in \mathbb{S} : |A| > 1$,
–$\mathbb{C} \subseteq 2^{\mathcal{E}}$, the *choice* relation such that $\forall A \in \mathbb{C} : |A| > 1$,
–$\mathbb{W} : \mathcal{E} \to \{1, 2, 3, ...\}$, the capacity function,
–$\mathbb{F} \subseteq 2^{\mathcal{E}}$, the set of finally available event sets,
–$\mathcal{T}_i : \bigcup_{X \in \mathbb{I}} X \to$ **Time**, the initial time function,
–$\mathcal{T}_e : \mathcal{E} \to$ **Time**, the event delay function, and
–$\mathcal{T}_t : \mathbb{T} \to$ **Time**, the transformation delay function.

Here, for all $A, B \in 2^{\mathcal{E}}$, $(A, B) \in \mathbb{T}$ is called a *transformation dependency*, denoted as $A \to B$, all read as B *depending on* A, and A, B are called the *pre-* and *post-dependency set* of the dependency (A, B), respectively. The events in A, B are called the *pre-* and *post-events* of (A, B), respectively.

Transformation is a binary relation between event sets where a transformation dependency $(A, B) \in \mathbb{T}$ is that the occurrences of all the events in B depends on the occurrences of all the events in A. A set $A \in \mathbb{S}$ and a set $B \in \mathbb{C}$ are called a *synchronism set* and a *choice set*, respectively. The capacity function $\mathbb{W}$ restricts the available number of events, that is, if an event e may cause the occurrence of n events, then the capacity of such an event is n ($\mathbb{W}(e) = n$). The capacity function is similar to the token capacity function of places in a Petri net [9], which is used to control a loop.

To support multiple clock modeling, the initial time function is introduced. $\mathcal{T}_i(e)$ refers to the initial clock valuation of the initial available event e. The occurrence of an event may go on for some time. $\mathcal{T}_e(e)$ specifies the timing constraint of the event e. A transformation dependency expresses the dependency relationship between the two event sets and may have a time delay constraint. If a transformation dependency (A, B) has time delay t', $\mathcal{T}_t((A, B)) = t'$.

In our running example (see Figure 1(b)), we can assume that it takes 1 time unit to enter an ambient and takes 2 time units to cross a ambient for vehicle A and B. we assume that the initial time of the vehicles A, B and passenger John are 3, 0, and 2, respectively. Therefore, the running example can be modeled as $\mathcal{TDS}_{run} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ where
$\mathcal{E} = \{en^A_{c11}, en^A_{c7}, en^A_{c3}, en^B_{c8}, en^B_{c7}, en^B_{c6}, en^B_{c5}, en^J_{c6}, en^J_B\}$,
$\mathbb{I} = \{\{en^A_{c11}\}, \{en^B_{c8}\}, \{en^J_{c6}\}\}$,
$\mathbb{T} = \{(\{en^A_{c11}\}, \{en^A_{c7}\}), (\{en^A_{c7}\}, \{en^A_{c3}\}), (\{en^B_{c8}\}, \{en^B_{c7}\}), (\{en^B_{c7}\}, \{en^B_{c6}\}), (\{en^B_{c6}, en^J_{c6}\}, \{en^J_B\}), (\{en^J_B\}, \{en^B_{c5}\})\}$,
$\mathbb{S} = \{\{en^B_{c6}, en^J_{c6}\}\}$, $\mathbb{C} = \emptyset$, $\forall e \in \mathcal{E}, \mathbb{W}(e) = \infty$, $\mathbb{F} = \{\{en^A_{c3}\}, \{en^B_{c5}\}\}$, and the timing constraints are as follows:
$\mathcal{T}_i(en^A_{c11}) = 3, \mathcal{T}_i(en^B_{c8}) = 0, \mathcal{T}_i(en^J_{c6}) = 2$,
$\mathcal{T}_e(en^A_{c11}) = \mathcal{T}_e(en^A_{c7}) = \mathcal{T}_e(en^A_{c3}) = \mathcal{T}_e(en^B_{c8}) = \mathcal{T}_e(en^B_{c7}) = \mathcal{T}_e(en^B_{c6}) = \mathcal{T}_e(en^B_{c5}) = \mathcal{T}_e(en^J_{c6}) = \mathcal{T}_e(en^J_B) = 1$,
$\mathcal{T}_t((\{en^A_{c11}\}, \{en^A_{c7}\})) = \mathcal{T}_t((\{en^A_{c7}\}, \{en^A_{c3}\})) = \mathcal{T}_t((\{en^B_{c8}\}, \{en^B_{c7}\})) = \mathcal{T}_t((\{en^B_{c7}\}, \{en^B_{c6}\})) = \mathcal{T}_t((\{en^B_{c6}, en^J_{c6}\}, \{en^J_B\})) = \mathcal{T}_t((\{en^J_B\}, \{en^B_{c6}\})) = 2$.

A system can just run along the path formed by its transformation dependencies. Synchronism, choice and timing constraints only control the execution of such a system. While a system runs, each of transformation dependencies may lead to the change of its states. With the passage of time, a transformation dependency may be *activated*. Only activated transformation dependencies will be possibly executed. Thus, a state includes the current possibly available events, the number of possibly activating transformation dependencies, the "absolute" time of the occurrence of every possibly available event, and currently activated transformation dependencies.

Definition III.2 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure. A *state* of $\mathcal{TDS}$ is a tuple $\mathcal{S} = \langle \Delta, \digamma, f_t, \Gamma \rangle$ where $\Delta \subseteq \mathcal{E}$ is the set of possibly available events, the *availability function* $\digamma : \Delta \to \mathbb{Z}^*$ is a function from Δ to the set $\mathbb{Z}^*$ of nonnegative integers, the time function $f_t : \Delta \to \mathbf{Time}$, and $\Gamma \subseteq \mathbb{T}$ is the set of activated transformation dependencies satisfying for all dependencies $(A, B) \in \Gamma \Rightarrow A \subseteq \Delta$. The *initial state* of $\mathcal{TDS}$ is defined as $\mathcal{S}_0 = \langle \Delta_0, \digamma_0, f_{t0}, \Gamma_0 \rangle$ such that $\Delta_0 = \bigcup_{X \in \mathbb{I}} X$, $\forall e \in \Delta_0 : \digamma_0(e) = |\{(A, B) \in \mathbb{T} \mid e \in A\}|$, $\forall e \in \Delta_0 : f_{t0}(e) = \mathcal{T}_i(e) + \mathcal{T}_e(e)$, and $\Gamma_0 = \{(A, B) \mid A \in \mathbb{I}, (A, B) \in \mathbb{T}\}$.

For example, the initial state of $\mathcal{TDS}_{run}$ is $\mathcal{S}_0 = \langle \Delta_0, \digamma_0, f_{t0}, \Gamma_0 \rangle$ where $\Delta_0 = \{\{en^A_{c11}\}, \{en^B_{c8}\}, \{en^J_{c6}\}\}$, $\digamma_0(en^A_{c11}) = \digamma_0(en^B_{c8}) = \digamma_0(en^J_{c6}) = 1$, $f_{t0}(en^A_{c11}) = 4$, $f_{t0}(en^B_{c8}) = 1$, $f_{t0}(en^J_{c6}) = 3$ and $\Gamma_0 = \{(en^A_{c11}, en^A_{c7}), (en^B_{c8}, en^B_{c7}), (\{en^J_{c6}, en^B_{c6}\}, en^J_B)\}$.

For convenience, the state $\langle \Delta, \digamma, f_t \rangle$ is denoted as $\{\langle e, \digamma(e), f_t(e) \rangle \mid e \in \Delta\}$. The availability function $\digamma$ is similar to the marking of Petri nets. Given an event e and $\digamma(e) = n$, n is called the *availability value* of e.

Given a synchronism set C, the *latest available time delay* of the events in a synchronism set is denoted by $Max\{f_t(e) \mid e \in C\}$. Note that since the absolute time of the occurrence of the events in a synchronism set is computed from the initial state of a system, one cannot directly determine which events occur in what order before the system starts to run.

Definition III.3 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure and $\mathcal{S}_1 = \langle \Delta_1, \digamma_1, f_{t1}, \Gamma_1 \rangle$, $\mathcal{S}_2 = \langle \Delta_2, \digamma_2, f_{t2}, \Gamma_2 \rangle$ be two of its states.

$\mathcal{S}_1$ can evolve into $\mathcal{S}_2$ by executing a transformation dependency (A, B), denoted by $\mathcal{S}_1 \stackrel{(A,B)}{\longrightarrow} \mathcal{S}_2$, if the following conditions hold:

(1) $(A, B) \in \Gamma_1$,

(2) $\nexists (E, F) \in \Gamma_1 : Max\{f_{t1}(e) \mid e \in E\} + \mathcal{T}_t((E, F)) < Max\{f_{t1}(e) \mid e \in A\} + \mathcal{T}_t((A, B))$,

(3) $\Delta_2 = \{e \in \Delta_1 \mid e \notin A \vee (\digamma_1(e) - (1 + x) > 0 \wedge e \in A)\} \cup B$,

(4) $\forall e \in \Delta_2 : \digamma_2(e) \leq \mathbb{W}(e) \wedge \digamma_2(e) =$

$$\begin{cases} \digamma_1(e) - (1 + x) & : \ e \in A \setminus B \\ \digamma_1(e) & : \ e \in (\Delta_1 \setminus (A \cup B)) \\ \digamma_1(e) - (1 + x) + y & : \ e \in A \cap B \\ \digamma_1(e) + y & : \ e \in (\Delta_1 \setminus A) \cap B \\ y & : \ e \in B \setminus \Delta_1 \end{cases}$$

where $y = |\{(X, Y) \in \mathbb{T} \mid X \cap B \neq \emptyset\}|$ and $x = |\{(A, X) \in \mathbb{T} \mid \exists e \in X, \exists e' \in B, \exists C \in \mathbb{C} : e \neq e' \wedge \{e, e'\} \in C\}|$,

(5) $\Gamma_2 = (\Gamma_1 \setminus (\{(A, B)\} \cup B^{\mathbb{C}})) \cup B^{\mathbb{T}} \cup B^{\mathbb{S}}$ where $B^{\mathbb{T}} = \{(B, X) \mid (B, X) \in \mathbb{T}\}$, $B^{\mathbb{S}} = \{(X, Y) \in \mathbb{T} \mid X \in \mathbb{S}, X \subseteq \Delta_1 \cup B, B \subseteq X, Y \subseteq \mathcal{E}\}$ and $B^{\mathbb{C}} = \{(W, X) \in \mathbb{T} \mid W \subseteq \mathcal{E}, \exists e \in X, \exists e' \in B, \exists C \in \mathbb{C} : e \neq e' \wedge \{e, e'\} \in C\}$, and

(6) $\forall e \in \Delta_2 : f_{t2}(e) = if\ e \in B\ then\ Max\{f_{t1}(e') \mid e' \in A\} + \mathcal{T}_t((A, B)) + \mathcal{T}_e(e)\ else\ f_{t1}(e)$.

According to the condition (2) of the preceding definition, we have $\mathcal{S}_0 \stackrel{(en^B_{c8}, en^B_{c7})}{\longrightarrow} \mathcal{S}_1$ in the running example, and then we have $\Delta_1 = \{\{en^A_{c11}\}, \{en^B_{c7}\}, \{en^J_{c6}\}\}$ (by the condition (3)), $\digamma_1(en^A_{c11}) = \digamma_1(en^B_{c7}) = \digamma_1(en^J_{c6}) = 1$ (by the condition (4)), and $\Gamma_1 = \{(en^A_{c11}, en^A_{c7}), (en^B_{c7}, en^B_{c6}), (\{en^J_{c6}, en^B_{c6}\}, en^J_B)\}$ (by the condition (5)). Thus, $\mathcal{S}_1 = \langle \Delta_1, \digamma_1, f_{t1}, \Gamma_1 \rangle$ where $f_{t1}(en^A_{c11}) = 4$, $f_{t1}(en^B_{c7}) = 4$, and $f_{t1}(en^J_{c6}) = 3$ (by the condition (6)).

A time dependency structure can be used to reason about the behavior and properties of a real-time system. We define some properties of a time dependency structure here.

Definition III.4 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure and $\mathcal{S}_0$ be the initial state of $\mathcal{TDS}$. Let $\mathcal{S}, \mathcal{S}'$ be two states of $\mathcal{TDS}$. A state $\mathcal{S}$ is said to be *reachable from* $\mathcal{S}'$, denoted as $\mathcal{S}' \stackrel{*}{\to} \mathcal{S}$, if there exist the states $\mathcal{S}'_1, \cdots, \mathcal{S}'_{n-1}$ such that $\mathcal{S}' \stackrel{d'_1}{\to} \mathcal{S}'_1 \cdots \mathcal{S}'_{n-1} \stackrel{d'_n}{\to} \mathcal{S}$ ($d'_i \in \mathbb{T}, i \in \{1, \cdots, n\}$). $Sta(\mathcal{TDS})$ denotes the set of all reachable states in $\mathcal{TDS}$.

IV. Scheduling and Isolation control

In the section, we will introduce the notion of a schedule and analyze the isolation relationship of mobile objects in a real-time mobile system in order to explore the isolation.

Definition IV.1 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure.

A sequence $f = \mathcal{S}_0 X_1 \mathcal{S}_1 \cdots X_n \mathcal{S}_n$ is called a *full sequence* of $\mathcal{TDS}$ iff $\mathcal{S}_0 = \langle \Delta, \digamma, f_t, \Gamma \rangle, \mathcal{S}_1 = \langle \Delta_1, \digamma_1, f_{t1}, \Gamma_1 \rangle, \cdots, \mathcal{S}_n = \langle \Delta_n, \digamma_n, f_{tn}, \Gamma_n \rangle$ are states in $\mathcal{TDS}$ and $\forall i \in \{0, 1, \cdots, n\}, X_i \subseteq \mathcal{E}$ such that $\mathcal{S}_0 \stackrel{*}{\to} \mathcal{S}_1 \stackrel{*}{\to} \cdots \stackrel{*}{\to} \mathcal{S}_n$ and $\forall i \in \{1, \cdots, n\}, \forall e_1, e_2 \in X_i : (X_i \cap \Delta_i = X_i) \wedge (f_{ti-1}(e_1) = f_{ti-1}(e_2))$. Here, the sequence $s = X_1 \cdots X_n$ is called *a schedule* of $\mathcal{TDS}$ or is said to be *schedulable* in $\mathcal{TDS}$. $Sches(\mathcal{TDS})$ denotes the set of all the schedules in $\mathcal{TDS}$.

As the Definition IV.1, $f_{run} = \mathcal{S}_0\{en^B_{c8}\}\mathcal{S}_1\{en^A_{c11}, en^B_{c7}\}\mathcal{S}_2\{en^A_{c7}, en^B_{c6}, en^J_{c6}\}\mathcal{S}_3\{en^J_B\}\mathcal{S}_4$ is a full sequence of $\mathcal{TDS}_{run}$, and $s_{run} = \{en^B_{c8}\}\{en^A_{c11}, en^B_{c7}\}\{en^A_{c7}, en^B_{c6}, en^J_{c6}\}\{en^J_B\}$ is a schedule of $\mathcal{TDS}_{run}$.

A schedule is an ordered event set sequence, where the events in the front event set occur prior to those in the back event set. The scheduler of a system in fact is a controller that restricts the behaviour of such a system so that given scheduling requirements are met [1].

Definition IV.2 Let $\mathcal{TDS}$ be a time dependency structure and $s \in Sches(\mathcal{TDS})$. *The restriction of* $\mathcal{TDS}$ *to* the schedule s is denoted by $\mathcal{TDS}\rfloor_s$.

In this definition, $\mathcal{TDS}\rfloor_s$ means the time dependency structure $\mathcal{TDS}$ whose behavior is restricted to the schedule s or the time dependency structure $\mathcal{TDS}$ runs in terms of the control of the schedule s.

Proposition IV.1 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure and let s be a schedule in $\mathcal{TDS}$, then $Sta(\mathcal{TDS}\rfloor_s) \subseteq Sta(\mathcal{TDS})$.

The proposition shows that the states of scheduled mobile system are part of those of the original system, respectively.

Definition IV.3 Let $\mathcal{TDS}$ be a time dependency structure and $s = X_1 \cdots X_n \in Sches(\mathcal{TDS})$. The restriction of $\mathbf{A}$ to the schedule s is defined as $\mathbf{A}\!\uparrow_s = \{\mathcal{A} \in \mathbf{A} \mid \exists \mathcal{M} \in \mathbf{M} : en^{\mathcal{M}}_{\mathcal{A}} \in X_1 \cup \cdots \cup X_n\}$.

In fact we use $\mathbf{A}\uparrow_s$ to denote the set of all the ambients that are involved in the schedule s.

Definition IV.4 Let $\mathcal{TDS}$ be a time dependency structure. Let $\mathbf{M}_s \subseteq \mathbf{M}$. Let $\mathcal{A} \in \mathbf{A}$, $\mathcal{M}_1, \mathcal{M}_2 \in \mathbf{M}_s$ and $\mathcal{A}, \mathcal{M}_1, \mathcal{M}_2 \sqsubset \mathcal{TDS}$.

TABLE I

STATES OF THE TIME DEPENDENCY STRUCTURE OF THE RUNNING EXAMPLE SYSTEM

Source state	$\langle \Delta, F, f_t \rangle$	Γ	ED	Target state
$\mathcal{S}_0$	$\{< en^A_{c11}, 1, 4 >, < en^B_{c8}, 1, 1 >, < en^J_{c6}, 1, 3 >\}$	$\{(\{en^A_{c11}\}, \{en^A_{c7}\}), (\{en^B_{c8}\}, \{en^B_{c7}\})\}$	$(\{en^B_{c8}\}, \{en^B_{c7}\})$	$\mathcal{S}_1$
$\mathcal{S}_1$	$\{< en^A_{c11}, 1, 4 >, < en^B_{c7}, 1, 4 >, < en^J_{c6}, 1, 3 >\}$	$\{(\{en^A_{c11}\}, \{en^A_{c7}\}), (\{en^B_{c7}\}, \{en^B_{c6}\})\}$	$(\{en^A_{c11}\}, \{en^A_{c7}\})$, $(\{en^B_{c7}\}, \{en^B_{c6}\})$	$\mathcal{S}_2$
$\mathcal{S}_2$	$\{< en^A_{c7}, 1, 7 >, < en^B_{c6}, 1, 7 >, < en^J_{c6}, 1, 3 >\}$	$\{(\{en^A_{c7}\}, \{en^A_{c3}\}), (\{en^B_{c6}, en^J_{c6}\}, \{en^J_B\})\}$	$(\{en^A_{c7}\}, \{en^A_{c3}\})$, $(\{en^B_{c6}, en^J_{c6}\}, \{en^J_B\})$	$\mathcal{S}_3$
$\mathcal{S}_3$	$\{< en^A_{c3}, 0, 10 >, < en^J_B, 1, 10 >$	$\{(\{en^J_B\}, \{en^B_{c5}\})\}$	$(\{en^J_B\}, \{en^B_{c5}\})$	$\mathcal{S}_4$
$\mathcal{S}_4$	$\{< en^A_{c3}, 0, 10 >, < en^B_{c5}, 0, 13 >$			

Note that "ED" means currently executed transformation dependencies.

$\mathcal{M}_1$ is said to be *isolated from* $\mathcal{M}_2$ for $\mathcal{A}$ in $\mathcal{TDS}$, denoted by $\mathcal{M}_1 \bigcirc_{\mathcal{A}} \mathcal{M}_2$ in $\mathcal{TDS}$, iff either $\forall s \in Sches(\mathcal{TDS})$, $\mathcal{A} \notin \mathbf{A}{\uparrow}_s$, or $\forall s = B_1 \cdots B_n \in Sches(\mathcal{TDS}), (\exists \mathcal{X} \in \mathbf{A}, \nexists en^{M_2}_{\mathcal{A}} \in B_1 \cup \cdots \cup B_n : en^{M_1}_{\mathcal{A}} \in B_1 \wedge en^{M_1}_{\mathcal{X}} \in B_n) \vee (\exists \mathcal{Y} \in \mathbf{A}, \nexists en^{M_1}_{\mathcal{A}} \in B_1 \cup \cdots \cup B_n : en^{M_2}_{\mathcal{A}} \in B_1 \wedge en^{M_2}_{\mathcal{Y}} \in B_n)$.

$\mathbf{M}_s$ is the set of mobile objects which need to isolate in order to avoid collision. For example, in the running example, we let $\mathbf{M}_s = \{A, B\}$ because of vehicle A and B need to isolate while John as a passenger and the vehicle B are not isolated from each other in the ambient $c6$ so that John takes the vehicle B.

This definition shows that if $\mathcal{M}_1$ is isolated from $\mathcal{M}_2$, this means one of the following three cases holds: (1) $\mathcal{M}_1$ and $\mathcal{M}_2$ do not enter $\mathcal{A}$, (2) one of $\mathcal{M}_1$ and $\mathcal{M}_2$ enters $\mathcal{A}$, and (3) when the two mobile objects $\mathcal{M}_1$ and $\mathcal{M}_2$ both need to enter $\mathcal{A}$, one does not enter the ambient $\mathcal{A}$ until the other exits $\mathcal{A}$.

Theorem IV.1 Let $\mathcal{TDS} = \langle \mathcal{E}, \mathbb{I}, \mathbb{T}, \mathbb{S}, \mathbb{C}, \mathbb{W}, \mathbb{F}, \mathcal{T}_i, \mathcal{T}_e, \mathcal{T}_t \rangle$ be a time dependency structure. Let $\mathbf{M}_s \subseteq \mathbf{M}$. Let $\mathcal{A} \in \mathbf{A}$, $\mathcal{M}_1, \mathcal{M}_2 \in \mathbf{M}_s$, $en^{\mathcal{M}_1}_{\mathcal{A}}, en^{\mathcal{M}_2}_{\mathcal{A}} \in \mathcal{E}$, and $\mathcal{A}, \mathcal{M}_1, \mathcal{M}_2 \sqsubset \mathcal{TDS}$. Let $s = X_1 X_2 \ldots X_n \in Sches(\mathcal{TDS})$.

If $\forall s_1 = X_i X_{i+1} \ldots X_j \in Sches(TDS\rfloor_s), \exists x \in \mathbf{A}$: $x \neq \mathcal{A} \wedge en^{\mathcal{M}_1}_{\mathcal{A}} \in X_i \wedge en^{\mathcal{M}_2}_{\mathcal{A}} \in X_j \wedge en^{\mathcal{M}_1}_{x} \in X_{i+1} \cup \cdots \cup X_j$, then $\mathcal{M}_1 \bigcirc_{\mathcal{A}} \mathcal{M}_2$ in $\mathcal{TDS}\rfloor_s$.

In the running example, because of $s_{run} = \{en^B_{c8}\}\{en^A_{c11}, en^B_{c7}\}\{en^A_{c7}, en^B_{c6}, en^J_{c6}\}\{en^J_B\} \in Sches(TDS)$, we have $s_{run1} = \{en^A_{c11}, en^B_{c7}\}\{en^A_{c7}, en^B_{c6}, en^J_{c6}\} \in Sches(TDS\rfloor_{s_{run}})$. Since $en^B_{c7} \in \{en^A_{c11}, en^B_{c7}\}$, $en^A_{c7} \in \{en^A_{c7}, en^B_{c6}, en^J_{c6}\}$, and $en^B_{c6} \in \{en^A_{c11}, en^B_{c7}, en^A_{c7}, en^B_{c6}, en^J_{c6}\}$, we have $A \bigcirc_{c7} B$.

The theorem states that we can decide whether two mobile objects are isolated from each other for a single ambient under a given schedule.

Theorem IV.2 Let $\mathcal{TDS}$ be a time dependency structure. Let $\mathbf{M}_s \subseteq \mathbf{M}$.

If $\forall \mathcal{M}, \mathcal{N} \in \mathbf{M}_s, \forall \mathcal{X} \in \mathbf{A}, \forall s \in Sches(\mathcal{TDS})$, $\mathcal{M}$ is isolated from $\mathcal{N}$ for $\mathcal{X}$ in $\mathcal{TDS}\rfloor_s$, then $\forall \mathcal{M}', \mathcal{N}' \in \mathbf{M}_s, \forall \mathcal{X}' \in \mathbf{A}_s : \mathcal{M}' \bigcirc_{\mathcal{X}'} \mathcal{N}'$ in $\mathcal{TDS}$.

Theorem IV.2 in fact shows that given the set of mobile objects and the set of ambients, we can decide whether multiple mobile objects are isolated from each other for multiple ambients under a given schedule by checking the available movement events of the states in a real-time mobile system.

V. CONCLUSION

A time dependency structure has been introduced and discussed. Based on the time dependency structure model, we have presented an approach for modeling a real-time mobile system. We have also defined a schedule for the isolation of mobile objects and have investigated the isolation schedulabilty analysis in a real-time mobile system. These results may be used for intelligent transportation systems and autonomous driving systems. In the future, we will further explore the isolation control and scheduling policies of the concurrent complex real-time mobile system. In practice, we will develop the scheduling policy generation method and wish it to be really used for autonomous driving.

REFERENCES

[1] Karine Altisen and et al. Scheduler modeling based on the controller synthesis paradigm. *Real-time Systems*, 23(1):55–84, 2002.

[2] Neil C. Audsley and et al. Deadline monotonic scheduling theory and application. *Control Engineering Practice*, 1(1):71–78, February 1993.

[3] Luca Cardelli and Andrew D Gordon. Mobile ambients. In *FoSSaCS*, pages 140–155. Springer, 1998.

[4] Jian-Min Jiang and et al. Event-based mobility modeling and analysis. *TCPS*, 1(2):9:1–9:32, February 2017.

[5] Jianmin Jiang and et al. Analyzing event-based scheduling in concurrent reactive systems. *ACM TECS*, 14(4):86:1–86:27, 2015.

[6] Kyoung-Dae Kim and P. R. Kumar. Cyber-physical systems: A perspective at the centennial. *Proceedings of the IEEE*, 100(1):1287–1308, May 2012.

[7] Qiao Li and R. Negi. Maximal scheduling in wireless ad hoc networks with hypergraph interference models. *IEEE Trans. Vehicular Technology*, 61(1):297–310, November 2012.

[8] Chung Laung Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, November 1973.

[9] Tadao Murata. Petri nets: properties, analysis, and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[10] John A. Stankovic and et al. *Deadline scheduling for real-time systems-EDF and related algorithms*, volume 460. Springer, 1998.

[11] Qinghui Tang and et al. A unified methodology for scheduling in distributed cyber-physical systems. *ACM TECS*, 11(S2), August 2012.

[12] Fumin Zhang and et al. Task scheduling for control oriented requirements for cyber-physical systems. In *Proceedings of the RTSS*, pages 47–56. IEEE, 2008.