Modelling Elevator System With Coloured Petri Nets

Mohammed Assiri, Mohammed Alqarni and Ryszard Janicki

Department of Computing and Software, McMaster University

ABSTRACT

In this paper, we present our experiences with modelling an elevator system based on Coloured Petri Net (CPN). With the implementation of several rules and algorithms. We introduce a standard models that assist simulation-based analysis of different algorithms. In this paper, it is mainly shown how elevator system are constructed and modelled into CPN models. A general structure is designed for purpose of being compatible with diverse stations. As a case study, a multi-elevator system is built in a multi-floors building. The results prove the compatibility and applicability of this model on different situations, and the expressive power and convenience of CPN.

INTRODUCTION

The elevator system is one of the software engineering benchmarks that are frequently used to test the expressive power, readability and convenience of various formal specification techniques. Petri Nets is one of these formal specification techniques.

Nevertheless all the previous models are either static, i.e. dependent on a particular values of numbers of elevators and floors (often one place was required for each elevator car), or the concept of colour as a data type was not fully utilized, or other formalisms as UML were substantially involved.

Our model is independent on the number of floors and elevators, it covers in substantial detail different stages of the. elevator system. We believe our model is flexible enough to be adapted to different algorithms and rules, and may eventually evolve in a 'standard' formal model of the elevator system.

THE ELEVATOR SYSTEM

… such systems are usually very complex as they require controlling multiple elevators by a centralized control mechanism. The complexity of those elevator arise from the need for scheduling, resource allocation, and stochastic control, to name a few. Handling these jobs usually results in systems behaving as discrete event systems …

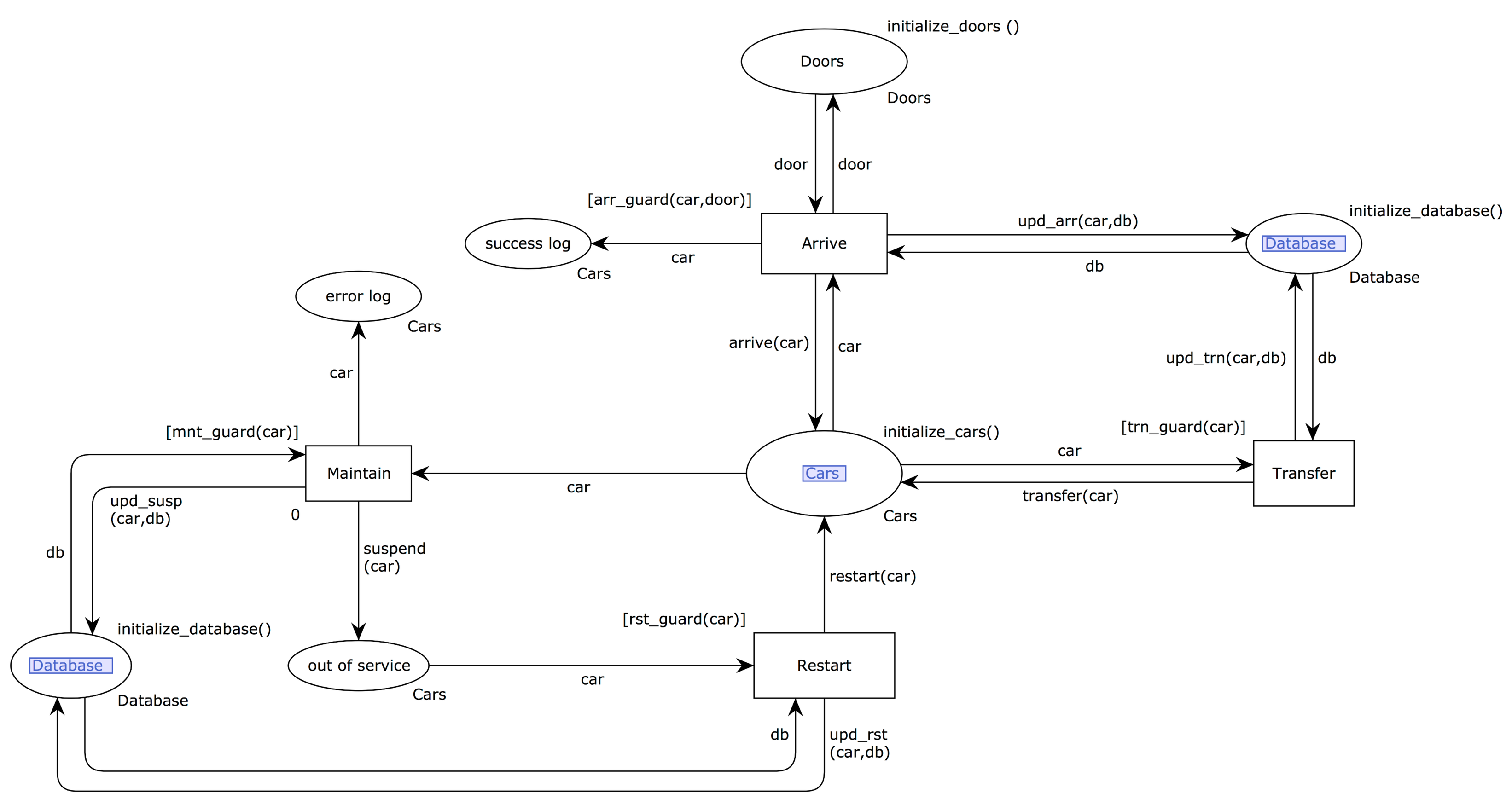
COLOURED PETRI NETS (CPN)

Coloured Petri Nets (CPN) is an extension of Petri Nets that is often used to model behaviours of rather complex systems. CPN preserves useful properties of Petri Nets and at the same time extends initial formalism to allow the distinction between tokens. Coloured Petri Nets is a graphical language for constructing models of concurrent systems and analyzing their properties. CPN is a discrete-event modelling language combining the capabilities of Petri nets with the capabilities of a high-level programming language. Petri nets provides the foundation of the graphical notation and the basic primitives for modelling concurrency, communication, and synchronization. CPN allows tokens to have a data value attached to them. Such attached data value is called token colour. Although the colour can be of arbitrarily complex type, places in CPN usually contain tokens of one type. This type is called colour set of the place.

CPN-BASED MODELLING OF ELEVATOR SYSTEM

Due to the complexity of the elevator system, and the desired flexibility of the structure, the proposed model is composed of five major (interconnected, but independent) sub-models. They are the car-structure sub-model, the hall-call sub-model, the car-call sub-model, the system-cycle sub-model, and the *hierarchical parking-optimizer model*. The functions and connection between models are can be described as follows: The car-structure sub-model represents the elevator cars. It is at the centre of all other sub-models that control the elevator's cars concurrently. Typically, an elevator car is requested through two types of controls, either a hall-call or a car-call. As names suggest, a hall-call is placed by pressing the button located in a hallway of a given floor while a car-call is place by pressing the button inside the car of the elevator. When a hall call is placed, the hall-call sub-model, relying on algorithms, will assign appropriately the hall call to a car of the car-structure sub-model. Similarly, the car-call sub-model coordinates the placed car calls with the cars of the car-structure sub-model. The system- cycle sub-model operates the cars of the car-structure sub-model to service the requested calls. Finally, the parking-optimizer model reduces the waiting time between the placing hall call and the arrival of the assigned car by electing constantly the holding floors of the idle cars.

The System-Cycle Sub-Model



ANALYSIS

… Two analysis techniques were applied. The first technique is the reachability analysis by means of the State Space tool. The tool verified and generated an automatic report. The proposed model has a number of dead markings that occur in cases such as a placed hall call with no appropriate car to be assigned to. The transition Maintain is a dead transition, which is an indicator that no error was catch by the The System-Cycle Sub-Model … The second technique is the simulation-based analysis by means of CPN Tools. This technique is flexible but also time-consuming. The model was simulated with two hundred thousand calls for each available algorithms, and the entire definition of the system was achievable.

PUBLICATION

This paper was accepted for publishing and presenting in SERP - Software Engineering Research and Practice Conference which is one of the top conferences in its filed. (http://www.worldacademyofscience.org)

ACKNOWLEDGMENT

assirims@mcmaster.ca :: m.assiri@sau.edu.sa