Summary

Modeling and analysis of cyber-physical systems are still challenging. One reason is that cyber-physical systems involve many different parts (cyber or physical), of different nature (discrete or continuous), and in constant interaction via sensing and actuating. For instance, consider a group of robots, each running a program that takes decision based on the sequence of sensor readings. The sensors that equip a robot return the current position of the robot and the position of any adjacent obstacle. The interaction occurring between each robot in the group cannot be derived solely from the specification of individual robots. If the field on which the robots roam changes its property, the same group of robots might sense different values, and therefore take different actions. Also, the time at which a robot acts and senses will affect the decision of each controller and will change the resulting collective behavior.

This thesis proposes a compositional approach to the design and programming of interacting cyber-physical components. Chapter 2 presents an algebra for interaction of cyber-physical components, where interactions in cyber-physical systems are expressed as parametrized algebraic operators. The model not only provides ways to compose components, but also allows, under certain conditions, for decomposition of components.

We instantiate our component model in Chapter 3 with a set of cyber components whose behaviors are independent of time. Specifically, we give an algebraic semantics of the Reo coordination language. The result adds some strength to existing work on Reo, by providing a suitable algebraic framework to show equivalence of connector behaviors in Reo. We consider temporal properties of Reo connectors, specifically to verify data flow properties of composite connectors.

In order to make our component model executable, in Chapter 4 we present several steps that lead to an operational yet compositional specification of components. First, we operationally define the specification of a component behavior using labeled

transition systems called a TES transition systems. We show that the behavior of a syntactic product of TES transition systems is the behavior of the semantic product of their corresponding two components. In order to have a finite executable specification of components, we give a specification of agents as rewrite theories, and show their compositional semantics.

To demonstrate the utility of our component model, in Chapter 5 we give an implementation of the rewriting framework in Maude. We define agent as a module that implements a set of operations for interacting with other agents. A system module runs all agents concurrently, and ensures composability of their actions. We present three main applications. First, we apply our model to the Reo coordination language, and give a simulation and verification framework. Next, we consider a system consisting of a controller, a valve, and N water reservoirs. The controller needs to move the valve in order to keep the water level in the reservoirs within some threshold. We contrast the safety property of the controller not seeing any invalid states with the possibility for the controller to miss some states. Finally, we analyse a system consisting of a set of robots, each equipped with a battery, running on a shared field. More specifically, we consider liveness and safety properties, such as the possibility for a robot to patrol through a set of points without running out of battery, or to coordinate with another robot to swap position and get in a sorted position.