International Journal of Innovative
Computing, Information and Control
Volume **7**, Number **7(A)**, July **2011**

ICIC International ©2011 ISSN 1349-4198
pp. **4067–4081**

COMPONENTWISE MODELLING AND SYNTHESIS OF DYNAMIC INTERACTIVE SYSTEMS USING THE EQUIVALENT TRANSFORMATION FRAMEWORK

Courtney Powell[1], Kiyoshi Akama[2] and Keisuke Nakamura[1]

[1]Graduate School of Information Science and Technology
[2]Information Initiative Center
Hokkaido University
Kita 11, Nishi 5, Kita-ku, Sapporo, Hokkaido 060-0811, Japan
{ kotoni; knakamura }@ist.hokudai.ac.jp; akama@iic.hokudai.ac.jp

Received February 2010; revised July 2010

Abstract. *The development and maintenance of Concurrent Systems, Reactive Systems and Dynamic Systems, remain fraught with challenges attributable to factors such as: 1) the tendency of concurrently executing processes to interact in unforeseen ways; 2) the necessity for Reactive Systems to facilitate continuous interactions with their environments and; 3) the constantly changing internal states of Dynamic Systems. It turns out that all three types of systems consist of a common subset, which accounts for most of the challenges. We term systems falling into this common subset, Dynamic Interactive Systems (DISs). We propose an incremental, componentwise, correct-by-construction approach to DIS development, using Equivalent Transformation (ET). This facilitates the construction of comprehensible DISs by decomposing global analysis of the DIS into local analysis of each component and its feasible interactions. To this end, we introduce two small, uniform, sets of ET rule types for comprehensively specifying DIS components, and their interaction patterns. We also outline our synthesis technique in which the properties of the model are reliably transferred to the actual implementation code. Finally, we demonstrate the efficacy of our approach by modelling and synthesizing a fully functional Web-based interactive application.*

Keywords: Equivalent transformation, Feasible interactions, Concurrent components, Dynamic interactive systems, Correctness by construction, Web applications

1. **Introduction.** Careful analysis of the existing literature on Concurrent Systems [1], Reactive Systems [2] and Dynamic Systems [3], reveals a group of systems common to all of them. This common subset consists of systems having multiple concurrent, interactive processes, which can continue to change state, even during periods when there are no inputs from the external environment. We refer to such systems as Dynamic Interactive Systems (DISs). Unlike transformational systems [2], which are considered correct if their final results are correct and they terminate; DISs produce results incrementally throughout their execution, and may not even be required to terminate. As a result, emergent misbehavior [4], such as deadlocks [5], livelocks [5] and race conditions [6], is difficult to find, as execution paths and behaviors may vary from one execution to the next, due to the arbitrariness of interaction among the concurrently executing processes. As a result, construction of DISs is time-consuming, and implemented DISs are error-prone and difficult to comprehend and analyze.

From Hoare [7], we know that: "There are two ways of constructing a software design: *One way is to make it so simple that there are obviously no deficiencies*, and the other way is to make it so complicated that there are no *obvious* deficiencies". The *Correct-by-*