International Journal of Innovative Computing, Information and Control
Volume **3**, Number **3**, June **2007**

ICIC International ©2007 ISSN 1349-4198
pp. **685–696**

EQUIVALENT TRANSFORMATION RULES AS COMPONENTS OF PROGRAMS

Hiroshi Mabuchi

Faculty of Software and Information Science
Iwate Prefectural University
152-52, Sugo, Takizawa, Iwate 020-0193, Japan
mabu@soft.iwate-pu.ac.jp

Kiyoshi Akama and Toshihiro Wakatsuki

Information Initiative Center
Hokkaido University
Sapporo, Hokkaido 060-0811, Japan
akama@iic.hokudai.ac.jp; aloha@uva.cims.hokudai.ac.jp

Received October 2006; revised January 2007

Abstract. *Recently, there has been increasing attention to Component-Based Software Development (CBSD). This development technology enhances productivity and reliability by combining existing verified components to construct a target software. This development technology, however, lacks methods to extract and preserve components; methods to retrieve query-satisfied components; and technologies and theories that guarantee the correctness of each component and the entire program. Considering that rule-based programming with theoretical framework provides good, theoretical supports to component-based programming and may solve these problems, this paper uses equivalent transformation (ET) programming that has advantages in the number of rules, granularity of rules, and guarantee of the correctness in each rule and the entire program over the conventional rule-based programming and discusses how to accumulate and use ET rules to, by employing ET rules as components, create a correct system from a system specification. ET programming is expected to overcome the problems the conventional CBSD and rule-based programming have.*
Keywords: Component, Computation model, Equivalent transformation, Correctness, Program construction

1. **Introduction.** As software systems are becoming larger and more complex, the need for the development of cost-effective and reliable software in a short period of time increases accordingly.

With such situations, there has been increasing attention to Component-Based Software Development (CBSD) [12, 13, 25].

CBSD is the development technology which promotes componentization of software systems, where complex software systems are constructed using reusable simpler components.