International Journal of Innovative Computing, Information and Control
Volume **5**, Number **12(A)**, December **2009**

ICIC International ©2009 ISSN 1349-4198
pp. 4655–4666

ET–BASED DISTRIBUTED COOPERATIVE SYSTEM

Hiroshi Yoshikawa[1], Kiyoshi Akama[2] and Hiroshi Mabuchi[3]

[1]Graduate School of Information Science and Technology
[2]Information Initiative Center
Hokkaido University
Kita-11 Nishi-5, Sapporo, 060-0811, Japan
josch@uva.cims.hokudai.ac.jp; akama@iic.hokudai.ac.jp

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

Received July 2008; revised December 2008

Abstract. *There are a great many advantages to hardware–software cooperative computing. As a result, cooperative systems are already being used in many areas. Constraint Satisfaction Problems (CSPs) are also expected to be efficiently solved by cooperative systems. However, we have trouble describing a CSP in system description languages (SDL) such as SystemC or SystemVerilog, because these SDL don't have enough power to describe CSPs, or they do have enough theoretical support for partitioning and mapping. We developed a framework in which a CSP can be described easily and a program can be partitioned into a distributed cooperation environment. In this framework, Equivalent Transformation (ET) language serves an important role. CSPs can be easily described in ET. Besides ET computation can be divided into a variety of parallel computations so that ET computation is easily mapped to a distributed cooperation system. Also we developed ET–HwNet Server System that enables ET programs to cooperate with FPGA boards, and demonstrate how the system accelerates computations.*

Keywords: Equivalent transformation, ET rules, HW/SW cooperative system

1. Introduction.

Hardware–software cooperative system can solve problems which contain complicated and large amount of parallel computation efficiently. *Constraint Satisfaction Problems* (CSPs) are expected to be efficiently solved by HW/SW cooperative systems because there might be many parallel computations in them. It is also possible to use *CSP solvers* for the purpose of solving CSPs. Research to accelerate them, e.g. [15], is actively being carried out. CSP solvers can deal with a wide range of problems. However it is difficult to use system resources effectively while utilizing characteristics of problems. Our aim is to establish a theoretical basis to synthesize cooperation systems which use their resource efficiently while utilizing characteristics of problems.

There is a lot of hard work involved in the development of such an optimal cooperative system. For example, to describe a problem with semantic correctness, to describe both software and hardware, and to develop an interface between them. In addition, challenges exist as to how to extract parallelism and also how much parallelism is important for efficient computation and to guarantee correctness, which is essential for computation. HwModule board [12] reduces the workload of interface development, however, a special skill is still required to develop cooperative systems. Besides conventional description