Title	ネットワーク実証実験環境の モジュール単位構築に関する研究
Author(s)	村上，正太郎
Citation	
Issue Date	2013-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/11316
Rights	
Description	Supervisor: 篠田　陽一，情報科学研究科，修士

Modular Construction of Network Experiments

Shotaro Murakami (1110060)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 6, 2013

Keywords: testbed,language-design,large-scale network.

There are many services on the Internet. The Internet is necessary for many people. In addition, the Internet is an important infrastructure. Many users are using a lot of services in the Internet. Therefore, before the introduction of new technologies, it is necessary to verify that new technologies do not affect other services and that they work fine. There is a practical verification method to build a large-scale network experiment envieoment, which enable us to verify the quanlity of the technologies, such as a new hardware/software mechanism. However, the construction cost of large-scale network experiment and the human cost was high. So, SpringOS, a network-experiment supporting software, has developed to build large-scale network experiments in low cost. The network experiment environment is constructed and execute with SpringOS's own language. The experimenters describes the flow of whole experiment and the role, of each component. However, the experimenter must describe one by one components of the network and nodes used in the network experiment. The experimenter need to describe the experimental system from lower layer. Therefore, the description tends to become complicated for the experimenter who wants to experiment on upper layer. In SpringOS, the experimenter defines a certain node as a nodeclass, and it is possible to generate nodes of same specification at once. When the experimenter describes a complex network experiment with specifications if many different nodes, quantity of a description increase every different node. Therefore, it takes time to create a script and the readability of is low.

In this research, I review the structure of the description language in a network experiment. I present a method to build a network experiment environment by a breaf description. In a method, the experimenter can focus on describing for an experiment object. It is possible to represent the entire network experiment environment in fewer descriptions. In consequence, the experimenter can build a network experiment environment with short time for script making. The experimenter can inspect the software using a network.

In this research, I define *Dodule(Defined Module)* as a combination of components used in a experiment environments. A Dodule is an abstraction of components elements of a network experiment environment or its part. By using Dodules, the experimenter can choose the description from lower to upper layer only. The experimenter can build a surrounding experiment environment of an experiment object by combining Dodules. As a result, the experimenter did not have to describe all experiment environments for each node and is able to focus on describing for an experiment object. In addition, I treat Dodule as object-oriented class. Dodule is able to use a various initial value by using a constructer. When I make Dodule instance, a class can be specified as a parameter. As a result, I improved the variety of the description.

M is the experiment description language which is able to use Dodule. M language is converted to script file that SpringOS can read by a translator. We compared the proposed technique and the description language of SpringOS were compared for description of the experimental environment construction. As a result of consideration, I confirmed improvement in reusability, flexibility and extensibility of the description by using Dodule. From these consideration, I became able to build the network experiment environment by a breaf description.

Furthermore, I discuss about unification of many experiment environment. Many experimenters build each experiment environment. They unity each experiment enviroment to large-scale network experiment environment. As a result, the experimenter can focus on each experiment environment. In addition, by abstracting the experimental environment, I considered reguired conditions when multiple experiment environments are integrated. I demonstrated the experiment environment unification using

a simple example.

Finally, discussions on fortune work is given.