Name	Laboratory Location	Research Area
Zhenjiang Hu, Professor	**National Institute of Informatics**	**Computer Software**

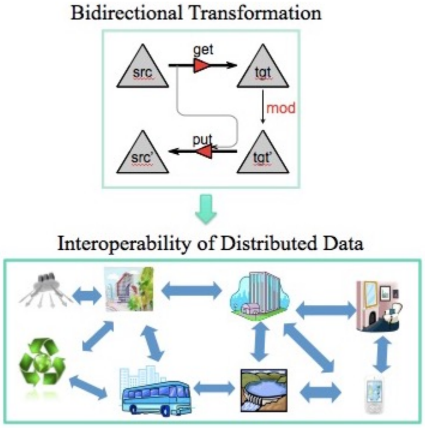
Open New Era of Software

We live in a world that runs on software; software is everywhere from the things you touch every day (laptops, smart phones, TVs, cars) to the infrastructure of society (hospitals, transportation, finance, government). Hu Lab (Programming Research Lab) explores and develops new programming theories, novel programming languages, and advanced programming tools for construction, implementation, testing, validation and verification of software. For details, please refer to the following web page for the researches and the activities of our laboratory.

`http://www.prg.nii.ac.jp`

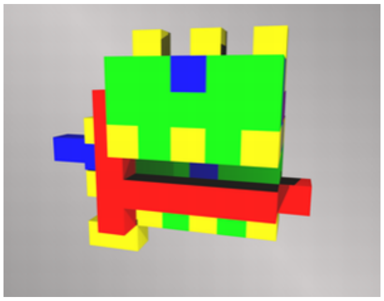
Bidirectional Programming and Data Interoperability

Bidirectional transformations, originated from the view updating mechanism in the database community, have been recently attracting a lot of attention from researchers in the communities of programming languages and software engineering. Bidirectional transformations provide a novel mechanism for synchronizing and maintaining the consistency of information between input and output, and have seen many interesting applications, including the synchronization of replicated data in different formats, presentation-oriented structured document development, interactive user interface design or coupled software transformation. We are designing bidirectional programming languages for development of efficient and correct bidirectional transformations, applying them to synchronization of different data, and establishing software foundations for controlling, integrating and coordinating distributed data.


Bidirectional Transformation
src
get
tgt
mod
put
src'
tgt'
Interoperability of Distributed Data

High-level Parallel Programming

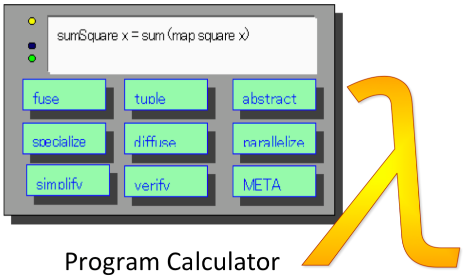
The problems involved in developing efficient and correct parallel programs have proved much harder than those in developing efficient sequential ones, both for programmers and for compilers. We have successfully construct a new parallelization framework, including a novel inductive synthesis algorithm and a calculation algorithm for parallelization of general recursive functions (on sequences and trees). We are now developing domain specific languages and efficient compilation methods for systematic construction of efficient parallel programs for processing big graphs.



Skeleton Parallel Programming

Mathematical Structures of Programs

Mathematical structure of programs plays a fundamental role in program development (programming). To clarify the structure of programs, we may take programs as mathematical objects, and programming as a mathematical activity, where mathematical reasoning, based on a small set of calculational laws such as fusion, tupling, accumulation, lies at the heart of the subject. With this view, many useful programming principles can be formulated so precisely that programmers can easily follow and apply them to practical program development. This is in sharp contrast to the prejudice of learning programming only by accumulating experience.



```
sumSquare x = sum (map square x)
```

Program Calculator