HILBERT: AN AUTONOMOUS EVOLUTIONARY INFORMATION SYSTEM FOR TEACHING AND LEARNING LOGIC[1]

Jingde Cheng, Naoki Akimoto, Yuichi Goto, Masato Koide, Kouichi Nanashima, Sinsuke Nara

ABSTRACT
In order to establish a widely useful computer-based environment for teaching and learning logic, we are developing an autonomous evolutionary information system, named "HILBERT," for teaching and learning various logic systems underlying diverse reasoning forms in scientific discovery as well as everyday logical thinking. This paper presents our basic design ideas for the system, facilities and services provided by the system, and the architecture of the system.

KEYWORDS
Reasoning and proof, conditional, entailment, classical mathematical logic, relevant logic, philosophical logic, autonomous evolution, computer-based teaching, computer-based learning

INTRODUCTION

Logic is a special discipline, which is considered to be the basis for all other sciences, and therefore, it is a science prior to all others, which contains the ideas and principles underlying all sciences [13, 26]. Because of the fundamental and abstract characteristics of logic, both teaching logic and learning logic are not so easy tasks. However, although there are a lot of tools and environments designed and developed for teaching and learning various sciences and technologies, until now, there are far fewer on logic [3]. This situation is not quite adequate to the important role that logic plays in modern science and technology. In order to establish a widely useful computer-based environment for teaching and learning logic, we are developing an autonomous evolutionary information system [9], named "HILBERT," for teaching and learning various logic systems underlying diverse reasoning forms in scientific discovery as well as everyday logical thinking. This paper presents our basic design ideas for the system, facilities and services provided by the system, and the architecture of the system.

REASONING AND PROVING

Reasoning is the *process* of drawing *new conclusions* from given premises, which are already known facts or previously assumed hypotheses (Note that how to define the notion of "new" formally and satisfactorily is still a difficult open problem until now). Therefore, reasoning is intrinsically ampliative, i.e., it has the function of enlarging or extending some things, or adding to what is already known or assumed. In general, a reasoning consists of a number of arguments (or inferences) *in some order*. An *argument* (or *inference*) is a set of declarative sentences consisting of one or more sentences as its premises, which contain the evidence, and one sentence as its conclusion. In an argument, a claim is being made that there is some sort of *evidential relation* between its premises and its conclusion: the

---

[1] This work is supported in part by The Ministry of Education, Culture, Sports, Science and Technology of Japan under Grant-in-Aid for Scientific Research on Priority Areas (No. 14022211).

conclusion is supposed to *follow from* the premises, or equivalently, the premises are supposed to *entail* the conclusion. Therefore, the correctness of an argument is a matter of the *connection* between its premises and its conclusion, and concerns the *strength* of the relation between them (Note that the correctness of an argument depends neither on whether the premises are really true or not, nor on whether the conclusion is really true or not). Thus, there are some fundamental questions: What is the criterion by which one can decide whether the conclusion of an argument or a reasoning really does follow from its premises or not? Is there the only one criterion, or are there many criteria? If there are many criteria, what are the intrinsic differences between them? It is logic that deals with the validity of argument and reasoning in general.

A *logically valid reasoning* is a reasoning such that its arguments are justified based on some *logical validity criterion* provided by a logic system in order to obtain correct (Note that here the term "correct" does not necessarily mean "true") conclusions from given premises. Today, there are so many different logic systems motivated by various philosophical considerations. As a result, a reasoning may be valid on one logical validity criterion but invalid on another. For example, the *classical account of validity*, which is one of fundamental principles and assumptions underlying classical mathematical logic and its various conservative extensions, is defined in terms of *truth-preservation* (in some certain sense of truth). It requires that an argument is valid if and only if it is impossible for all its premises to be true while its conclusion is false. Therefore, a classically valid reasoning must be *truth-preserving*. In general, for a deductive reasoning to be valid, it must be truth-preserving. On the other hand, for any correct argument in scientific reasoning as well as our everyday reasoning, its premises must somehow be *relevant* to its conclusion, and vice versa. The *relevant account of validity* is defined in terms of *relevance*. It requires that for an argument to be valid there must be some connection of meaning, i.e., some relevance, between its premises and its conclusion, among other things. Obviously, the relevance between the premises and conclusion of an argument is not accounted for by the classical logical validity criterion, and therefore, a classically valid reasoning is not necessarily relevant.

On the other hand, *proving* is the process of finding a justification for an explicitly specified statement from given premises, which are already known facts or previously assumed hypotheses. A *proof* is a description of a found justification. A *logically valid proving* is a proving such that it is justified based on some logical validity criterion provided by a logic system in order to obtain a correct proof.

The most intrinsic difference between reasoning and proving is that the former is intrinsically prescriptive and predictive while the latter is intrinsically descriptive and non-predictive. The purpose of reasoning is to find some new conclusion previously unknown or unrecognized, while the purpose of proving is to find a justification for some specified statement previously given. Proving has an explicitly given target as its goal while reasoning does not. Unfortunately, until now, many studies in Computer Science and Artificial Intelligence disciplines still confuse proving with reasoning.

LOGIC AND THE NOTION OF CONDITIONAL

What is logic? *Logic* deals with *what entails what* or *what follows from what*, and aims at determining which are the correct conclusions of a given set of premises, i.e., to determine which arguments are valid. Therefore, the most essential and central concept in logic is the logical consequence relation that relates a given set of premises to those conclusions, which validly follow from the premises. Based on different philosophical considerations, one can define different logical consequence relations, and therefore, result in different logic systems. To define a logical consequence relation is nothing but to provide a logical validity criterion for those arguments and reasoning considered to be correct in some certain sense of philosophical consideration.

In general, a *formal logic system* L consists of a formal language, called the *object language* and denoted by F(L), which is the set of all *well-formed formulas* of L, and a *logical consequence relation*, denoted by meta-linguistic symbol $\vdash_L$, such that for $P \subseteq F(L)$ and $c \in F(L)$, $P \vdash_L c$ means that within the framework of L, c is a valid conclusion of premises P, or that within the framework of L, given P as premises, c as a valid conclusion follows from P. For a formal logic system $(F(L), \vdash_L)$, a *logical* *theorem* t is a formula of L such that $\phi \vdash_L t$ where ϕ is the empty set. We use Th(L) to denote the set of all logical theorems of L. Th(L) is completely determined by the logical consequence relation $\vdash_L$. According to the representation of the logical consequence relation of a logic, the logic can be represented as a Hilbert style formal system, a Gentzen natural deduction system, a Gentzen sequent calculus system, or other type of formal system. A formal logic system L is said to be *explosive* if and only if $\{A, \neg A\} \vdash_L B$ for any two different formulas A and B; L is said to be *paraconsistent* if and only if it is not explosive.

In the literature of mathematical, natural, social, and human sciences, it is probably difficult, if not impossible, to find a sentence form that is more generally used for describing various definitions, propositions, and theorems than the sentence form of "if ... then ... ." In logic, a sentence in the form of "if ... then ..." is usually called a *conditional proposition* or simply *conditional* which states that there exists a relation of sufficient condition between the "if" part and the "then" part of the sentence. Scientists always use conditionals in their descriptions of various definitions, propositions, and theorems to connect a concept, fact, situation or conclusion to its sufficient conditions. Indeed, Russell 1903 has said, "Pure Mathematics is the class of all propositions of the form 'p implies q,' where p and q are propositions containing one or more variables, the same in the two propositions, and neither p nor q contains any constants except logical constants" [25].

In general, a conditional must concern two parts which are connected by the connective "if ... then ... " and called the *antecedent* and the *consequent* of that conditional, respectively. The truth of a conditional depends not only on the truth of its antecedent and consequent but also, and more essentially, on a necessarily relevant and conditional relation between them. The notion of conditional plays the most essential role in reasoning because any reasoning form must invoke it, and therefore, it is historically always the most important subject studied in logic and is regarded as the heart of logic [1]. In fact, from the age of ancient Greece, the notion of conditional has been discussed by the ancients of Greek. For example, the extensional truth-functional definition of the notion of material implication was given by Philo of Megara in the 4th century B.C. [17, 26].

When we study and use logic, the notion of conditional may appear in both the *object logic* (i.e., the logic we are studying) and the *meta-logic* (i.e., the logic we are using to study the object logic). In the object logic, there usually is a connective in its formal language to represent the notion of conditional, and the notion of conditional is also usually used for representing a logical consequence relation in its proof theory or model theory. On the other hand, in the meta-logic, the notion of conditional, usually in the form of natural language, is used for defining various meta-notions and describing various meta-theorems about the object logic.

From the viewpoint of object logic, there are two classes of conditionals. One class is empirical conditionals and the other class is logical conditionals. For a logic, a conditional is called an *empirical conditional* of the logic if its truth-value, in the sense of that logic, depends on the contents of its antecedent and consequent and therefore cannot be determined only by its abstract form (i.e., from the viewpoint of that logic, the relevant relation between the antecedent and the consequent of that conditional is regarded to be empirical); a conditional is called a *logical conditional* of the logic if its truth-value, in the sense of that logic, depends only on its abstract form but not on the contents of its antecedent and consequent, and therefore, it is considered to be universally true or false (i.e., from the viewpoint of that logic, the relevant relation between the antecedent and the consequent of that conditional is regarded to be logical). A logical conditional that is considered to be universally true, in the sense of that logic, is also called an *entailment* of that logic. Indeed, the most intrinsic difference between various different logic systems is to regard what class of conditionals as entailments, as Diaz pointed out: "The problem in modern logic can best be put as follows: can we give an explanation of those conditionals that represent an entailment relation?" [11]

For a formal logic system where the notion of conditional is represented by connective "$\Rightarrow$", a formula is called a *zero degree formula* if and only if there is no occurrence of $\Rightarrow$ in it; a formula of the form

$A \Rightarrow B$ is called a *first degree conditional* if and only if both A and B are zero degree formulas; a formula A is called a *first degree formula* if and only if it satisfies the one of the following conditions: (1) A is a first degree conditional, (2) A is in the form +B (+ is a one-place connective such as negation and so on) where B is a first degree formula, and (3) A is in the form B*C (* is a non-implicational two-place connective such as conjunction or disjunction and so on) where both of B and C is first degree formulas, or one of B and C is a first degree formula and another is a zero degree formula. Let k be a natural number. A formula of the form $A \Rightarrow B$ is called a *k^{th} degree conditional* if and only if both A and B are $(k-1)^{th}$ degree formulas, or one of A and B is a $(k-1)^{th}$ degree formula and another is a j^{th} ($j<k-1$) degree formula; a formula A is called a *k^{th} degree formula* if and only if it satisfies the one of the following conditions: (1) A is a k^{th} degree conditional, (2) A is in the form +B where B is a k^{th} degree formula, and (3) A is in the form B*C where both of B and C is k^{th} degree formulas, or one of B and C is a k^{th} degree formula and another is a j^{th} ($j<k$) degree formula. Let $(F(L), \vdash_L)$ be a formal logic system and k be a natural number. The *k^{th} degree fragment* of L, denoted by $Th^k(L)$, is a set of logical theorems of L which is inductively defined as follows (in the terms of Hilbert style formal system): (1) if A is a j^{th} ($j \leq k$) degree formula and an axiom of L, then $A \in Th^k(L)$, (2) if A is a j^{th} ($j \leq k$) degree formula which is the result of applying an inference rule of L to some members of $Th^k(L)$, then $A \in Th^k(L)$, and (3) Nothing else are members of $Th^k(L)$, i.e., only those obtained from repeated applications of (1) and (2) are members of $Th^k(L)$.

BASIC DESIGN IDEAS

The ultimate goal that we design and develop HILBERT system is to establish a widely useful computer-based environment for teaching and learning various logic systems underlying diverse reasoning forms in scientific discovery as well as everyday logical thinking. The major factors and ideas that we considered in designing HILBERT are as follows.

Users: The users of HILBERT that we considered are teachers and students of junior high schools, high schools, colleges and/or universities, and graduate schools. One of characteristics of HILBERT is the wide range of its users. We intend to make HILBERT really useful to its all users such that teachers and students from junior high schools to graduate schools can find their own interesting and challenging issues concerned with teaching and learning logic and its applications in various sciences and the real world. Although we consider the users of HILBERT as general and wide as possible but not pay our attentions to those users of any special area, we do expect that those researchers and scientists in the areas of Philosophical, Logic, Linguistics, Mathematics, Computer Science, Artificial Intelligence, Knowledge Science, and so on will find that HILBERT is a useful reference resource.

Contents: The contents of HILBERT for teaching and learning logic that we are preparing and will provide include logic puzzles and answers, explanations of logic connectives and their roles in everyday reasoning, elementary introduction to logic (what is logic and why study it), classical propositional calculus, classical predicate calculus, set theory, relevant logic, intuitionistic logic, modal logic, temporal logic, many-valued logic, deontic logic, epistemic logic, paraconsistent logic, conditional logic, linear logic, a dictionary of logic, a history of logic, and a logic bibliography [12, 17, 19]. For each logic, HILBERT provides all known formalizations of the logic, if any, such as Hilbert style formal system, Gentzen natural deduction system, Gentzen sequent calculus system, semantic tableau system, resolution system, and various exercises in different levels and their answers. An important one of our ideas and considerations is that we intend to provide users contents for teaching and learning various logic systems as logical validity criteria (in particular, the relevant account of validity [1, 2, 5, 8, 11, 20, 23, 24] as well as the classical account of validity) for reasoning as well as proving, teaching and learning various logic systems as different interpretations of the notion of entailment, and teaching and learning various logic systems as the most fundamental tools for scientific discovery.

Usages: We designed HILBERT such that it can serve as a partner for both teachers and students respectively and personally. A teacher or student can distantly, interactively, and personally use HILBERT as his/her own system through the Internet. HILBERT can also be used in the way of a group such that a teacher and his/her students share a topic and some contents and communicate each other through the Internet. A user can use HILBERT anywhere anytime by a web browser. HILBERT provides users with not only contents for teaching and learning logic but also facilities for monitoring, recording, and evaluating the processes and effects of teaching and learning.

Autonomous evolution: We designed HILBERT as an autonomous evolutionary information system. Unlike a traditional information system serving just as a storehouse of data or knowledge and working passively according to queries or transactions explicitly issued by users and application programs, an autonomous evolutionary information system serves as a partner of its users such that it discovers new knowledge from its database or knowledge-base autonomously, cooperates with its users in solving problems actively by providing the users with useful advices, and improves its own extent of 'knowing' and ability of 'working' evolutionarily [9]. HILBERT behaves differently toward different users and provides advices in different levels at different stage for the same user according to advances of the user.

System configuration: We designed HILBERT as a reconfigurable reactive system (a reactive system is a computing system that maintains an ongoing interaction with its environment, as opposed to computing some final value on termination) rather than a usual CAI system. The ideal running way of HILBERT is that its all function components can be reconfigurated and replaced without stopping its run [10].

FACILITIES AND SERVICES

In order to be a widely useful computer-based environment for teaching and learning logic through the Internet, HILBERT provides the following facilities and services for its users.

A logic puzzle database system: The database system stores various logic puzzles and answers, and explanations of logic connectives and their roles in everyday reasoning.

A logic course database system: The database system stores teaching and learning materials for all courses about various logic systems.

A logic reference database system: The database system stores all data of a dictionary of logic, a history of logic, and a logic bibliography.

A logic knowledge-base system: The system consists of a knowledge-base that stores all known formalizations, formal semantics, metatheorems, and open problems of various logic systems, and a forward reasoning engine for general-purpose entailment calculus named EnCal [6]. Many facilities and services provided by HILBERT are directly or indirectly provided by EnCal. Primitively, EnCal provides its users with the following major facilities [6]. For a logic L which may be a propositional logic, a first-order predicate logic, or a second-order predicate logic, a non-empty set P of formulas as premises, a natural number k (usually $k<5$), and a natural number j all specified by the user, EnCal can (1) reason out all logical theorem schemata of the k^{th} degree fragment of L $Th^k(L)$, (2) verify whether or not a formula is a logical theorem of the k^{th} degree fragment of L, if yes, then give the proof, (3) reason out all empirical theorems of the j^{th} degree fragment of the formal theory with premises P based on $Th^k(L)$, and (4) verify whether or not a formula is an empirical theorem of the j^{th} degree fragment of the formal theory with premises P based on $Th^k(L)$, if yes, then give the proof. Although EnCal is designed and implemented primitively for entailment calculi of relevant logics [1, 2, 5, 8, 11, 20, 23], it can also be used for entailment calculi of classical mathematical logic and its various classical or non-classical conservative extensions without problems in principle. HILBERT provides zero, first, and second degeree fragments for each formal logic system, and third or more high degeree fragments for some non-classical logic systems.

An intelligent adviser: The adviser cooperates with its users in learning logic autonomously and actively by providing the users with useful advices.

A group of automated tools for theorem proving and proof checking : The tools include automated theorem provers and proof checkers which can be obtained publicly such as Nqthm [4], ACL2 [15, 16], OTTER [18, 27-29], HOL [14, 21] and so on. These tools can be used by HILBERT itself for finding or checking a proof of a given theorem of classical mathematical logic.

A formula translator: The translator helps its users in translating natural language (English and Japanese) sentences into their logic formula representations, and also provides transformations between various representation forms of formulas.

A monitoring and recording system: The system monitors and records learning processes of all users.

An examing and evaluating system: The system examines and evaluates learning effects of all users.

A user database system: The system is used for managing all user data including user IDs, passwords, access authority data, the current levels, examination records, and so on.

A Web server: The Web server provides the user interface between HILBERT and its users. All operations by users on the use of HILBERT and almost all operations by system administrators on the management of HILBERT are performed indirectly by web browsers.

ARCHITECTURE

A *reactive system* is a computing system that maintains an ongoing interaction with its environment, as opposed to computing some final value on termination. An autonomous evolutionary information system should be a reactive system with the capability of concurrently maintaining its database or knowledge-base, discovering and providing new knowledge, interacting with and learning from its users and environment, and improving its own state of 'knowing' and ability of 'working'.

The term '*evolution*' means a gradual process in which something changes into a different and usually better, maturer, or more complete form. Therefore, the *autonomous evolution* of a system, which may be either natural or artificial, should be a gradual process in which everything changes by conforming to the system's own laws only, and not subject to some higher ones. Note that in order to identify, observe, and then ultimately control any gradual process, it is indispensable to measure and monitor the behavior of that gradual process.

Measuring the behavior of a computing system means capturing run-time information about the system through detecting attributes of some specified objects in the system in some way and then assigning numerical or symbolic values to the attributes in such a way as to describe the attributes according to clearly defined rules.

Monitoring the behavior of a computing system means collecting and reporting run-time information about the system, which are captured by measuring the system. Measuring and monitoring mechanisms can be implemented in hardware technique, or software technique, or both.

For any computing system, we can identify and observe its evolution, i.e., a gradual change process, only if we can certainly measure and monitor the system's behavior. Also, an autonomous evolutionary computing system must have some way to measure and monitor its own behavior by itself.

Figure 1 shows a reconfigurable architecture of HILBERT we designed based on the following fundamental principles [7, 10].

The dependence principle in measuring, monitoring, and controlling: "Any system cannot control what it cannot measure and monitor."

The wholeness principle of concurrent systems: "The behavior of a concurrent system is not simply the mechanical putting together of its parts that act concurrently but a whole such that one cannot find some way to resolve it into parts mechanically and then simply compose the sum of its parts as the same as its original behavior."

The uncertainty principle in measuring and monitoring concurrent systems: "The behavior of an observer such as a run-time measurer or monitor cannot be separated from what is being observed."

The self-measurement principle in designing, developing, and maintaining concurrent systems: "A large-scale, long-lived, and highly reliable concurrent system should be constructed by some function components and some (maybe only one) permanent self-measurement components that act concurrently with the function components, measure and monitor the system itself according to some requirements, and pass run-time information about the system's behavior to the outside world of the system."

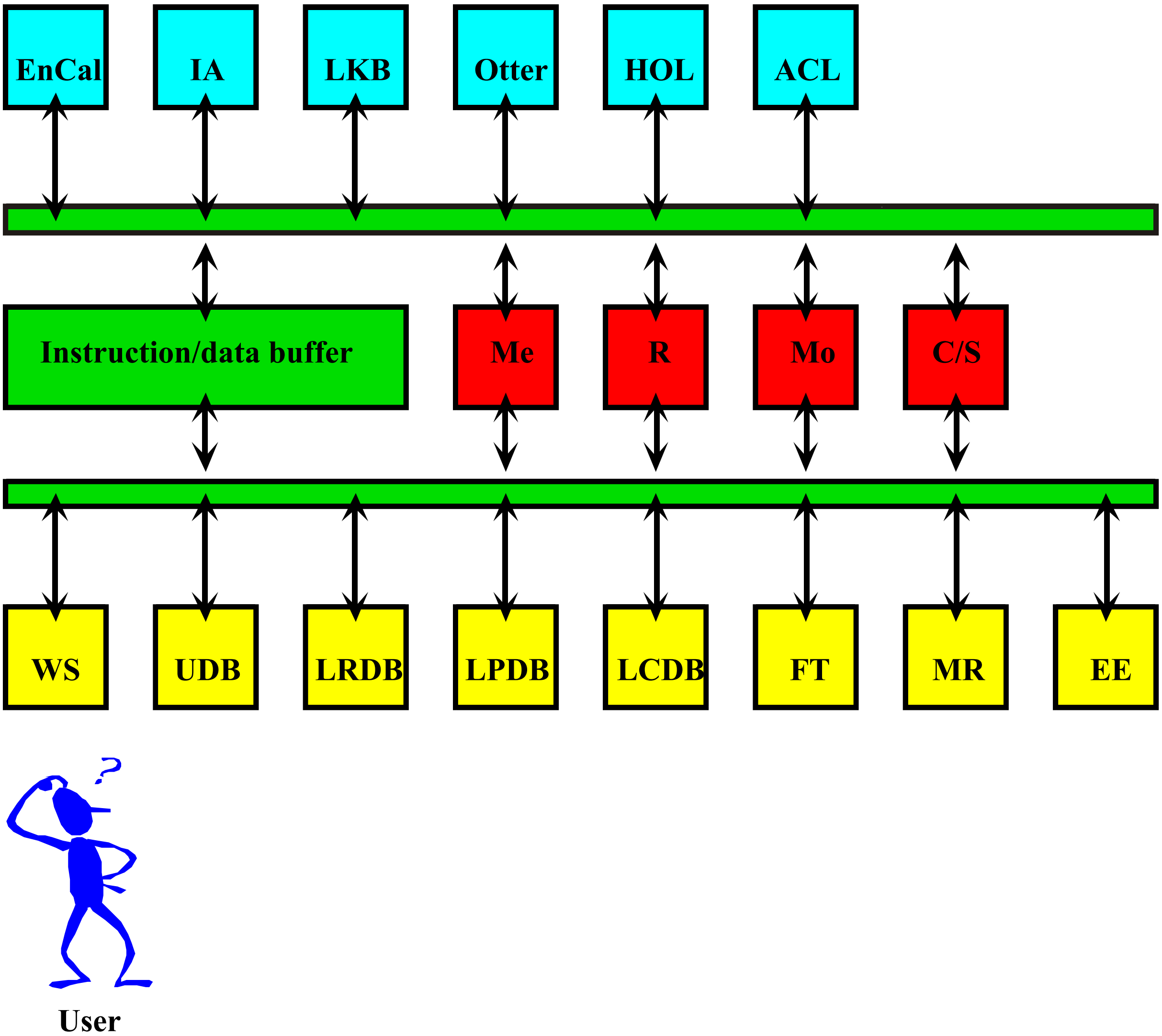

Figure 1. A reconfigurable architecture of HILBERT

The central components of HILBERT include a central measurer (Me), a central recorder (R), a central monitor (Mo), a central controller/scheduler (C/S), and an instruction/data buffer, all of which are permanent components of HILBERT. The functional components of HILBERT are separated into two groups which are measured, recorded, monitored, and controlled by the central components through two (internal and external) instruction/data buses, named “system buses” [10]. The central group of measurer, recorder, monitor, controller/scheduler components can be regarded as the “heart” and/or “brain” of HILBERT while the system buses can be regarded as “nerves” and/or “blood vessels” of HILBERT.

One group (named “internal group”) of the functional components of HILBERT includes EnCal, IA (intelligent adviser), LKB (logic knowledge-base system), and some automated tools for theorem proving and proof checking such as ACL, OTTER, HOL, and so on. These components of internal group can be directly invoked only by the central components since they are general-purpose components serving for the whole of HILBERT rather than individual other functional components or individual users. The functional components of another group can communicate with these components through the system bus and the instruction/data buffer. While any user of HILBERT cannot invoke these components of internal group directly.

Another group (named “external group”) of the functional components of HILBERT includes WS (Web user interface), UDB (user database system), LRDB (logic reference database), LPDB (logic puzzle database system), LCDB (logic course database system), FT (formula translator), MR (monitoring and recording system), and EE (examing and evaluating system). These components of external group can directly communicate with users through external instruction/data bus and the instruction/data buffer under the monitor and control of the central components.

The above architecture of HILBERT is reconfigurable because either an internal functional component or an external functional component can be easily added into or removed from the system.

CONCLUDING REMARKS

We have presented our basic design ideas for HILBERT system, facilities and services provided by the system, and the architecture of the system. The development of HILBERT system is an ongoing project. Although many implementation issues have been specified and are being implemented, some important implementation issues are still being investigated.

The ultimate goal that we design and develop HILBERT system is to establish a widely useful computer-based environment for teaching and learning various logic systems underlying diverse reasoning forms in scientific discovery as well as everyday logical thinking. Because logic plays the most fundamental role in all sciences and disciplines, such a widely useful environment will play an important role in the modern information society.

REFERENCES

A. R. Anderson and N. D. Belnap Jr., “Entailment: The Logic of Relevance and Necessity,” Vol. I, Princeton University Press, 1975.

A. R. Anderson, N. D. Belnap Jr., and J. M. Dunn, “Entailment: The Logic of Relevance and Necessity,” Vol. II, Princeton University Press, 1992.

J. Barwise and J. Etchemendy, “Language, Proof and Logic,” CSLI Publications, 2002.

R. S. Boyer and J. S. Moore, “A Computational Logic Handbook (2nd Edition),” Academic Press, 1998.

J. Cheng, “The Fundamental Role of Entailment in Knowledge Representation and Reasoning,” Journal of Computing and Information, Vol. 2, No. 1, pp. 853-873, 1996.

J. Cheng, “EnCal: An Automated Forward Deduction System for General-Purpose Entailment Calculus,” in N. Terashima and E. Altman (Eds.), “Advanced IT Tools, Proc. IFIP World Conference on IT Tools, IFIP 96 – 14th World Computer Congress,” pp. 507-514, CHAPMAN & HALL, 1996.

J. Cheng, “The Self-Measurement Principle: A Design Principle for Large-scale, Long-lived, and Highly Reliable Concurrent Systems,” Proc. 1998 IEEE-SMC Annual International Conference on Systems, Man, and Cybernetics, Vol. 4, pp. 4010-4015, 1998.

J. Cheng, “A Strong Relevant Logic Model of Epistemic Processes in Scientific Discovery,” in E. Kawaguchi, H. Kangassalo, H. Jaakkola, and I. A. Hamid (Eds.), “Information Modelling and Knowledge Bases XI,” pp. 136-159, IOS Press, 2000.

J. Cheng, “Autonomous Evolutionary Information Systems,” Wuhan University Journal of Natural Sciences, Vol. 6, No. 1-2, Special Issue: Proceedings of the International Software Engineering Symposium 2001, pp. 333-339, Wuhan University Journals Press, 2001.

J. Cheng and K. Nanashima, “System Bus: A Mechanism for Designing, Developing, and Maintaining Reconfigurable Reactive Systems,” Proc. International Conference on Systems, Development and Self-organization, pp. 232-236, 2002.

M. R. Diaz, “Topics in the Logic of Relevance,” Philosophia Verlag, 1981.

L. Goble (Ed.), “The Blackwell Guide to Philosophical Logic,” Blackwell, 2001.

K. Godel, “Russell’s Mathematical Logic,” in Schilpp (Ed.) “The Philosophy of Bertrand Russell,” Open Court Publishing Company, 1944.

M. J. C. Gordon and T. F. Melham, “Introduction to HOL – A Theorem Proving environment for Higher Order Logic,” Cambridge University Press, 1993.

M. Kaufmann, P. Manolios, and J. S. Moore, “Computer-Aided Reasoning: An Approach,” Kluwer Academic, 2000.

M. Kaufmann, P. Manolios, and J. S. Moore, “Computer-Aided Reasoning: ACL2 Case Studies,” Kluwer Academic, 2000.

W. Kneale and M. Kneale, “The Development of Logic,” Oxford University Press, 1962.

W. McCune and R. Padmanabhan, “Automated Deduction in Equational Logic and Cubic Curves,” Lecture Notes in Computer Science, Vol. 1095, Springer-Verlag, 1996.

W. Marciszewski (Ed.), “Dictionary of Logic as Applied in the Study of Language,” Martinus Nijhoff Publishers, 1981.

E. D. Mares and R. K. Meyer, “Relevant Logics,” in L. Goble (Ed.), “The Blackwell Guide to Philosophical Logic,” pp. 280-308, Blackwell, 2001.

T. Nipkow, L. C. Paulson, and M. Wenzel, “Isabelle/HOL: A Proof Assistant for Higher-Order Logic,” Lecture Notes in Computer Science, Vol. 2283, Springer-Verlag, 2002.

A. Quaife, “Automated Development of Fundamental Mathematical Theories,” Kluwer Acadamic, 1992.

S. Read, “Relevant Logic: A Philosophical Examination of Inference,” Basil Blackwell, 1988.

S. Read, “Thinking About Logic: An Introduction to the Philosophy of Logic,” Oxford University Press, 1995.

B. Russell, “The Principles of Mathematics,” Cambridge University Press, 1903, 2nd Edition, 1938, Norton Paperback Edition, Norton, 1996.

A. Tarski, “Introduction to Logic and to the Methodology of the Deductive Sciences (4th Edition, Revised),” Oxford University Press, 1941, 1946, 1965, 1994.

L. Wos, R. Overbeek, E. Lusk, and J. Boyle, “Automated Reasoning: Introduction and Applications (2nd Edition),” McGraw-Hill, 1992.

L. Wos, “The Automation of Reasoning: An Experimenter’s Notebook with OTTER Tutorial,” Academic Press, 1996.

L. Wos, “A Fascinating Country in the World of Computing: Your Guide to Automated Reasoning,” World Scientific, 1999.

Jingde Cheng, Naoki Akimoto, Yuichi Goto, Masato Koide, Kouichi Nanashima, and Sinsuke NaraDepartment of Information and Computer Sciences
Saitama University
Saitama, 338-8570, Japan
Email: {cheng, naoki, gotoh, koide, seven, nara}@aise.ics.saitama-u.ac.jp