A Laboratory for Basic Research in Computer Science and Software Engineering

Jian Zhang, Wenhui Zhang, Naijun Zhan, Yidong Shen, Haiming Chen,
Yunquan Zhang, Yongji Wang, Enhua Wu, Hongan Wang, Xueyang Zhu

State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing 100080, China
{zj,zxy}@ios.ac.cn

Abstract. The State Key Laboratory of Computer Science is committed to basic research in computer science and software engineering. The research topics of the laboratory include: concurrency theory, theory and algorithms for real-time systems, formal specifications based on context-free grammars, semantics of programming languages, model checking, automated reasoning, logic programming, software testing, software process improvement, middleware technology, parallel algorithm and parallel software, computer graphics and human-computer interaction. This paper describes these topics in some detail and summarizes some results obtained in recent years.

1 Introduction

The Laboratory of Computer Science (LCS) was established in 1993, supported by the Institute of Software, Chinese Academy of Sciences (ISCAS). It became a key laboratory of the Chinese Academy of Sciences (CAS) in 1994 and a State Key Laboratory in 2005.

Currently the State Key Laboratory of Computer Science (SKLCS) has more than 30 staff members, including 4 members of the Chinese Academy of Sciences. They have won numerous awards, the most prominent being: National Natural Science Awards, one first-class (1989), two second-class (1987, 1999); National Science and Technology Progress Awards, four second-class (1992, 2000, 2002, 2005).

The main research directions of SKLCS include:

- Theory and Application of Concurrent and Real-time Systems
  We study various computing phenomena with the characteristics of concurrency, timeliness and mobility. On the theoretical side, we focus on different approaches to the modeling, analysis and design of concurrent and real-time systems, including process algebras, temporal logics, timed automata, and so on. On the technology side, we try to invent new techniques for solving practical problems arising from distributed software, networking, life sciences, etc.
- Principles of Software Technology
  Research in this area focuses on theories and methodologies for the analysis and development of software, especially techniques for improving software quality. It

encompasses a broad range of subjects, from formal specification, program analysis and testing, middleware, to software process improvement, software quality measurement and assurance.

- Parallel Algorithm and Parallel Software
  Research in this area focuses on large-scale parallel numerical simulation, parallel computation model, non-numerical parallel computing, parallel processing of huge volume of data, high performance algorithms, performance optimization, performance evaluation and benchmarking.
- Computer Graphics and Human-Computer Interaction
  In computer graphics, we focus on realistic image synthesis, real time rendering of large and complex virtual environments, information representation and imaging, and animation. In Human-Computer interaction, we are interested in investigating the natural interaction between the user and the computer, especially ink understanding, multi-modal and context-aware interaction.

Our recent contributions in these fields are detailed in the following sections.

2 Concurrency and Model Checking

Some examples of concurrent systems are computer networks and communicating networks. The difficulties associated with concurrency have been tackled both through the construction of languages and concepts to make the complexity of concurrent execution manageable, and through the development of theories for reasoning about interacting concurrent processes. Theories of concurrency include process algebra, Petri nets and temporal logics. For verification of concurrent systems, one of the most practical methods is model checking.

Process algebras Research on process algebras conducted in SKLCS covers a wide range of topics, such as operational semantics, axiomatisation, verification algorithms and tools. One emphasis is on the "symbolic approach" to message-passing processes. Works along this line include:

- Extending "symbolic transition graphs with assignment" to incorporate complex data types as well as "void actions", which provides a theoretical framework for the implementation of a model checker for value-passing concurrent systems.
- Complete proof systems for weak bisimulations in the pi-calculus [1].
- Complete axiomatization of timed-bisimulation for timed automata (joint work with Wang Yi at Uppsala University, Sweden) [2].
- A tableau method for deciding weak bisimulation equivalence of context-free processes.

Petri nets Research on Petri nets conducted in SKLCS focuses on component-based methodology for the creation and verification of design specifications. Research results include (joint work with To-Yat Cheung at City University of Hong Kong and other researchers):

– Proposing a property-preserving place-merging approach for the synthesis of components [3].
– Investigating a kind of decompositions in terms of places, and providing conditions under which if the large system possesses some properties, then the resulting components also preserve the desirable properties without the need of further verification [3].
– Suggesting a property-preserving transition-reduction transformation to handle the synchronization problem in Petri-net-based system design, by first designing correct components without taking transition reduction consideration and then introducing synchronizations by merging transitions of components [5].
– Introducing three kinds of property-preserving subnet reduction methods for verifying system properties [4].
– Extending WF-nets (for workflow analysis) by considering resources and allowing for multi-cases in a process, and providing a compositional method to construct and verify sound WF-nets [6].

Model checking Research on model checking conducted in SKLCS focuses on techniques for alleviating the state explosion problem and for the verification of parameterized infinite state systems. Research results include:

– An approach for reducing the peak memory usage in model checking by first doing an analysis of a model with non-deterministic choices and then dividing the verification problem into sub-problems according to the underpinning LTL formulas that characterize different patterns of the executions of the model [7].
– Automating the "guard strengthening" method for parameterized model checking cache coherence protocols. During case studies a data inconsistency error was identified (and fixed) in the German 2004 protocol [8].
– Implementing a model checker for value-passing concurrent systems. Using this tool a number of case studies have been carried out.
– Applying model checking techniques to the testing of concurrent programs. A first-order extension of Computation Tree Logic has been proposed to describe testing purposes. Test cases are generated by slicing a system model against given formulas in the proposed logic [9].

Modal logics Modal logics are extensions of classical logics. Their application to the specification and verification of concurrent systems relies on adding temporal and space operators for the description of the execution and location of individual processes. In this area, we have obtained the following results:

– A predicate-based approach to spatial logics for the Ambient calculus and the pi-calculus, in which fix-point formulas are formed using predicate variables (instead of propositional variables) [10].
– A model checking algorithm for finite-control mobile processes against formulas in the spatial logics, which is the first decidability results for model checking finite-control mobile processes against spatial properties where both the processes and the logics admit recursions [11].

– Showing that the non-deterministic choice + is definable in the mu-calculus, so that we can compare the expressiveness between the mu-calculus with process algebra-like modal logics such as STL (Synchronization Tree Logic), for example, it was shown that the mu-calculus is as expressive as STL [12].
– Extending the definability of + to FLC (Fixpoint Logic with Chop), which is an extension of the mu-calculus, that is strictly more expressive than the mu-calculus, as FLC can express non-regular properties, and accordingly obtaining the compositionality of FLC, and as an application of the compositionality of FLC, an algorithm is given to construct characteristic formulae of BPA (Basic Process Algebra) up to strong (weak) bisimulation directly from the syntax of processes in a compositional manner [13].
– A bounded semantics for ACTL and a characterization of ACTL properties by propositional formulas [14].

3 Real-time and Hybrid Systems

A *real-time system* consists of several independent processes, which interact with each other and cooperate via communication in order to complete a common task in a given time. The correctness of such systems depends not only on the logical results of computing, but also on the time at which the computing was completed. Most real-time systems are embedded in a complex system or physical device as a subsystem, called *embedded systems*. Such kind of systems, normally containing two parts, i.e. continuous part and discrete part, are also called *hybrid systems*.

How to design and develop embedded systems, in particular safety critical systems, is a grand challenge for computer scientists and experts of control theory. In the past several years, members of our lab have contributed much to advance the state of the art of attacking the challenge, including formal approaches to design real-time and hybrid systems, real-time scheduling theories and real-time applications.

Formal Approaches to the Design of Real-Time and Hybrid Systems Duration Calculus (DC) [15], due to Zhou, Hoare and Ravn, is an extension of Interval Temporal Logic (ITL) by introducing *integral*. It has been proved by numerous case studies suitable for specifying and reasoning about real-time properties of systems. A most recent story about applications of DC is from the project AVCAS (see www.avacas.org) in which DC was used to specify and verify the real-time properties of several real-world examples [16], for instance, the European Train Control System (ETCS), the Traffic Collision Avoidance System (TCAS) in Avionics and Automatic Cruise Control (ACC) in Automotive. In the past few years, we further developed the approach both in theory and in application. On the theoretical side, Zhou and Hansen summarized and generalized the well-established work related to DC and published a monograph on DC by Springer [17]. Regarding the applications of DC, we made lots of attempts to formalize scheduling theories with DC and obtained some exciting results. For example, Xu and Zhan [18] pointed out and fixed a mistake on the Rate Monotonic Scheduler in Liu and Layland's seminal work [19].

In addition, Li and Tang [20] proposed a continuous linear temporal logic, called LTLC, which is an extension of XYZ/E [21]. A real-time system and its properties can be uniformly represented in the logical frame. The decidability issues about LTLC are investigated in [22].

Moreover, we also presented another formal semantics for Timed Automata (TA) which is a formal model of real-time systems, by combining its continuous semantics with discrete semantics [23]. The new formalism is called Finite Precision Timed Automata (FPTA). Compared with continuous and discrete TA, on one hand, the reachability of FPTA is equivalent to that of continuous TA, if only timing constraints with left close and right open are considered; on the other hand, the language accepted by a FPTA is decidable, the same as discrete TA.

Real-Time Scheduling Theories In the past years, we proposed lots of real-time scheduling algorithms to solve problems in practice and conducted their schedulability analysis. Some of them are listed as follows: DUC-LB [24] (Distributed Utilization Control for real-time clusters with Load Balancing) is a novel distributed utilization control algorithm for cluster-based soft real-time applications. Compared with earlier work on utilization control, a distinguished feature of DUC-LB is its capability to handle system dynamics caused by load balancing, which is a common and essential component of most clusters today. To measure internet bottlenecks in terms of location, capacity and available bandwidth, we proposed the BNeck [25] algorithm. Compared with existing algorithms, BNeck allows end users to efficiently and accurately measure the three bottleneck issues. In addition, several algorithms to resolve network blocking in soft real-time systems were invented. For example, [26] presented a congestion control algorithm based on an improved model in large-delay networks to design PID and PID-like controller; PSO-PID (Particle Swarm Optimization - Proportional-Integral-Differential) [27] is a novel controller for AQM (Active Queue Management) routers; and EVPSO (model of Particle Swarm Optimization with Escape Velocity) [28] equips particles with the escape velocity to avoid them trapped into local minima and increase the diversity of population, which outperforms the basic PSO, especially for high dimension function, and can be used to handle multi-modal optimization problems.

Real-Time Applications In the past few years, we also applied well-established theories and techniques of real-time and hybrid systems to solve practical problems, such as real-time monitoring of industrial processes, computer forensics, etc. In particular, we developed a large-scale real-time database system called Agilor.

4 Automated Reasoning and Logic Programming

In the last five years, we have done extensive research on automated reasoning and logic programming with emphasis on the termination of logic programming, tabulated resolution, decidability of a class of theory, satisfiability problem in the propositional logic and the first-order logic, and so on. In this section, we briefly describe our major achievements in these fields.

Termination of logic programs For a program in any computer language, in addition to having to be logically correct, it should also terminate in a finite amount of time. Due to frequent use of recursion in logic programming, however, a logic program may more likely be non-terminating than a procedural program. Termination of logic programs then becomes an important topic in logic programming research. Because the problem is extremely hard (undecidable in general), it has been considered as a never-ending story. Most existing termination analysis approaches rely on some *static* information about the structure of the source code of a logic program, such as modes/types, norms/level mappings, models/interargument relations, and the like. In contrast, we proposed a *dynamic* approach which employs some key dynamic features of an infinite (generalized) SLDNF-derivation, such as repetition of selected subgoals and recursive increase in term size [29]. We also introduced a new formulation of SLDNF-trees, called generalized SLDNF-trees. Generalized SLDNF-trees deal with negative subgoals in the same way as Prolog and exist for any general logic programs.

SLTNF-resolution Global SLS-resolution is a well-known procedural semantics for top-down computation of queries under the well-founded model. It inherits from SLDNF-resolution the *linearity* property of derivations, which makes it easy and efficient to implement using a simple stack-based memory structure. However, like SLDNF-resolution it suffers from the problem of infinite loops and redundant computations. To resolve this problem, we developed a new procedural semantics, called *SLTNF-resolution*, by enhancing Global SLS-resolution with loop cutting and tabling mechanisms [30]. SLTNF-resolution is sound and complete w.r.t. the well-founded semantics for logic programs with the bounded-term-size property, and is superior to existing linear tabling procedural semantics such as SLT-resolution.

Decidability of fragments of predicate logic In cooperation with Gilles Dowek of l'école polytechnique and INRIA, we give a new proof of a theorem of Mints that the positive fragment of minimal predicate logic is decidable. The idea of the proof is to replace the eigenvariable condition of sequent calculus by an appropriate scoping mechanism. The algorithm given by this proof seems to be more practical than that given by the original proof. A naive implementation is given in Objective Caml, version 3.08. Another contribution is to show that this result extends to a large class of theories, including simple type theory (higher-order logic) and second-order propositional logic. We obtain in this way a new proof of the decidability of the inhabitation problem for positive types in system F [31].

Satisfiability of logical formulas Many problems can be regarded as deciding the satisfiability of certain logical formulas. In fact, the satisfiability problem in the propositional logic, known as SAT, is a fundamental problem in computer science and artificial intelligence. Since 1960, many researchers have been working on SAT, and many algorithms have been proposed to solve the problem. Generally speaking, there are two classes of SAT solving methods, namely, those based on backtracking search and those based on local search. We have proposed a new way for combining the two methods [32].

In practice, many problems are more naturally described by a set of first-order formulas rather than propositional formulas. However, the satisfiability problem in the first-order logic is undecidable in general. To alleviate this difficulty, we can choose to check the satisfiability in finite domains. In other words, we may only look for finite models of first-order formulas. Given a set of first-order formulas and a positive number n, it is always possible to decide whether there is an n-element model of the formulas. This is usually done through backtracking search. But a straightforward search procedure is almost useless, because of its low efficiency.

An important way to increase the efficiency of finite model searching is to employ the symmetry in the search space. In the early 1990s, we have proposed an isomorphism elimination method called the *Least Number Heuristic (LNH)*. The LNH is very effective on many problems. But it does not help much if the problem has many predicate symbols. To remedy this weakness, we suggested and experimented with some techniques for eliminating isomorphic subspaces, based on Ramsey numbers [33]. More recently, we proposed a general-purpose technique, called *Decision Assignment Scheme Heuristic (DASH)*, which can completely eliminate isomorphism during the search [34].

5 Specification Acquisition and Reuse

MLIRF method [35] is a formal method for representation, acquisition, and reusing of formal specifications. The aim is to study how to assist human users by computer, through human-machine cooperation, to develop precise, complete and consistent formal specifications. Specifications are developed from human users' vague, incomplete and inconsistent informal statement of needs about target problems, together with, and making full use of, known specification knowledge. These specifications are approved by human users through verification and then used as the basis of software design and implementation (shown in Figure 1). An ultimate solution to the problem still needs much research effort. Here we present our current achievements. The essentials of MLIRF method are introduced as follows.

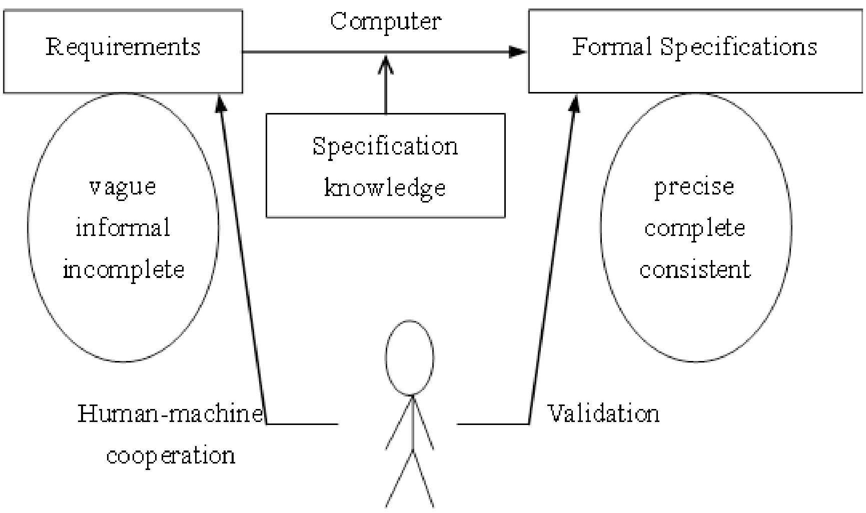

Fig. 1. Specification acquisition, reuse, and validation

The Representation of Formal Specifications A specification is split into two parts: a problem space and operations upon it, which respectively correspond to the syntactic and semantic aspects of the specification. The problem space or the syntactic part of a specification consists of syntactic descriptions of the problem class to be solved and its solution class. Each such description, called a concept, describes a set of all instances belonging to a class, and is represented as a context-free language. Operations or semantics of specifications are expressed by recursive functions on context-free languages (CFRF) [36, 37].

The Acquisition and Reuse of specifications It is possible to employ machine aided techniques in the acquisition of the specifications. It is known that context-free grammars can be inferred from samples, which is called grammatical identification. To make the acquisition of specifications effective, we need the algorithm with some features that existing algorithms do not have. A new method for grammatical identification is therefore proposed [38]. Acquisition of concepts is an interactive process which employs an integration of measures. It can either perform inductive learning to get grammar rules from representative samples of concepts, which produces grammars that can reflect naturally structures of the concepts, and works efficiently, or reuse known concepts in the learning process, or allow the specifier to give some pieces of concepts or edit the intermediate result of acquisition.

It is also possible to infer recursive functions, or assist in the construction of recursive functions. Since recursive function learning techniques are still premature in practice, we take a more practical way, in which the machine assists users to inductively construct definitions of functions according to grammars of the concepts. Reuse of specification knowledge is fused in the acquisition process.

Validation of Specifications The objective of specification validation is to assure that the specifications are in accordance with the intention of the specifiers. This can not be done automatically. What we can do is to provide techniques to help specifiers determine the validity of specifications. The measures provided for concept validation are as follows. The user may offer new samples conforming to their intention and check them against the obtained grammar, and, on the other hand, new samples of a grammar can be generated from the grammar as well for users to check if they are desired. The measure provided for the validation of operations is to perform the operations on samples to see if they conform to the user's original intention.

An experimental system SAQ (Specification AcQuisition system) has been implemented to support the MLIRF method [39]. Both a Solaris version and a Microsoft Windows version of SAQ have been developed. Several examples have been tried using SAQ, including formal specifications for classical algorithms on common data structures (such as list, stack, queue, tree, forest, graph, etc.), a Java to C++ convertor, a Basic to C convertor, a Lisp interpreter, XML processing, a music notation system, and formal differentiation of elementary functions.

6 Parallel Algorithm and Parallel Software

In recent years, the parallel computation group focuses on non-traditional structured grids and parallel computational model.

Non-traditional Structured Grids This concept is used in the research of the generalized Fast Fourier Transform (FFT) theories. By combination of the Approximation Theory and the Numerical Solution Methods for Partial Differential Equations (PDEs), we generalized the Fourier Methods to a class of irregular domains, such as arbitrary triangles and parallel hexagons in two dimensions [40–42], tetrahedrons and rhombic-dodecahedrons in three dimensions [42, 43]. The computational complexity is reduced from $O(N^4)$ to $O(N^2 \log N)$ for two dimension FFTs and from $O(N^6)$ to $O(N^3 \log N)$ for three dimension FFTs, respectively. The general algorithm framework (HFFT [44]) of the generalized FFTs for arbitrary simplices and super-simplices are set up and a public domain software package, FFTH, is developed.

Recently we found the internal relationship between the multivariate Fourier Transforms and the algebraic orthogonal polynomials in Numerical PDEs. Based on second order self-adjoint PDE eigen-problem, the definitions of orthogonal polynomials over many typical domains such as the cubes, the spheres, the simplices, the super-simplices and the curved-boundary simplices are established and many interesting facts such as the three layer recurrence formulas, the Chebyshev-like polynomials, the Gaussian cubatures, etc., are revealed.

Computational model RAM(h, k) [45–47] It is a computational model with considerations of instruction/thread level parallelism (k ways) and memory hierarchy (h levels) based on the memory access complexity concept [48] [1]. We argue that the complexity analysis of algorithms should include traditional computation complexity (time and space complexity) and new memory access complexity (to reflect the different memory access behavior on memory hierarchy of different implementations). Unlike the UMH model [49] that gives detailed architectural model and scheduling of data transfer, RAM(h, k) accounts for the different memory hierarchy level by their different real measurable memory access cost under different memory access patterns of an algorithm including temporal and spatial locality, contiguous and non-contiguous accesses, short message and long message, etc. So different implementations of one algorithm (usually with the same computation complexity) will have different memory access complexity under RAM(h, k) model. Thus we model the computer as one RAM with different memory access cost on each level under different memory access patterns.

Parallel computational model DRAM(h, k) model [45–47] It is the parallel version of RAM(h, k) model. It consists of p independent RAM(h, k) processors, each with its own local memory. They are connected at the memory level and communicate through message passing. It views message passing as another level of memory access to unify the analysis. The short message is viewed as non-contiguous memory access and long message is viewed as contiguous memory access. The cost of message passing is analyzed

[1] In the original paper, it is called memory complexity. To avoid confusion with space complexity, we changed it to memory access complexity later.

using the LogP model [50]. Thus sometimes it can be viewed as the combination of RAM(h,k) model and LogP model. DRAM(h,k) model pays more attention to the real measurable non-uniform access cost of local memory hierarchy and message passing.

7 Software Process

The expanding role of software and information systems in the world has focused increasing attention upon the need for assurances that software systems must be developed at an acceptable speed and cost, on a predictable schedule, and in such a way that the resulting systems are of acceptably high quality and can be evolved surely and rapidly as usage contexts change. This sharpened focus is creating new challenges and opportunities for software process technology. To address these challenges and opportunities, we have developed a series of innovative methods and techniques for software process. Through Software Process Workshop (SPW2005, SPW/ProSim2006), the First International Conference on Software Process (ICSP2007) and the 19^{th} International COCOMO Forum, they have spread widely. The results are summarized below.

Software Process Modeling TRidimensional Integrated SOftware development Model (TRISO-Model) integrates process management, development technology and human beings for software development processes. Based on it, a process assessment and improvement approach has also been proposed to evaluate software process models, software organizations, software projects and the software development processes in general [51–54]. Software processes are highly people-dependent. According to this character, an organization-entity holding certain capabilities is defined as the process agent, and an organizational entity capabilities based software process modeling method has been presented to construct the optimized software process model [55–60]. A graphical modeling language based on the process algebra has also been developed, through which the operational semantics of the software processes can be precisely described and the inherent mobility in the execution of software processes can be effectively modeled. Those theories have been successfully applied to the verification of software processes against specified properties and to the conformance checking between the process performance and the pre-defined standardized process [61, 62]. Based on the model driven architecture (MDA), the meta-model of different types of software processes, rules and the engine of transforming and merging among different meta-models have also been developed to merge different types of software processes [63, 64].

Empirical Software Engineering To cope with the uncertainties during software cost estimation, the COCOMO-U cost estimation model extending COCOMO II has been proposed. COCOMO-U is good at cost estimation and decision making in the early phase of software development when the project has various uncertainties [65, 66]. To improve cost estimation and assessment practices in Chinese government contract pricing, the COnstructive GOvernment cost MOdel (COGOMO) has been developed, which can provide guidances and insights for formal cost estimation [67]. Software measurement is an important facility to support effective and reasonable process management. Some process management and measurement methods have been proposed for controlling the process and improving the maturity level of an organization [68, 69]. An active measurement model (AMM) is developed for supporting software process improvement [70]. Based on AMM, software organizations can both design the appropriate measurement process according to the goals of the processes they focus and identify the opportunity and roadmap of improvement according to the result of measurement. In addition, a BSR (Baseline-Statistic-Refinement) method has also been proposed for establishing and refining the software process performance baseline, which has been applied in many software organizations and seems to be effective and efficient in the quantitative process management [68]. Productivity is a critical performance metric of process resources and an improved productivity time series prediction method named ARIMAmmse is developed, which can facilitate the software organizations to predict the trend of successive history productivity time series in an improved precision [71]. Software development task is the atomic unit of a software project and a novel software development tasks benchmarking approach based on data envelopment analysis (DEA) is developed for supporting the software developers to identify the high performance tasks under multivariate and variable return to scale requirements and to provide quantitative personal software process improvement suggestions [72]. Effective capability assessment of individual software processes is a key issue in the validation of the past and the adjustment of the future development processes. A novel combined method for capability assessment of individual software processes and practices is presented in [73]. Evaluation of project quality can lead to a better control of the schedule, cost and resources allocation and smooth the way for process improvement efforts. A DEA-based approach is proposed to evaluate the project quality [74].

Requirements Engineering A Tridimensional Requirements Model (TRISO-RM) has been proposed, which describes the win conditions of stakeholders on different aspects of software development. TRISO-Elements, each of which is formed by interconnected actors, an artifact, and an activity, are used as the medium to build and maintain the relationships among stakeholder win conditions. The production mechanism of model clashes is discussed and the process to find and avoid them is presented based on TRISO-RM [75]. A novel user-driven requirements elicitation method has been proposed. Based on users' individualities and context, support for users to present requirements is provided, including the recommendation of domain assets and advice about multi-user cooperation [76]. A risk-driven requirements prioritizing method which combines adaptive planning and risk-driven methodologies has been proposed. Requirements are prioritized adaptively, with risks as the foundation of priorities decisions [77]. A risk-driven release planning method for extreme programming has also been proposed. The method uses risks analysis to expose the strengths and weaknesses of a release plan and to improve developers' understanding of system goals and awareness of their abilities [78].

Tool Support Based on the methods and techniques mentioned above, an integrated software process services technology and management system has been developed. The system integrates quality assurance, development assistance and service support. It is quite extensible and easy to deploy. So far, the system has been successfully and widely applied in many software organizations including enterprises, universities and research institutes. This work won a second-class prize of the National Science and Technology Progress Award in 2005.

8 Distributed Computing and Middleware

The Distributed Computing and Middleware group focuses on developing a variety of middleware for distributed environments and tools supporting distributed system development. We work on the principles, technologies and systems for distributed computing. Our major research interests include:

- Distributed algorithms
- Key technologies for mobile computing environment
- Network-based content management technologies
- Models and technologies for adaptive middleware
- Service-oriented computing
- Component-based software methodology
- Model-driven architecture
- Testing and verification of distributed software

For the transactional composite service (TCS), we proposed a relaxed transaction model [79], including system mode, relaxed atomicity criterion, static checking algorithm and dynamic enforcement algorithm. With this model, the users are able to define different relaxed atomicity constraint for different TCS according to application-specific requirements, including acceptable configurations and the preference order. The checking algorithm determines whether the constraint can be guaranteed to be satisfied. The enforcement algorithm monitors the execution and performs transaction management work according to the constraint. Our approach can handle complex application requirements, avoid unnecessary failure recoveries and perform the transaction management work automatically.

We have also designed and implemented some software tools. For example, OnceAS/Q [80] is a web application server (WAS) with Quality of Service (QoS) guarantee. It provides a set of QoS related services and a framework to support QoS guarantee.

9 Computer Graphics

The major research areas of computer graphics group are virtual reality, realistic image synthesis and scientific visualization. Since 1990's the group has been doing research in these areas. Some results are summarized in the following:

- In the early nineties when the global illumination of radiosity technology was developed in its infantry stage, for the first time, a novel approach on the solution for bump texture generation by radiosity calculation was proposed. The work solved one of the crucial problems in the development of radiosity technology, and was accepted by ACM/SIGGRAPH90 [81]. At the same time, the radiosity solution to the generation of other related geometries such as furry surfaces, fractal surfaces was also proposed [82, 83]. By application of novel techniques in global illumination and image based modeling to natural objects and environment, excellent contribution has been made to rendering special effect such as knitwear, snowy scenery etc [85, 84]. Study on visualization of scalar data also started in the late nineties [86–89].

– The advance of graphics processing technology and hardware in recent years has been strongly pushing the progress of real time processing of realistic image synthesis. The computer graphics group has focused the investigation on this aspect, and has contributed mainly to two subjects: physically based modeling and animation, and graphics processing to large environments.

  On the physically based modeling and rendering, work has been conducted on the simulation of fluid dynamics based on contemporary graphics processing unit (GPU). In particular, design and implementation has been made on real time simulation of fluid dynamics for medium size environment with arbitrary boundary conditions, for the first time, fulfilled on GPU [90–92]. More realistic interaction has been simulated between fluid & solid substrate and among fluid mixtures between different features of fluids [93–95]. These works have been presented at important journals or conferences. We have also been invited as keynote speakers by prestigious international conferences like CASA2004 [90] and CyberWorlds2006 [95].
– On the subject of geometric processing to large out-of-core models, we proposed an approach which is recognized as one of the pioneering works for non-uniform simplification to out-of-core models [96].
– On the fundamental research for geometric processing, a geometric subdivision method was proposed for inclusion test of points within a geometry expressed as polygons(2d) or Polyhedrons(3d). The work has been accepted by IEEE Transaction and European publications [97, 98].

10 Human-Computer Interaction Techniques

Ink processing based on pen-paper metaphor Pen and paper are natural and effective interaction patterns used by people for a long time. The Human-Computer interaction group focuses on digital ink description based on temporal data model and ink understanding based on cognitive psychology. We proposed an ink structure understanding algorithm for handwriting digital ink, which can separate textual ink from graphical ink, recognize paragraphs, segment lines and extract characters. Based on the ink understanding algorithm, we proposed a method to correct handwriting recognition errors by speech. Users can repeat the text by speech to correct several character recognition errors in a line of Chinese handwriting. Our results were presented at ACM IUI [99–101].

Visualization of dynamic cursor We explored the visualization method of pen tilt information in pen-based user interface and its influence on interaction process and proposed a method to generate cursor shape on the computer screen which visualizes pen tilt information. Empirical evaluation shows that the method can enhance the stimulus-response compatibility of touchpad in pen-based user interface. This work was presented at ACM CHI 2007 [102].

Large hierarchical data visualization We proposed a circle packing technique to visualize large hierarchical data. Compared with other methods, it can show both data hierarchy and single data nodes. It provides a new way to visualize large volumns of hierarchical data. This work was presented at ACM CHI 2006 [103].

11 Concluding Remarks

Computer Science is a young field and is evolving rapidly. In addition to the major areas outlined in the previous sections, we are also interested in other research topics such as algorithms and complexity [104, 105], software testing [106, 107], data mining [108], semantic web, quantum computing, and so on.

We hope that our results will have a significant impact on the basic research in computer science and will also be helpful to Chinese software industry.

References

1. H. Lin (2003), Complete Inference Systems for Weak Bisimulation Equivalences in the π-calculus, Information and Computation, Vol.180, pp.1-29.
2. H. Lin and Y. Wang (2002), Axiomatising Timed Automata, Acta Informatica, Vol38, pp.277-305.
3. L. Jiao, H. J. Huang, and T. Y. Cheung (2005), Property-preserving Composition by Place Merging. Journal of Circuits, Systems and Computers, Vol. 14, No. 4, pp. 793-812
4. H. J. Huang, L. Jiao and T. Y. Cheung (2005), Property-preserving Subnet Reductions for Designing Manufacturing Systems with Shared Resources. Theoretical Computer Science, Vol. 332(1-3), pp. 461-485.
5. L. Jiao, T. Y. Cheung and W. M. Lu (2005), Handling Synchronization Problem in Petri-net-based System Design by Property-preserving Transition-reduction. The Computer Journal, Vol.48, No. 6, pp.692-701.
6. L. Jiao and T. Y. Cheung (2006), Compositional Verification for Workflow Nets. Journal of Circuits, Systems and Computers, Vol. 15, No. 4, pp.551-570.
7. W. Zhang (2003), Combining Static Analysis and Case-Based Search Space Partitioning for Reducing Peak Memory in Model Checking. Journal of Computer Science and Technology 18(6):762-770.
8. Y. Lv, H. Lin, H. Pan. Computing invariants for parameter abstraction. Fifth ACM-IEEE International Conference on Formal Methods and Models for Codesign(MEMOCODE'07), Nice, France, 2007.
9. P. Wu and H. Lin. Model-Based Testing of Concurrent Programs with Predicate Sequencing Constrains. International Journal of Software Engineering and Knowledge Engineering. Vol.16, No.5 (2006) pp.727-746, World Scientific
10. H. Lin, A Predicate mu-Calculus for Mobile Ambients. Journal of Computer Science and Technology, Vol. 20, No. 1, pp. 95 - 104.
11. H. Lin, A Predicate Spatial Logic and Model Checking for Mobile Processes. LNCS 3407 (ICTAC 2004): 36-36.

12. N. Zhan, M. Majster-Cederbaum, Deriving Non-determinism from Conjunction and Disjunction, LNCS 2896 (FORTE 2005), pp.351-365. 2005.
13. N. Zhan, J. Wu, Compositionality of Fixpoint Logic with Chop, LNCS 3722 (ICTAC 2005), pp.136-150. 2005.
14. W. Zhang. Model Checking with SAT-based Characterization of ACTL Formulas. Proceedings of the 9th International Conference on Formal Engineering Methods, LNCS 4789:191-211. 2007.
15. C.C. Zhou, C.A.R. Hoare and A. Ravn. A calculus of durations. Information Processing Letters,1991,40(5):269-276
16. W. Damm, H. Hungar and E.-R. Olderog. Verification of Cooperating Traffic Agents. International Journal of Control,2006,79(5):395-421
17. C.C. Zhou and M.R. Hansen. Duration Calculus: A Formal Approach to Real-Time Systems. EATCS Series of Monographs in Theoretical Computer Science, Springer, 2004
18. Q.W Xu and N.J. Zhan. Formalizing scheduling theories with Duration Calculus. To appear in Nordic Journal of Computing
19. C.L. Liu and J.W. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. J. ACM, 1973,20(1):46-61
20. G.Y. Li, Z.S. Tang. Modeling real-time systems with continuous-time temporal logic. In: Proceedings of ICFEM'02, LNCS2495,2002:231-236
21. Z.S. Tang, XYZ. A program development support enviroment based on temporal logic. In: Proceedings of Programming Language and System Design. North Hollland, 1983.
22. G.Y. Li, Z.S. Tang. Translating a continuous-time temporal logic into timed automata. In: Proceedings of APLAS'03, LNCS2895,2003:322-338
23. R.J. Yan, G.Y. Li, Z.S. Tang. Symbolic model checking of finite precision timed automata. In: Proceedings of ICTAC'05,LNCS3722,2005:272-287
24. Y. Fu, H. Wang. Distributed Utilization control for real-time clusters with load balancing. In: Proceedings of IEEE RTSS'06, Rio, Brazil, 2006.
25. H. Zhou, Y.J. Wang, Q. Wang. Measuring internet bottlenecks: location, capacity, and available bandwidth. In: Proceedings of International Conference on Computer Network and Mobile Computing, LNCS3619, 2005:1052-1062
26. X.X. Wang, Y.J. Wang, J.H. Zhou, X.L. Wang. Congestion Control Algorithm Based on Improved Model in Large-Delay Networks. Acta Electronica Sinica, 2005,33(5):842-846
27. X.L. Wang, Y.J. Wang, H. Zhou, X.Y. Huai. PSO-PID: a novel controller for AQM routers. In: Proceedings of IEEE/IFIP WOCN'06,2006:1-5
28. X.L. Wang, Y.J. Wang, H.T. Zeng, H. Zhou. Particle swarm optimization with escape velocity. In: Proceedings of CIS'06, IEEE Press,2006:457-460
29. Y. D. Shen, J. H. You, L. Y. Yuan, S. P. Shen and Q. Yang. A dynamic approach to characterizing termination of general logic programs. ACM Transactions on Computational Logic,2003,4(4):417-430
30. Y. D. Shen, J. H. You and L. Y. Yuan. Enhancing global SLS-resolution with loop cutting and tabling mechanisms. Theoretical Computer Science,2004,328(3):271-287
31. G. Dowek, Y. Jiang. Eigenvariables, bracketing and the decidability of positive minimal predicate logic. Theoretical Computer Science,2006,360(1-3):193-208
32. W. Zhang, Z. Huang and J. Zhang, Parallel Execution of Stochastic Search Procedures on Reduced SAT Instances, Proc. PRICAI 2002: 108-117.
33. X.X. Jia and J. Zhang, Predicate-Oriented Isomorphism Elimination in Model Finding, Proc. IJCAI 2005: 1525-1516.
34. X.X. Jia and J. Zhang, A Powerful Technique to Eliminate Isomorphism in Finite Model Search, Proc. IJCAR 2006: 318-331.
35. Y. Dong. MLIRF method for specification acquisition and reuse. (in Chinese) Proc. of the 9th National Conf. of China Computer Federation. 21-27, May 1996.

36. Y. Dong, Recursive functions of context free languages (I) – The definitions of CFPRF and CFRF, Science in China Series F 45(1) (2002) 25–39.
37. Y. Dong, Recursive functions of context free languages (II) – Validity of CFPRF and CFRF definitions, Science in China Series F 45(2) (2002) 1-21.
38. Y. Dong, An interactive learning algorithm for acquisition of concepts represented as CFL, J. Comput. Sci. & Technol. 13(1) (1998) 1-8.
39. Y. Dong, K. Li, H. Chen, et al., Design and implementation of the formal specification acquisition system SAQ, Conf. Software: Theory and Practice, IFIP 16th World Computer Congress 2000. Beijing, 2000, 201-211.
40. J.C. Sun. Multivariate Fourier Series Over A Class Of Non Tensor-Product Partition Domains. Journal of Computational Mathematics,2003,21(1):53-62
41. J.C. Sun and J.F. Yao. Fast Generalized Discrete Fourier Transforms on Hexagon Domains. Chinese J. of Numer. Math. Appl.,2004,26(3):351-366
42. J.C. Sun and H.Y. Li. Generalized Fourier transform on an arbitrary triangular domain. Advances in Computational Mathematics,2005,22:223-248
43. J.F. Yao and J.C. Sun. HFFT on Parallel Dodecahedron Domains and Its Parallel Implementation. Numerical computation and computer applications (in Chinese), 2004,4:304-313
44. J.C. Sun. Multivariate Fourier Transform Methods over Simplex and Super-simplex Domains. Journal of Computational Mathematics,2006,24(3):305-322
45. Y.Q. Zhang. Performance optimizations on parallel numerical software package and study on memory complexity. Ph.D. Thesis, Institute of Software, Chinese Academy of Sciences, Beijing, P.R. China , 2000(in Chinese).
46. Y.Q. Zhang. DRAM(h): A Parallel Computation Model for High Performance Numerical Computing. Chinese Journal of Computers, 2003, 26(12): 1660-1670
47. Y.Q. Zhang, G.L. Chen, G.Z Sun, Q.K. Miao. Models of Parallel Computation: A Survey and Classification, Frontiers of Computer Science in China, Vol.1, No.2,pp.156-165, Higher Education Press and Springer-Verlag, May 2007.
48. Y.Q. Zhang *etc*. Memory Complexity in High Performance Computing. In: Proceedings of the Third International Conference on High Performance Computing in Asia-Pacific Region, Singapore,1998:142-151
49. B. Alpern, et al. The Uniform Memory Hierarchy Model of Computation, Algorithmica, 1994, 12(2/3): 72-109.
50. D.Culler, etc. LogP: Towards A Realistic Model of Parallel Computation. In: Proceedings of PPoPP'93, San Diego, California, United States, May, 1993:1-12,
51. M. Li, Expanding the Horizons of Software Development Processes: A 3-D Integrated Methodology, in Software Process Workshop 2005, Beijing, China, 2005, pp. 54-67.
52. M. Li, Assessing 3-D Integrated Software Development Processes: A New Benchmark, in SPW/ProSim 2006, 2006, pp. 15-38.
53. Q. Wang and M. Li, Software Process Management: Practices in China, in Software Process Workshop 2005, Beijing, China, 2005, pp. 317-331.
54. M. Li, TRISO-Model: A New Approach to Integrated Software Process Assessment and Improvement, Journal of Software Process: Improvement and Practice, vol. 12, pp. 387-398, 2007.
55. Q. Wang, J. Xiao, M. Li, M. W. Nisar, R. Yuan, and L. Zhang, A Process-Agent Construction Method for Software Process Modeling in SoftPM, in SPW/ProSim 2006, Shanghai China, 2006, pp. 204-213.
56. N. Li, M. Li, Q. Wang, and S. Du, A Negotiation Model for the Process Agent in an Agent-Based Process-Centered Software Engineering Environment, in SEKE'06, San Francisco.California, pp. 664-669.

57. X. Zhao, K. Chan, and M. Li, Applying Agent Technology to Software Process Modeling and Process-Centered Software Engineering Environment, in The 20th Annual ACM Symposium on Applied Computing(SAC'05), Santa Fe, New Mexico, USA, 2005, pp. 1529-1533.
58. J. Xiao, L. J. Osterweil, L. Zhang, A. Wise, and Q. Wang, Applying Little-JIL to Describe Process-Agent Knowledge in SoftPM, in SPW/ProSim 2006, Shanghai, China, 2006, pp. 214-221.
59. L. Zhang, Q. Wang, J. Xiao, L. Ruan, L. Xie, and M. Li, A Tool to Create Process-Agents for OEC-SPM from Historical Project Data, in ICSP2007, 2007, pp. 84-95.
60. J. Xiao, L. J. Osterweil, L. Zhang, A. Wise, and Q. Wang, Applying Little-JIL to describe Process-Agent knowledge and support project planning in SoftPM Journal of Software Process: Improvement and Practice, vol. 12, pp. 437-448, 2007.
61. M. Li, Q. Yang, J. Zhai, and G. Yang, On Mibility of Software Processes, in SPW/ProSim 2006, Shanghai, China, 2006, pp. 105-114.
62. Q. Yang, M. Li, Q. Wang, GuoweiYang, J. Zhai, J. Li, L. Hou, and Y. Yang, An Algebraic Approach for Managing Inconsistencies in Software Processes, in ICSP2007, 2007, pp. 121-133.
63. F. Yuan, M. Li, and Z. Wan, SPEM2XPDL-Towards SPEM Model Enactment, in SERP' 06, Las vegas.Nevada, pp. 240-245.
64. J. Li, M. Li, Z. Wu, and Q. Wang, A SPEM-based Software Process Metamodel for CMM, Journal of Software, vol. 15, pp. 960-974, 2004.
65. D. Yang, Y. Wan, Z. Tang, J. Wu, M. He, and M. Li, COCOMO-U: An Extension of COCOMO II for Cost Estimation with Uncertainty, in SPW/ProSim 2006, Shanghai, China, 2006, pp. 132-141.
66. D. Yang, B. Boehm, Y. Yang, Q. Wang, and M. Li, Coping with the Cone of Uncertainty: An Empirical Study of the SAIV Process Model, in ICSP2007, 2007, pp. 37-48.
67. M. He, Y. Yang, Q. Wang, and M. Li, Cost Estimation and Analysis for Government Contract Pricing in China, in ICSP2007, 2007, pp. 134-146.
68. Q. Wang, N. Jiang, L. Gou, X. Liu, M. Li, and Y. Wang, BSR: a statistic-based approach for establishing and refining software process performance baseline, in ICSE 2006, Shanghai, China, 2006, pp. 585-594.
69. Q. Wang and M. Li, Measuring and Improving Software Process in China, in Proceedings of the International Symposium on Empirical Software Engineering (ISESE), 2005, pp. 183-192.
70. Q. Wang, M. Li, and X. Liu, An Active Measurement Model for Software Process Control and Improvement, Journal of Software, vol. 16, pp. 407-418, 2005.
71. L. Ruan, Y. Wang, Q. Wang, F. Shu, H. Zeng, and S. Zhang, ARIMAmmse: An Improved ARIMA-based Software Productivity Prediction Method, in 30th Annual International Computer Software and Applications Conference (COMPSAC 2006), Chicago, Illinois, 17-21 September 2006.
72. L. Ruan, Y. Wang, Q. Wang, M. Li, Y. Yang, L. Xie, D. P. Liu, H. Zeng, S. Zhang, J. Xiao, L. Zhang, M. W. Nisar, and J. Dai, Empirical Study on Benchmarking Software Development Tasks, in the International Conference on Software Process, Minneapolis, USA, May 19-20, 2007, pp. 221-232.
73. S. Zhang, Y. Wang, F. Yuan, and L. Ruan, Mining software repositories to understand the performance of individual developers, in 31st Annual IEEE International Computer Software and Applications Conference (COMPSAC 2007), Beijing, China, 2007, pp. 625-626.
74. S. Zhang, Y. Wang, J. Tong, J. Zhou, and L. Ruan, Evaluation of Project Quality: A DEA-based Approach, in SPW/ProSim 2006, Shanghai, China, 2005, pp. 88-96.
75. J. Wang and M. Li, A Tridimensional Requirements Model and its Support for Stakeholder Coordination, Journal of Software, vol. 18, pp. 2380-2392, 2007.

76. F. Shu, Y. Zhao, J. Wang, and M. Li, User-Driven Requirements Elicitation Method with the Support of Personalized Domain Knowledge, Journal of Computer Research and Development, vol. 44, pp. 1044-1052, 2007.
77. M. Huang, F. Shu, and M. Li, A Risk-Driven Method for Prioritizing Requirements in Iteration Development, Journal of Software, vol. 17, pp. 2450-2460, 2006.
78. M. Li, M. Huang, F. Shu, and J. Li, A risk-driven method for eXtreme programming release planning, in ICSE 2006, Shanghai, China, 2006, pp. 423-430.
79. T. Huang, X.N. Ding, J. Wei. An Application-Semantics-Based Relaxed Transaction Model for Internetware, Science in China Series F C Information Sciences, 49(6):774-791, 2006.
80. T. Huang, N.J. Chen, J. Wei, W.B. Zhang, Y. Zhang. OnceAS/Q: A QoS-enabled Web application server. Journal of Software, 2004,15(12):1787-1799.
81. H. Chen, E.H. Wu. An Efficient Radiosity Solution for Bump Texture Generation. Computer Graphics, 24(4), (ACM SIGGRAPH'90), ACM Press, Aug. 1990:125-134
82. H. Chen, E.H. Wu. Radiosity for Furry Surfaces. EUROGRAPHICS'91, F.H. Post and W. Barth, eds. Elsevier Science Publishers B. V. (North-Holland), Sept. 1991, pp. 447-457.
83. E.H. Wu. A Radiosity Solution for Illumination of Random Fractal Surfaces. The Journal of Visualization and Computer Animation, 6(4), John Wiley & Sons, Oct. 1995, pp219-229.
84. Y.Y. Chen, H.Q. Sun, E.H. Wu. Modeling and Rendering Snowy Natural Scenery Using Multi-Mapping Techniques, The Journal of Visualization and Computer Animation, 14(1), John Wiley & Sons, Feb. 2003, pp21-30.
85. Y.Q. Xu, Y.Y. Chen, S. Lin, H. Zhong, E.H. Wu, B. Guo, H.Y. Shum. Photo-Realistic Rendering of Knitwear Using the Lumislice, Proc. ACM SIGGRAPH2001, pp391-398.
86. X.H. Liu, E.H. Wu, Hierarchical Structure with Focus Criterion for Rendering Height Field, J. of Computer Science &Technology, V13, Dec. 1998, pp.1-8.
87. W.C. Wang, E.H. Wu. Adaptable Splatting for Irregular Volume Rendering, Computer Graphics Forum, 17(6), Blackwell Publishers, 1999.
88. W.C. Wang, E.H. Wu, N. Max. A Selective Rendering Method for Data Visualization, Journal of Visualization and Computer Animation, John Wiley & Sons, 10(3), 1999, pp123-131.
89. W.C. Wang, D.H. Zhou, E.H. Wu. Accelerating Techniques in Volume Rendering of Irregular Data, Computers & Graphics, 21(3), Elsevier Science, 1997, p289-295.
90. E.H. Wu, Y.Q. Liu, X.H. Liu. An Improved Study of Real Time Fluid Simulation on GPU (invited paper). J. Computer Animation & Virtual World, 15(3-4), John Wiley & Sons, July 2004. pp139-146.
91. Y.Q. Liu, X.H. Liu, E.H. Wu. Real-Time 3D Fluid Simulation on GPU with Complex Obstacles, Proceedings of Pacific Graphics 2004, IEEE Computer Society, Oct. 2004. pp247-256.
92. Y.Q. Liu, X.H. Liu, E.H. Wu. Fluid Simulations on GPU with Complex Boundary Conditions, Poster Workshop on GPGPU, at ACM SIGGRAPG 2004, August 2004.
93. Y.Q. Liu, H.B. Zhu, X.H. Liu, E.H. Wu. Real Time Simulation of Physically Based On-Surface Flow. The Visual Computer, 21(8-10), Springer, September 2005. pp727-734.
94. H.B. Zhu, X.H. Liu, Y.Q. Liu, E.H. Wu. Simulation of Miscible Binary Mixtures Based on Lattice Boltzmann Method. Computer Animation & Virtual Worlds (CAVW).17(3-4), John Wiley. July 2006. pp403-411.
95. E.H. Wu, H.B. Zhu, X.H. Liu, Y.Q. Liu. Physically Based Fluid Dynamics & Interactions. Proc. of CyberWorlds'06 (paper on Keynote Speech), IEEE Computer Society. Lausanne, Switzerland. Nov. 28-30, 2006. pp3-13.
96. G.Z. Fei, K.Y. Cai, B.N. Guo, E.H. Wu. An Adaptive Sampling Scheme for Out-of-Core Simplification, Computer Graphics Forum, Blackwell Publishers, 21(2), 2002: pp111-119.
97. W.C. Wang, J. Li, E.H. Wu. 2D point-in-polygon test by classifying edges into layers. Computers & Graphics. 29(3), 2005. pp427-439.

98. W.C. Wang, J. Li, H.Q. Sun, E.H. Wu. A Layer-Based Representation of Polyhedrons for Point Containment Tests. IEEE Transactions on Visualization and Computer Graphics, (accepted). 2007
99. X. Ao, X.G. Wang, F. Tian, G.Z. Dai, H.A Wang, Cross-modal error correction of continuous handwriting recognition by Speech, Proceedings of ACM International Conference on Intelligent User Interfaces, 2007, Honolulu, Hawaii, USA, pp: 243-250
100. X. Ao, J.F. Li, X.G. Wang and G.Z. Dai, Structuralizing Digital Ink for Efficient Selection, Proceedings of ACM International Conference on Intelligent User Interfaces, 2006, Sydney, Australia, pp. 148-153
101. J.F. Li, X.W. Zhang, X. Ao, G.Z. Dai, Sketch recognition with continuous feedback based on incremental intention extraction, Proceedings of ACM International Conference on Intelligent User Interfaces, 2005, San Diego, California, USA, pp. 145-150.
102. F. Tian, X. Ao, H.A. Wang, V. Setlur, G.Z. Dai, The Tilt Cursor: Enhancing Stimulus-Response Compatibility by Providing 3D Orientation Cue of Pen, ACM Conference on Human Factors in Computing Systems (CHI'07), San Jose, California, USA, pp. 303-306.
103. W.X. Wang, H. Wang, G.Z. Dai, H.A. Wang, Visualization of large hierarchical data by circle packing, ACM Conference on Human Factors in Computing Systems (CHI'06), Canada, April 22-27, 2006 pp. 517-520.
104. M. Xia, Maximum Edge-Disjoint Paths Problem in Planar Graphs, Proc. 4th International Conference on Theory and Applications of Models of Computation (TAMC), 2007, LNCS 4484, pp. 566-572.
105. P. Zhang, An Approximation Algorithm to the *k*-Steiner Forest Problem, Proc. 4th International Conference on Theory and Applications of Models of Computation (TAMC), 2007, LNCS 4484, pp. 728-737.
106. J. Yan and J. Zhang, Backtracking Algorithms and Search Heuristics to Generate Test Suites for Combinatorial Testing, Proc. 30th Annual International Computer Software and Applications Conference (COMPSAC'06), pp.385-394.
107. Z. Xu and J. Zhang, A Test Data Generation Tool for Unit Testing of C Programs, Proc. of the 6th International Conference on Quality Software (QSIC), 2006, pp.107-116.
108. Y.D. Shen, Z. Zhang, Q. Yang, Objective-Oriented Utility-Based Association Mining, Proc. 2nd IEEE International Conference on Data Mining (ICDM), 2002, pp.426-433.