RWTH Aachen

Department of Computer Science
Technical Report

coJIVE: A System to Support Collaborative Jazz Improvisation

Jan Buchholz, Eric Lee, Jonathan Klein and Jan Borchers

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2007-04

RWTH Aachen · Department of Computer Science · March 2007

coJIVE: A System to Support Collaborative Jazz Improvisation

Jan Buchholz, Eric Lee, Jonathan Klein and Jan Borchers

Lehrstuhl für Informatik X
RWTH Aachen University, Germany
Email: {buchholz, eric, klein, borchers}@cs.rwth-aachen.de

Abstract. Jazz improvisation is a complex and demanding art of on-the-fly composition and performance. We present *coJIVE*, a system that allows musically inexperienced people to improvise to jazz by substituting for musical knowledge and experience. Using well-established musical theory, it aids users in creating a harmonic performance, and also helps coordinate a collaborative session amongst the participants. Remarks from users during evaluation showed an overall positive response to the support provided by the system.

1 Introduction

Jazz is a complex musical genre, with a wider theoretical background than, for example, classical music; it can deter even well-educated classical musicians. Besides performing composed songs and melodies, improvisation, the art of creating performances on the fly, is a very influential part of jazz. Undoubtedly, improvisation requires a lot of experience and technical abilities: a musician needs to have precise control of his instrument and build up a personal repertoire of short melodic patterns to create new improvised melodies. Analysis of chord progressions that define a song structure allows jazz musicians to determine a musical scale for each of the chords in the progression. With this scale, they know which notes fit the underlying chords' harmonic context (i.e., the notes belonging to that scale), and what notes will be likely to cause dissonance (i.e., the *outside* notes). Additionally, jazz musicians need to coordinate with each other during a session to ensure a balanced performance. Usually, only one musician, the soloist, is improvising at a given time, while the other participants provide an accompaniment without disturbing the improvisation.

How computers can help inexperienced users to master such complex tasks has been explored previously by systems such as *WorldBeat*[2]. This system, amongst other things, allows its users to improvise over a blues song with computer support. Evaluation of this system with users showed that this type of computer-supported interaction can bring musically inexperienced people closer to this type of sophisticated performance. We identified several other previous systems supporting improvisation; their achievements and shortcomings will be discussed in subsequent sections. These findings motivated us to develop our own system to support collaborative jazz improvisation: the **co**llaborative **J**azz **I**mpro**v**isation **E**nvironment (*coJIVE*).

2 Related Work

A number of previous research projects have tried to tackle the different aspects of musical performances. In general, such systems address either harmony or collaboration.

A musician's performance "sounds good" if the notes that she plays fit the harmonic context at that particular position in the song structure. Jazz musicians determine the contexts of a song by conducting the aforementioned analysis of the song's chord structure. This analysis has been the topic of a few projects, several of which were conducted by François Pachet.

[12] describes how he used the Lempel-Ziv data compression method on chord changes to determine the level of surprise in a chord progression (i.e., how likely a chord change is). The resulting Lempel-Ziv tree also allows for the extraction of a grammar of chord production rules without the input of musical knowledge. In [13], a system is presented that is able to perform a hierarchical harmony analysis on chord progressions. This system uses *shapes* derived from short sequences of chords; larger shapes can be derived from a sequence of smaller ones. The deployed method, however, does not always choose interesting scales; it was not meant for scale selection. A different approach is described in [5]: the presented system, created by Andrew Choi, uses minimisation to choose scales for a chord sequence. Therefore, a global scale distance measure is deployed, which describes the difference in pitches between two scales. This rather mathematical approach, of course, is not concerned with musical rules; it often chooses different scales than a real jazz musician would.

Most systems concerned with harmony in human performances follow the design principle for new instruments described in [10]: unnecessary degrees of freedom (i.e., the number of available notes) are reduced and the interface is specialised for the song to be played (i.e., context-aware reduction of notes).

Two approaches for improvised performances using scales to determine harmony (similar to what jazz musicians do) are presented in Nishimoto et al.'s *RhyMe*[11], and the aforementioned *WorldBeat*[2]. While *WorldBeat* uses predefined scales to map gestures performed with digital batons to notes, *RhyMe* calculates the scales to use based on a given song structure, and maps the notes of the current scale to the white keys on a keyboard. *ism* [6], Ishida et al.'s improvisation supporting system, exhibits a more dynamic approach: *ism* uses a database of melodic patterns and an N-gram model to calculate the probability of the last notes based on the notes played before. A probability threshold is used to decide on the note's appropriateness and can be used to adjust the system's support to fit the abilities of a user.

Another important aspect of musical performance is collaboration. A balanced interaction renders a more varied, and thus more interesting performance for the audience. Most research projects concerned with collaboration have turned their attention to connecting distributed users with networks — the users are not supported in organising their collaboration. Since *coJIVE* is not concerned with the aspect of remote collaboration, these systems are not discussed here. William F. Walker's *ImprovisationBuilder* [14] tries to recreate the behaviour of a human participant in jazz sessions. While this system can create improvisation on its own, it also listens to the other participants in the session and acts accordingly; based on the range of pitches used and the amount of notes played, it can determine which player is currently soloing.

Thus, our system is unique in the following ways:

- The system looks at several aspects of musical performance (e.g., harmony, collaborative behaviour, handling of the interfaces) rather than focusing on a single one.
- There is no assumption that users know about jazz theory or how to handle the instruments correctly, and various types of errors in user input are accounted for.
- The system not only corrects, but also enriches a user's input in accordance with the musical structure of the piece.
- The system is not only aware of collaboration amongst the performers, but actively mediates and supports it.

3 Requirements

Our goal was to create a system to enable people with little or no training in jazz improvisation to participate in a typical collaborative jazz improvisation session. To accomplish this, our system corrects and enriches users' musical input, as depicted in Fig. 1.

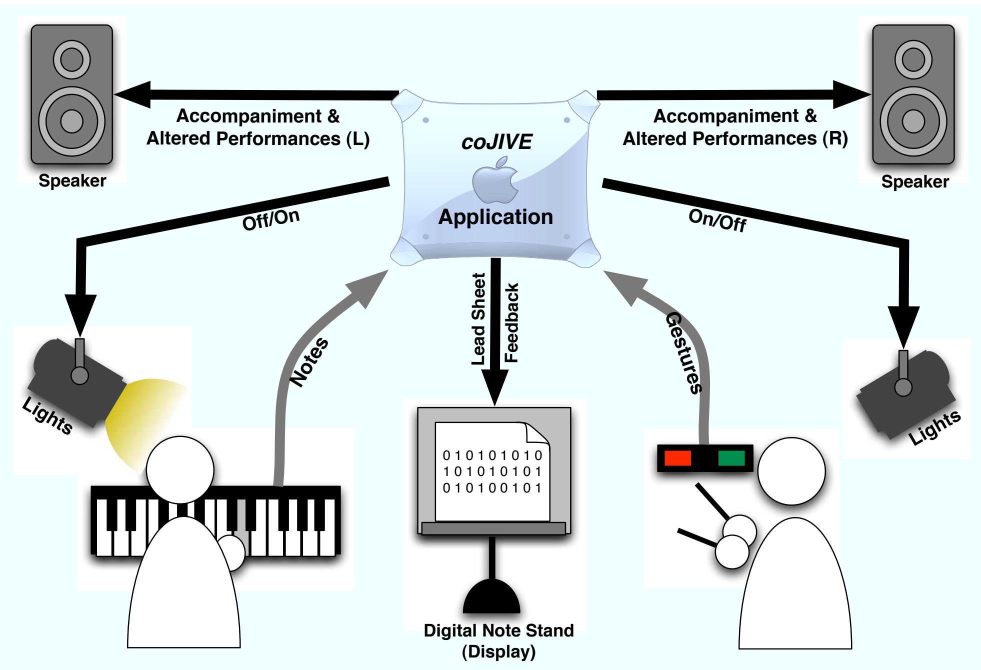

Fig. 1. Interaction between *coJIVE* and the users. The users create input with their interfaces, while the system plays an accompaniment and alters the given input. Information on the session and the input is provided for the users.

In a real jazz session, the musicians decide on a song to play, the order of the soloists, and the maximum length of a solo. The performance starts with the musicians playing the song's main melody, also called the *theme*. Thereafter, the musicians take turns soloing in the previously defined order, which can be dynamically changed using gestures. Another recitation of the main melody marks the end of the session. The left-hand side of Fig. 2 depicts the structure of such sessions.

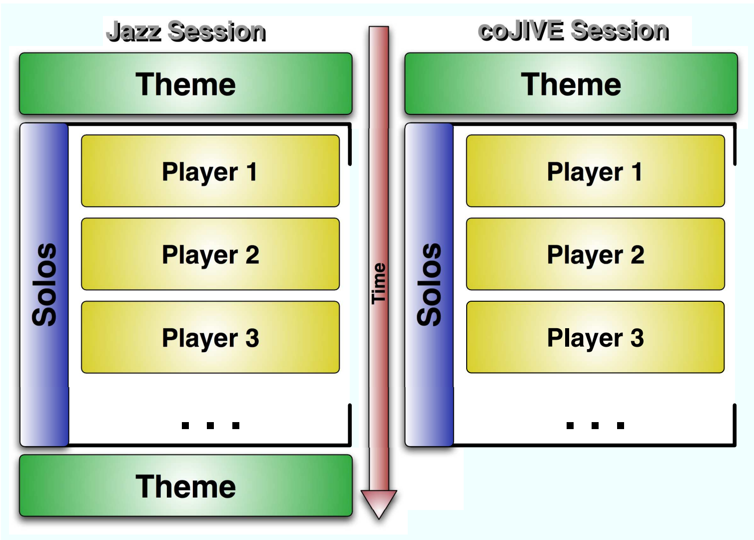

Fig. 2. The time flow of the different stages in a typical jazz session and a session created by *coJIVE*.

We envisioned a system that is able to recreate a simplified version of such sessions for its users. People can walk up to the system, choose an instrument, and agree on a song to perform to. Once the session is started, the system creates a drum rhythm and a bass-line to accompany the users. In addition, it recites the song's theme. The participants receive information on their current role in the session (i.e., whether it is their turn to solo or to accompany another soloist), and feedback on their performance. They are also informed about upcoming changes of their role. Furthermore, the system alters the participants' input to ensure a harmonic performance, and enriches it if necessary. The system does not repeat the current song's theme at the end of session to not leave the participants unoccupied. Fig. 2 shows the general session structure of this system on the right-hand side.

To facilitate a jazz session, the system has to provide:

- musical interfaces (instruments) and appropriate feedback;
- mechanisms to substitute for the users' lack of knowledge in jazz theory;
- musical support based on these mechanisms;
- a *lead sheet*[1], depicting the structure of the song to be performed as a sequence of chords;

[1] The lead sheet usually is a piece of paper showing important information about a song: name, composer, tempo, metre, as well as the song's chord structure and a notation of its main melody

- accompaniment defining the base tempo and harmony of the piece;
- mechanism(s) to faciliate collaboration.

4 Design

The final design of coJIVE was a result of a series of refined prototypes. The improvements and extensions made to these prototypes were based on opinions expressed by users who had tested earlier versions, and based on observations made during these tests.

4.1 Musical Interfaces

After preliminary design analyses and user interviews, we decided to pursue two musical interfaces with different characteristics and input methods for *coJIVE*, and explore different ways to facilitate performing with these interfaces:

Keyboard: The piano keyboard is often present in jazz sessions in the form of a piano or organ, and it allows the user accurate control over input timing and pitch. This interface, unsurprisingly, requires much practice to master. The precise one-to-one mapping of key to note also limits the range in which the input can be altered without aggravating the user.

Digital batons: These devices are played like a xylophone with hitting gestures, but without actual plates to hit. While users retain precise control over timing of their input, pitch control is less precise. For novice musicians, the batons have successfully been used in systems like *WorldBeat*[2].

4.2 Harmony Analysis

The analysis of a song's chord structure mentioned above is one of the key abilities necessary for improvisation. This procedure tries to identify *patterns*[7] of short sequences indicating a tonality. In accordance, one scale for each of the chords in the progression can be determined, providing a musician with a pool of notes to use. As this analysis is a complex ability to learn, novice users can not be expected to perform this analysis on their own. Therefore, *coJIVE* was provided with a mechanism to conduct this technique, often referred to as *Roman Numeral Analysis*, by itself.

This mechanism works in two phases. In the first phase, the chord sequence is scanned for known patterns. For each pattern known to the system, a rule is present and applied to each position. On finding a candidate sequence of chords for its pattern, a rule will label the chords with the Roman Numerals corresponding to their harmonic functions in the pattern. If a chord is labelled already, it depends on the behaviour of the pattern whether the rule overwrites the label; some patterns depend on other patterns being identified first. All rules are repeatedly applied to the whole chord sequence until no chord is relabelled anymore.

In the second phase, each chord is assigned a scale based on the roman numeral it is labelled with. For each roman numeral, only a small number of scales are available; the final decision is then based on the characteristics of the chord itself (e.g., root note, major, melodic minor). With a given scale, a distribution of note probabilities can be calculated.

As an additional aspect, the note played last by the user is used to further modify the probability distribution. We integrated this feature to allow advanced users to play chromatic sequences of notes, including notes from outside the current scale.

4.3 Musical Support

The system was designed to adjust its assistance not only based on the instrument, but also based on the user's specific abilities and needs. We created a user level scheme to estimate a user's skills based on four self-rated parameters: knowledge of musical theory, knowledge of jazz theory, experience with the piano keyboard, and experience in group performances. Support is adjusted based on how users rate themselves in each of these parameters.

Since the batons provide no physical targets to hit, the system needs to calculate *virtual targets* for the available notes. The "fuzzy" nature of the control over the batons allows for the system to adjust the size and amount of virtual targets based on the note probabilities derived from the analysis described in the previous section. Jazz musicians conduct a similar analysis during their own performances. Therefore, a target's width reflects its note's suitability based on the current harmonic context of the song — more harmonic notes are also more likely to be hit. Fig. 3 shows a mock-up visualisation of the virtual targets. A skill-based threshold is used to rule out less probable notes for novice users; notes with a probability below this threshold are not represented by a virtual target.

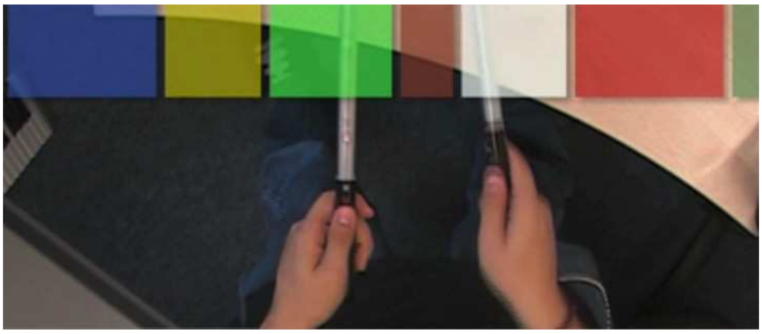

Fig. 3. The batons with a mock-up visualisation of the virtual targets calculated by the system. Each target represents a note and its width reflects that note's probability.

The more accurate control implied by the keyboard resulted in a more varied set of supporting mechanisms. Novices may not know what notes to play, but they can also not be expected to properly handle this sophisticated instrument. In addition, their performance is expected to be not as rich as the performances of seasoned jazz musicians. The rest of this subsection therefore describes the mechanisms deployed for the keyboard.

To preserve a general harmony in the keyboard performance, the user's input needs to be checked in terms of harmony. Therefore, each note is compared to its direct neighbours in terms of their probabilities; the most probable of these notes is then played. To enable adjustment of this mechanism to a user's skills, another threshold is used. The substitution of a note by a more probable neighbour only takes place if its probability is lower than the threshold.

Evaluation of an early prototype showed that novice users made several mistakes in handling the instrument: they often pressed two neighbouring keys with one finger by accident, or they pressed too many keys at once in a specific area (i.e., by pressing a flat hand on the keyboard). A timer was introduced to find out if such groups of notes belong together. The system checks for each pressed key whether one of its *physically* neighbouring keys — two white keys, for example, can physically be neighbours although they are separated by a black key — has been pressed within the last 100 milliseconds. If this is the case, the last key pressed is deemed to be played by mistake, and left out by the system. To prevent users from pressing too many keys at once in a specific area, a weighting function is used: for a key pressed by a user, the system accumulates the weights of all other keys currently held down in this key's neighbourhood. A key's weight corresponds to its distance to the key that initiated the scan. If the accumulated weights surpass another threshold, which again is based on the user's skills, the *density* in that neighbourhood is deemed too high and the initial key is discarded.

To enrich a user's input, we followed the idea that jazz musicians often use chord *voicings* (a specific arrangement of the chords' notes) to acquire richer performances. Commercially available keyboards sometimes allow playing selected chords at the push of a single key. We adapted this mechanism for *coJIVE* adding two additional modes for different user groups. While the one-key chord triggering is meant for novice users, classically educated musicians can trigger a chord voicing with three keys. The latter mode is based on the idea that chords in classical music usually have less notes (i.e., three) than jazz chords (i.e., four or five). On recognising a three-note chord played by a classical musician, the system will add an appropriate fourth note. The third mode, meant for intermediate users (i.e., musicians in training), looks for two appropriate keys pressed at the same time, adding two keys to form a jazz chord voicing. To allow the different user groups to benefit from these modes without making them mandatory, we split the keyboard into two sections below middle C ($C5$) and distributed the different modes over the two sections based on the user's skills (this distribution is depicted in Fig. 4). The voicings are acquired for each chord in the song structure individually, making the mechanism context-aware.

To allow users to interpret and perhaps even learn to anticipate the reaction of the system to their input, they are provided with feedback. The form of this feedback is closely tied to the instruments: the keyboard is recreated by the system and the system's output is shown on that keyboard along with the user's input. For the batons, a more fuzzy type of feedback was selected to reflect their inherent imprecision. The hitting gestures are represented as green circles on a black background with only one mark showing the middle C ($C5$) to provide a rough orientation. The size of a circle shows the gestures velocity.

4.4 Accompaniment

Although not the primary focus of this work, some type of accompaniment was nonetheless a necessary part of the system, to provide a rhythmic and harmonic foundation for the users' interaction with *coJIVE*.

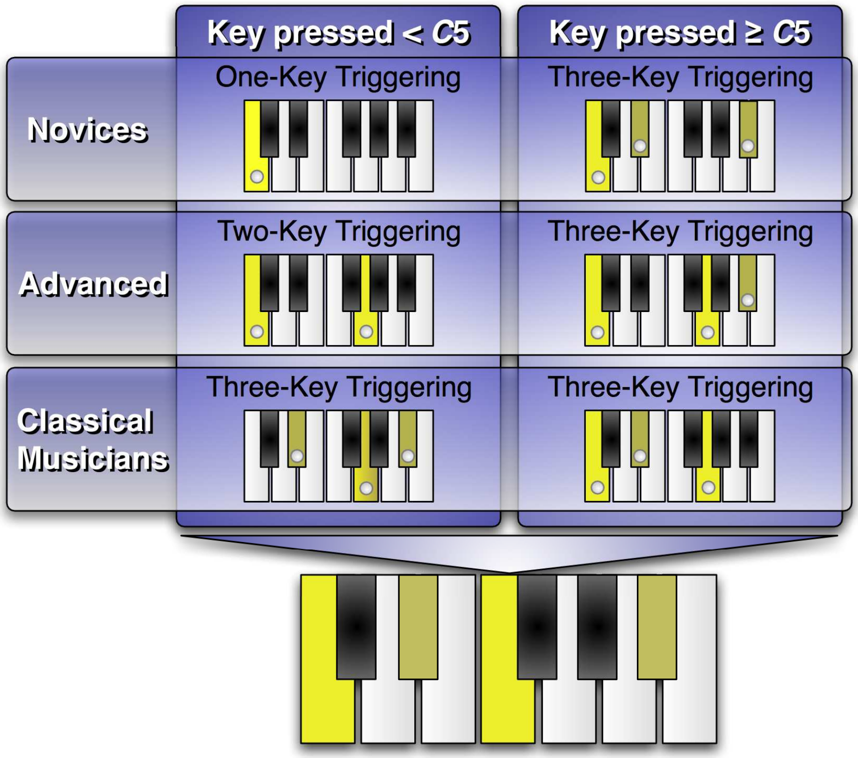

Fig. 4. Deployment scheme for the three different modes of chord triggering. The combinations of keys shown lead to the triggering of the same chord voicing.

We chose to automatically generate a simple swing pattern for the drum rhythm, repeated throughout the entire performance. The bass was provided as either an automatically generated walking bass up and down the current scale, or read from a pre-recorded file.

4.5 Collaboration Support

One of the primary goals of *coJIVE* is to assist users participating in a collaborative jazz improvisation session. In preliminary user interviews, we determined that many of our users had never participated in such a session; therefore, it was important for *coJIVE* to more explicitly expose and support the structure of a jazz improvisation session. For simplicity, we limited ourselves to *Solo & Comping*[3], where the musicians take turns soloing and the soloist is accompanied by the musicians.

In jazz sessions, the order of the soloists, and length of each solo, are typically determined beforehand and, possibly, decided on-the-fly during the session. To assist users, *coJIVE* instead takes over this leading role by dictating the order of soloists and the length of each solo.

Each solo has a length between 30 and 60 seconds, allowing the soloist ample time to create an improvised solo, but not boring the other players. The exact length of the solo was determined using a scheme inspired by Walker's *ImprovisationBuilder*[14], where a solo's temporal density and the dynamics (i.e., velocity) affect the length of the solo.

The system communicates to a particular user that he is soloing by shining a spotlight on them, and a change of roles is indicated using a countdown timer.

Finally, *coJIVE* analyzes the temporal density of an accompanying user's input. If it is found to be too interfering with the soloist, the dynamics of the accompanying player's input is modified such that it is more "difficult" for them to play loud notes, via a *velocity curve* (see Fig. reffig:VeloCurve).

5 Implementation

Our implementation of *coJIVE* consists of two parts: the *coJIVE* front-end[4] is responsible for reading and responding to user input, and the *coJIVE* back-end is a software framework encapsulating the musical knowledge necessary for analysing and augmenting the user input[8].

The system was developed in a user-centred, iterative process: several prototypes were designed, implemented, and subsequently evaluated in user studies. The results of an evaluation were then fed back into the design process to improve subsequent prototypes. With this process, the users' needs could be identified, and the system could be altered to satisfy them.

5.1 The Back-End

The back-end's main task is to perform the analysis of the chord progression and calculate the note probabilities. In addition, the back-end maintains a database of chord voicings for the aforementioned chord triggering assistance and accompaniment generation, and it allows dynamic loading of songs. The latter aspect was rendered possible by defining an XML format that describes the different

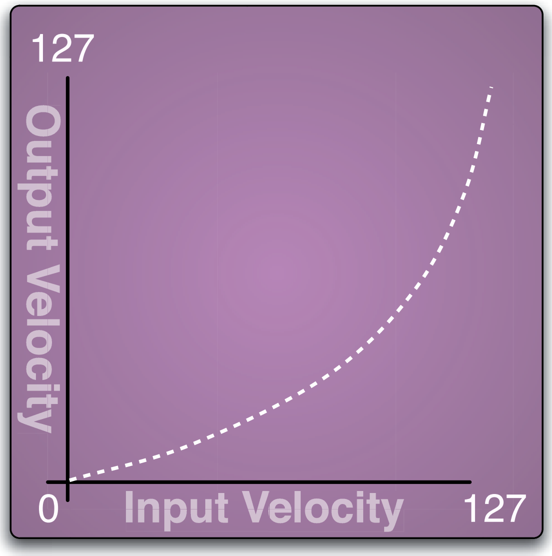

Fig. 5. A modified velocity curve. Velocity values of incoming notes are dampened, but higher values are not ruled out. With this mechanism, it gets harder to reach higher velocity values.

pieces of information of a song (name, tempo, metre, chord structure, etc.). The back-end was implemented in the form of a Mac OS X framework to allow for an easy integration with the front-end.

5.2 The Front-End

The front-end was implemented as a Cocoa[1] application in Mac OS X ; it provides the users with a graphical user interface, establishes connections to the instruments via MIDI, and implements the behaviour described in the design section. The structure of a song loaded into the system is displayed in the main window (*coJIVE*'s version of the lead sheet shown in Fig. 6), the current chord is highlighted by an orange backdrop, and a cursor marks the current position in the song for orientation.

For each player added to the system, a player field is created in a drawer beneath the window. Player-specific information and options are collected in this field: the player's role and the feedback mentioned above are depicted, and additional buttons are added to access the user's settings (MIDI ports and skill settings). The player field can also be detached from the main window (e.g., to make use of multiple screens). The LEDs, used for the aforementioned light signal mediating roles in collaboration, are controlled with a Teleo module[9]: the soloist's LED is lit, and the countdown to the next solo is accompanied by blinking of the affected users' LEDs.

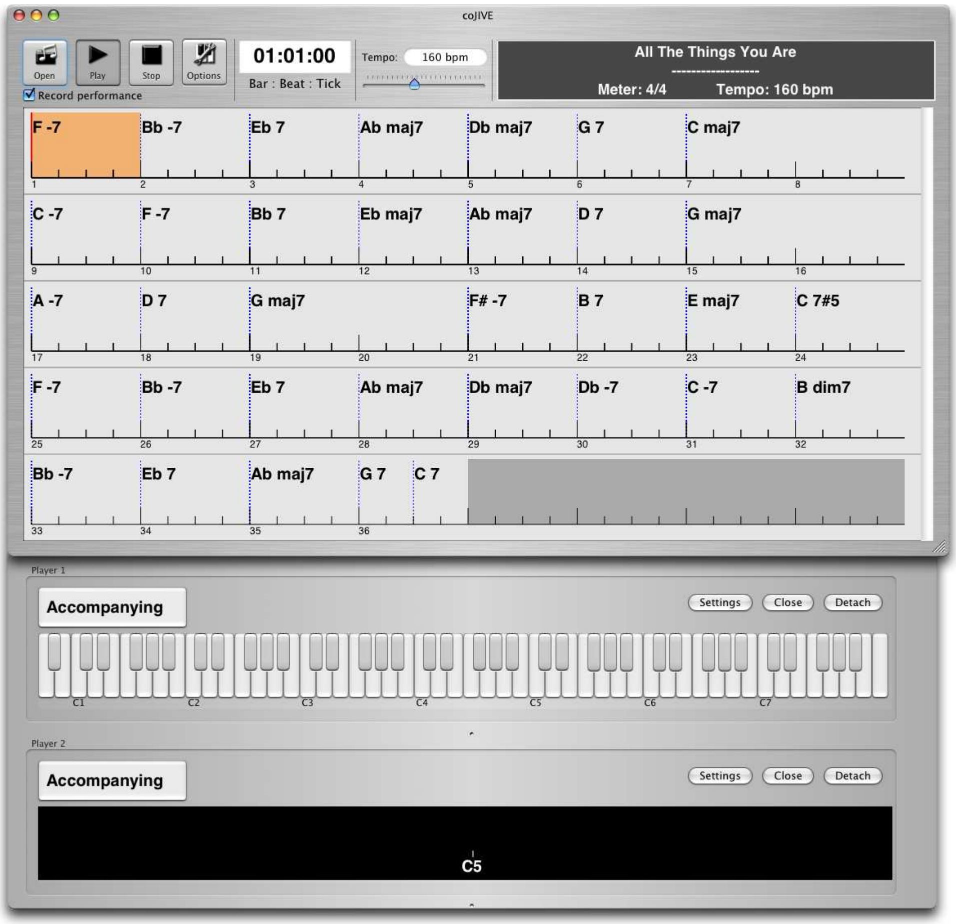

Fig. 6. The main application window of coJIVE with additional fields for a keyboard player and a baton player.

6 Evaluation

Besides the user tests during the development of *coJIVE*, we conducted a final study. The test subjects performed with the system in pairs of two; one subject played the keyboard, the other one used the batons to perform. Additionally they were asked to use two different versions of the system. They were not told that they were going to use the same system with full support in one pass, and without any support in the other pass. We collected data after the tests by means of questionnaires.

In this questionnaire, the subjects were asked to point out the differences between the two passes, and rate the two "versions" of the system concerning different aspects of musical support and collaboration support. They also had the opportunity to give comments, remarks, or simply write down their opinion.

7 Discussion

The qualitative part of the evaluation (i.e., comments, remarks, and opinions) offered some positive feedback: some of the subjects stated that the system's support offered an entertaining experience, a few even mentioned to have been more motivated by it during the tests.

Unfortunately, the ratings of the different aspects of the system's support did not show any clear results. We compared the ratings of the first and the second pass for all aspects, but we could not find any statistically significant difference.

Additionally, we observed differing reactions to the system during the tests. While some subjects started experimenting with their instrument almost instantly, exploring and exploiting the support, others were hesitant and only cautiously used their instrument.

8 Future Work

The results obtained from the last user study clearly showed that the support *coJIVE* currently provides is not sufficient for some users. One can imagine several improvements of the system to better support the users.

To allow for a more hands-on experience with the system, a new graphical design could be more appropriate. A layout that presents the task of improvising in the form and fashion of a game might help people to lose their reservation. The formerly "serious musical task" could then be perceived as entertainment rather than a chore.

Since the inexperienced subjects hardly created recognisable melodies, more guidance in that respect might be of help. Beside the scales, a database with melodic patterns could be used to determine the note probabilities. If the last sequence of notes resembled a pattern stored in the system, the subsequent note in that pattern would be rated more probable. Thus, a player would be directed towards a melody.

Beside the harmony of the notes played, their timing in respect to the rhythm is an important expressive parameter. A cautious real-time quantisation could be used to delay notes to rhythmically significant positions (e.g., on a beat or a swing note). We are currently conducting a separate study to examine the effect of such "timing corrections" on users.

Finally, it would be interesting to explore additional musical interfaces that often appear in jazz sessions: the guitar, the trumpet, the saxophone, etc. With the different characteristics of such instruments, new supporting mechanisms would be needed, but at the same time, new possibilities might emerge.

9 Conclusions

We presented *coJIVE*, a software system for computer-aided jazz improvisation. Our user-centred design of the system was aimed at identifying the users' needs in terms of musical performances and improvisation using the interfaces provided. Based on the results, we implemented and iteratively improved a set of mechanisms to substitute for the users' lack of knowledge and experience.

Our experiments have indicated that, although the performances novices can create with the system are far from what real jazz musicians are able to do, *coJIVE* can be effective in facilitating jazz improvisation, and that such research helps further interaction between humans and technology.

References

1. Apple Computer Inc. Cocoa. `http://developer.apple.com/cocoa/`, 2001.
2. Borchers, J. WorldBeat: Designing A Baton-Based Interface for an Interactive Music Exhibit. *Proceedings of the ACM CHI'97 International Conference on Human Factors in Computing Systems (Atlanta, Georgia).* ACM, New York (1997), 131–138.
3. Borchers, J. *A Pattern Approach to Interaction Design.* Wiley Series in Software Design Patterns. John Wiley & Sons Ltd, Chichestr, England, 2001.
4. Buchholz, J. A Software System for Computer Aided Jazz Improvisation. Master's thesis, RWTH Aachen University, Aachen, Germany, 2005.
5. Choi, A. Analysis of Jazz Chords as Optimization. `www.sixthhappiness.ca/blog`, 2004.
6. Ishida, K., Kitahara, T., and Takeda, M. ism: Improvisation Supporting System based on Melody Correction. *Proceedings of NIME '04 Conference on new Interfaces for Musical Expression.* IEEE, Washington (2004), 177–180.
7. Jungbluth, A. *Jazz-Harmonielehre, Funktionsharmonik und Modalität.* B. Schott's Söhne, Mainz, Germany, 1981.
8. Klein, J. A Pattern-based Software Framework for Computer Aided Jazz Improvisation. Master's thesis, RWTH Aachen University, Aachen, Germany, 2005.
9. Making Thinks LLC. Teleo. `http://www.makingthings.com/teleo.html`, 2002.
10. Nishimoto, K., Oshima, C., and Miyagawa, Y. Why always versatile? Dynamically Customizable Musical Instuments Facilitate Expressive Performances. *Proceedings of NIME '03 Conference on new Interfaces for Musical Expression.* McGill University, Montreal (2003), 164–169.
11. Nishimoto, K., Watanabe, H., Umata, I., Mase, K., and Nakatsu, R. A supporting Method for Creative Music Performance - Proposal of Musical Instrument with Fixed Mapping of Note-functions. *Trans. of Information Processing Society of Japan 39*, 5 (1998), 1556–1567.
12. Pachet, F. Surprising Harmonies. *International Journal on Computing Anticipatory Systems* (1999).
13. Pachet, F. (1997) Computer Analysis of Jazz Chord Sequences: Is Solar a Blues? *Readings in Music and Artificial Intelligence* (2000).
14. Walker, W. F. A Computer Participant in Musical Improvisation. *Proceedings of the ACM CHI'97 International Conference on Human Factors in Computing Systems (Atlanta, Georgia).* ACM, New York (1997), 123–130.

Aachener Informatik-Berichte

This is the list of all technical reports since 1987. To obtain copies of reports please consult
`http://aib.informatik.rwth-aachen.de/` **or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email:** `biblio@informatik.rwth-aachen.de`

1987-01	*	Fachgruppe Informatik: Jahresbericht 1986
1987-02	*	David de Frutos Escrig, Klaus Indermark: Equivalence Relations of Non-Deterministic Ianov-Schemes
1987-03	*	Manfred Nagl: A Software Development Environment based on Graph Technology
1987-04	*	Claus Lewerentz, Manfred Nagl, Bernhard Westfechtel: On Integration Mechanisms within a Graph-Based Software Development Environment
1987-05	*	Reinhard Rinn: Über Eingabeanomalien bei verschiedenen Inferenzmodellen
1987-06	*	Werner Damm, Gert Döhmen: Specifying Distributed Computer Architectures in AADL*
1987-07	*	Gregor Engels, Claus Lewerentz, Wilhelm Schäfer: Graph Grammar Engineering: A Software Specification Method
1987-08	*	Manfred Nagl: Set Theoretic Approaches to Graph Grammars
1987-09	*	Claus Lewerentz, Andreas Schürr: Experiences with a Database System for Software Documents
1987-10	*	Herbert Klaeren, Klaus Indermark: A New Implementation Technique for Recursive Function Definitions
1987-11	*	Rita Loogen: Design of a Parallel Programmable Graph Reduction Machine with Distributed Memory
1987-12		J. Börstler, U. Möncke, R. Wilhelm: Table compression for tree automata
1988-01	*	Gabriele Esser, Johannes Rückert, Frank Wagner Gesellschaftliche Aspekte der Informatik
1988-02	*	Peter Martini, Otto Spaniol: Token-Passing in High-Speed Backbone Networks for Campus-Wide Environments
1988-03	*	Thomas Welzel: Simulation of a Multiple Token Ring Backbone
1988-04	*	Peter Martini: Performance Comparison for HSLAN Media Access Protocols
1988-05	*	Peter Martini: Performance Analysis of Multiple Token Rings
1988-06	*	Andreas Mann, Johannes Rückert, Otto Spaniol: Datenfunknetze
1988-07	*	Andreas Mann, Johannes Rückert: Packet Radio Networks for Data Exchange
1988-08	*	Andreas Mann, Johannes Rückert: Concurrent Slot Assignment Protocol for Packet Radio Networks
1988-09	*	W. Kremer, F. Reichert, J. Rückert, A. Mann: Entwurf einer Netzwerktopologie für ein Mobilfunknetz zur Unterstützung des öffentlichen Straßenverkehrs
1988-10	*	Kai Jakobs: Towards User-Friendly Networking
1988-11	*	Kai Jakobs: The Directory - Evolution of a Standard
1988-12	*	Kai Jakobs: Directory Services in Distributed Systems - A Survey

1988-13 * Martine Schümmer: RS-511, a Protocol for the Plant Floor

1988-14 * U. Quernheim: Satellite Communication Protocols - A Performance Comparison Considering On-Board Processing

1988-15 * Peter Martini, Otto Spaniol, Thomas Welzel: File Transfer in High Speed Token Ring Networks: Performance Evaluation by Approximate Analysis and Simulation

1988-16 * Fachgruppe Informatik: Jahresbericht 1987

1988-17 * Wolfgang Thomas: Automata on Infinite Objects

1988-18 * Michael Sonnenschein: On Petri Nets and Data Flow Graphs

1988-19 * Heiko Vogler: Functional Distribution of the Contextual Analysis in Block-Structured Programming Languages: A Case Study of Tree Transducers

1988-20 * Thomas Welzel: Einsatz des Simulationswerkzeuges QNAP2 zur Leistungsbewertung von Kommunikationsprotokollen

1988-21 * Th. Janning, C. Lewerentz: Integrated Project Team Management in a Software Development Environment

1988-22 * Joost Engelfriet, Heiko Vogler: Modular Tree Transducers

1988-23 * Wolfgang Thomas: Automata and Quantifier Hierarchies

1988-24 * Uschi Heuter: Generalized Definite Tree Languages

1989-01 * Fachgruppe Informatik: Jahresbericht 1988

1989-02 * G. Esser, J. Rückert, F. Wagner (Hrsg.): Gesellschaftliche Aspekte der Informatik

1989-03 * Heiko Vogler: Bottom-Up Computation of Primitive Recursive Tree Functions

1989-04 * Andy Schürr: Introduction to PROGRESS, an Attribute Graph Grammar Based Specification Language

1989-05 J. Börstler: Reuse and Software Development - Problems, Solutions, and Bibliography (in German)

1989-06 * Kai Jakobs: OSI - An Appropriate Basis for Group Communication?

1989-07 * Kai Jakobs: ISO's Directory Proposal - Evolution, Current Status and Future Problems

1989-08 * Bernhard Westfechtel: Extension of a Graph Storage for Software Documents with Primitives for Undo/Redo and Revision Control

1989-09 * Peter Martini: High Speed Local Area Networks - A Tutorial

1989-10 * P. Davids, Th. Welzel: Performance Analysis of DQDB Based on Simulation

1989-11 * Manfred Nagl (Ed.): Abstracts of Talks presented at the WG '89 15th International Workshop on Graphtheoretic Concepts in Computer Science

1989-12 * Peter Martini: The DQDB Protocol - Is it Playing the Game?

1989-13 * Martine Schümmer: CNC/DNC Communication with MAP

1989-14 * Martine Schümmer: Local Area Networks for Manufactoring Environments with hard Real-Time Requirements

1989-15 * M. Schümmer, Th. Welzel, P. Martini: Integration of Field Bus and MAP Networks - Hierarchical Communication Systems in Production Environments

1989-16 * G. Vossen, K.-U. Witt: SUXESS: Towards a Sound Unification of Extensions of the Relational Data Model

1989-17 * J. Derissen, P. Hruschka, M.v.d. Beeck, Th. Janning, M. Nagl: Integrating Structured Analysis and Information Modelling
1989-18 A. Maassen: Programming with Higher Order Functions
1989-19 * Mario Rodriguez-Artalejo, Heiko Vogler: A Narrowing Machine for Syntax Directed BABEL
1989-20 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Graph-based Implementation of a Functional Logic Language
1990-01 * Fachgruppe Informatik: Jahresbericht 1989
1990-02 * Vera Jansen, Andreas Potthoff, Wolfgang Thomas, Udo Wermuth: A Short Guide to the AMORE System (Computing Automata, MOnoids and Regular Expressions)
1990-03 * Jerzy Skurczynski: On Three Hierarchies of Weak SkS Formulas
1990-04 R. Loogen: Stack-based Implementation of Narrowing
1990-05 H. Kuchen, A. Wagener: Comparison of Dynamic Load Balancing Strategies
1990-06 * Kai Jakobs, Frank Reichert: Directory Services for Mobile Communication
1990-07 * Kai Jakobs: What's Beyond the Interface - OSI Networks to Support Cooperative Work
1990-08 * Kai Jakobs: Directory Names and Schema - An Evaluation
1990-09 * Ulrich Quernheim, Dieter Kreuer: Das CCITT - Signalisierungssystem Nr. 7 auf Satellitenstrecken; Simulation der Zeichengabestrecke
1990-11 H. Kuchen, R. Loogen, J.J. Moreno Navarro, M. Rodriguez Artalejo: Lazy Narrowing in a Graph Machine
1990-12 * Kai Jakobs, Josef Kaltwasser, Frank Reichert, Otto Spaniol: Der Computer fährt mit
1990-13 * Rudolf Mathar, Andreas Mann: Analyzing a Distributed Slot Assignment Protocol by Markov Chains
1990-14 A. Maassen: Compilerentwicklung in Miranda - ein Praktikum in funktionaler Programmierung (written in german)
1990-15 * Manfred Nagl, Andreas Schürr: A Specification Environment for Graph Grammars
1990-16 A. Schürr: PROGRESS: A VHL-Language Based on Graph Grammars
1990-17 * Marita Möller: Ein Ebenenmodell wissensbasierter Konsultationen - Unterstützung für Wissensakquisition und Erklärungsfähigkeit
1990-18 * Eric Kowalewski: Entwurf und Interpretation einer Sprache zur Beschreibung von Konsultationsphasen in Expertensystemen
1990-20 Y. Ortega Mallen, D. de Frutos Escrig: A Complete Proof System for Timed Observations
1990-21 * Manfred Nagl: Modelling of Software Architectures: Importance, Notions, Experiences
1990-22 H. Fassbender, H. Vogler: A Call-by-need Implementation of Syntax Directed Functional Programming
1991-01 Guenther Geiler (ed.), Fachgruppe Informatik: Jahresbericht 1990
1991-03 B. Steffen, A. Ingolfsdottir: Characteristic Formulae for Processes with Divergence
1991-04 M. Portz: A new class of cryptosystems based on interconnection networks

1991-05 H. Kuchen, G. Geiler: Distributed Applicative Arrays
1991-06 * Ludwig Staiger: Kolmogorov Complexity and Hausdorff Dimension
1991-07 * Ludwig Staiger: Syntactic Congruences for w-languages
1991-09 * Eila Kuikka: A Proposal for a Syntax-Directed Text Processing System
1991-10 K. Gladitz, H. Fassbender, H. Vogler: Compiler-based Implementation of Syntax-Directed Functional Programming
1991-11 R. Loogen, St. Winkler: Dynamic Detection of Determinism in Functional Logic Languages
1991-12 * K. Indermark, M. Rodriguez Artalejo (Eds.): Granada Workshop on the Integration of Functional and Logic Programming
1991-13 * Rolf Hager, Wolfgang Kremer: The Adaptive Priority Scheduler: A More Fair Priority Service Discipline
1991-14 * Andreas Fasbender, Wolfgang Kremer: A New Approximation Algorithm for Tandem Networks with Priority Nodes
1991-15 J. Börstler, A. Zündorf: Revisiting extensions to Modula-2 to support reusability
1991-16 J. Börstler, Th. Janning: Bridging the gap between Requirements Analysis and Design
1991-17 A. Zündorf, A. Schürr: Nondeterministic Control Structures for Graph Rewriting Systems
1991-18 * Matthias Jarke, John Mylopoulos, Joachim W. Schmidt, Yannis Vassiliou: DAIDA: An Environment for Evolving Information Systems
1991-19 M. Jeusfeld, M. Jarke: From Relational to Object-Oriented Integrity Simplification
1991-20 G. Hogen, A. Kindler, R. Loogen: Automatic Parallelization of Lazy Functional Programs
1991-21 * Prof. Dr. rer. nat. Otto Spaniol: ODP (Open Distributed Processing): Yet another Viewpoint
1991-22 H. Kuchen, F. Lücking, H. Stoltze: The Topology Description Language TDL
1991-23 S. Graf, B. Steffen: Compositional Minimization of Finite State Systems
1991-24 R. Cleaveland, J. Parrow, B. Steffen: The Concurrency Workbench: A Semantics Based Tool for the Verification of Concurrent Systems
1991-25 * Rudolf Mathar, Jürgen Mattfeldt: Optimal Transmission Ranges for Mobile Communication in Linear Multihop Packet Radio Networks
1991-26 M. Jeusfeld, M. Staudt: Query Optimization in Deductive Object Bases
1991-27 J. Knoop, B. Steffen: The Interprocedural Coincidence Theorem
1991-28 J. Knoop, B. Steffen: Unifying Strength Reduction and Semantic Code Motion
1991-30 T. Margaria: First-Order theories for the verification of complex FSMs
1991-31 B. Steffen: Generating Data Flow Analysis Algorithms from Modal Specifications
1992-01 Stefan Eherer (ed.), Fachgruppe Informatik: Jahresbericht 1991
1992-02 * Bernhard Westfechtel: Basismechanismen zur Datenverwaltung in strukturbezogenen Hypertextsystemen
1992-04 S. A. Smolka, B. Steffen: Priority as Extremal Probability
1992-05 * Matthias Jarke, Carlos Maltzahn, Thomas Rose: Sharing Processes: Team Coordination in Design Repositories

1992-06 O. Burkart, B. Steffen: Model Checking for Context-Free Processes
1992-07 * Matthias Jarke, Klaus Pohl: Information Systems Quality and Quality Information Systems
1992-08 * Rudolf Mathar, Jürgen Mattfeldt: Analyzing Routing Strategy NFP in Multihop Packet Radio Networks on a Line
1992-09 * Alfons Kemper, Guido Moerkotte: Grundlagen objektorientierter Datenbanksysteme
1992-10 Matthias Jarke, Manfred Jeusfeld, Andreas Miethsam, Michael Gocek: Towards a logic-based reconstruction of software configuration management
1992-11 Werner Hans: A Complete Indexing Scheme for WAM-based Abstract Machines
1992-12 W. Hans, R. Loogen, St. Winkler: On the Interaction of Lazy Evaluation and Backtracking
1992-13 * Matthias Jarke, Thomas Rose: Specification Management with CAD
1992-14 Th. Noll, H. Vogler: Top-down Parsing with Simultaneous Evaluation on Noncircular Attribute Grammars
1992-15 A. Schuerr, B. Westfechtel: Graphgrammatiken und Graphersetzungssysteme(written in german)
1992-16 * Graduiertenkolleg Informatik und Technik (Hrsg.): Forschungsprojekte des Graduiertenkollegs Informatik und Technik
1992-17 M. Jarke (ed.): ConceptBase V3.1 User Manual
1992-18 * Clarence A. Ellis, Matthias Jarke (Eds.): Distributed Cooperation in Integrated Information Systems - Proceedings of the Third International Workshop on Intelligent and Cooperative Information Systems
1992-19-00 H. Kuchen, R. Loogen (eds.): Proceedings of the 4th Int. Workshop on the Parallel Implementation of Functional Languages
1992-19-01 G. Hogen, R. Loogen: PASTEL - A Parallel Stack-Based Implementation of Eager Functional Programs with Lazy Data Structures (Extended Abstract)
1992-19-02 H. Kuchen, K. Gladitz: Implementing Bags on a Shared Memory MIMD-Machine
1992-19-03 C. Rathsack, S.B. Scholz: LISA - A Lazy Interpreter for a Full-Fledged Lambda-Calculus
1992-19-04 T.A. Bratvold: Determining Useful Parallelism in Higher Order Functions
1992-19-05 S. Kahrs: Polymorphic Type Checking by Interpretation of Code
1992-19-06 M. Chakravarty, M. Köhler: Equational Constraints, Residuation, and the Parallel JUMP-Machine
1992-19-07 J. Seward: Polymorphic Strictness Analysis using Frontiers (Draft Version)
1992-19-08 D. Gärtner, A. Kimms, W. Kluge: pi-Red^+ - A Compiling Graph-Reduction System for a Full Fledged Lambda-Calculus
1992-19-09 D. Howe, G. Burn: Experiments with strict STG code
1992-19-10 J. Glauert: Parallel Implementation of Functional Languages Using Small Processes
1992-19-11 M. Joy, T. Axford: A Parallel Graph Reduction Machine
1992-19-12 A. Bennett, P. Kelly: Simulation of Multicache Parallel Reduction

1992-19-13 K. Langendoen, D.J. Agterkamp: Cache Behaviour of Lazy Functional Programs (Working Paper)

1992-19-14 K. Hammond, S. Peyton Jones: Profiling scheduling strategies on the GRIP parallel reducer

1992-19-15 S. Mintchev: Using Strictness Information in the STG-machine

1992-19-16 D. Rushall: An Attribute Grammar Evaluator in Haskell

1992-19-17 J. Wild, H. Glaser, P. Hartel: Statistics on storage management in a lazy functional language implementation

1992-19-18 W.S. Martins: Parallel Implementations of Functional Languages

1992-19-19 D. Lester: Distributed Garbage Collection of Cyclic Structures (Draft version)

1992-19-20 J.C. Glas, R.F.H. Hofman, W.G. Vree: Parallelization of Branch-and-Bound Algorithms in a Functional Programming Environment

1992-19-21 S. Hwang, D. Rushall: The nu-STG machine: a parallelized Spineless Tagless Graph Reduction Machine in a distributed memory architecture (Draft version)

1992-19-22 G. Burn, D. Le Metayer: Cps-Translation and the Correctness of Optimising Compilers

1992-19-23 S.L. Peyton Jones, P. Wadler: Imperative functional programming (Brief summary)

1992-19-24 W. Damm, F. Liu, Th. Peikenkamp: Evaluation and Parallelization of Functions in Functional + Logic Languages (abstract)

1992-19-25 M. Kesseler: Communication Issues Regarding Parallel Functional Graph Rewriting

1992-19-26 Th. Peikenkamp: Charakterizing and representing neededness in functional loginc languages (abstract)

1992-19-27 H. Doerr: Monitoring with Graph-Grammars as formal operational Models

1992-19-28 J. van Groningen: Some implementation aspects of Concurrent Clean on distributed memory architectures

1992-19-29 G. Ostheimer: Load Bounding for Implicit Parallelism (abstract)

1992-20 H. Kuchen, F.J. Lopez Fraguas, J.J. Moreno Navarro, M. Rodriguez Artalejo: Implementing Disequality in a Lazy Functional Logic Language

1992-21 H. Kuchen, F.J. Lopez Fraguas: Result Directed Computing in a Functional Logic Language

1992-22 H. Kuchen, J.J. Moreno Navarro, M.V. Hermenegildo: Independent AND-Parallel Narrowing

1992-23 T. Margaria, B. Steffen: Distinguishing Formulas for Free

1992-24 K. Pohl: The Three Dimensions of Requirements Engineering

1992-25 * R. Stainov: A Dynamic Configuration Facility for Multimedia Communications

1992-26 * Michael von der Beeck: Integration of Structured Analysis and Timed Statecharts for Real-Time and Concurrency Specification

1992-27 W. Hans, St. Winkler: Aliasing and Groundness Analysis of Logic Programs through Abstract Interpretation and its Safety

1992-28 * Gerhard Steinke, Matthias Jarke: Support for Security Modeling in Information Systems Design

1992-29 B. Schinzel: Warum Frauenforschung in Naturwissenschaft und Technik

1992-30 A. Kemper, G. Moerkotte, K. Peithner: Object-Orientation Axiomatised by Dynamic Logic

1992-32 * Bernd Heinrichs, Kai Jakobs: Timer Handling in High-Performance Transport Systems

1992-33 * B. Heinrichs, K. Jakobs, K. Lenßen, W. Reinhardt, A. Spinner: Euro-Bridge: Communication Services for Multimedia Applications

1992-34 C. Gerlhof, A. Kemper, Ch. Kilger, G. Moerkotte: Partition-Based Clustering in Object Bases: From Theory to Practice

1992-35 J. Börstler: Feature-Oriented Classification and Reuse in IPSEN

1992-36 M. Jarke, J. Bubenko, C. Rolland, A. Sutcliffe, Y. Vassiliou: Theories Underlying Requirements Engineering: An Overview of NATURE at Genesis

1992-37 * K. Pohl, M. Jarke: Quality Information Systems: Repository Support for Evolving Process Models

1992-38 A. Zuendorf: Implementation of the imperative / rule based language PROGRES

1992-39 P. Koch: Intelligentes Backtracking bei der Auswertung funktional-logischer Programme

1992-40 * Rudolf Mathar, Jürgen Mattfeldt: Channel Assignment in Cellular Radio Networks

1992-41 * Gerhard Friedrich, Wolfgang Neidl: Constructive Utility in Model-Based Diagnosis Repair Systems

1992-42 * P. S. Chen, R. Hennicker, M. Jarke: On the Retrieval of Reusable Software Components

1992-43 W. Hans, St.Winkler: Abstract Interpretation of Functional Logic Languages

1992-44 N. Kiesel, A. Schuerr, B. Westfechtel: Design and Evaluation of GRAS, a Graph-Oriented Database System for Engineering Applications

1993-01 * Fachgruppe Informatik: Jahresbericht 1992

1993-02 * Patrick Shicheng Chen: On Inference Rules of Logic-Based Information Retrieval Systems

1993-03 G. Hogen, R. Loogen: A New Stack Technique for the Management of Runtime Structures in Distributed Environments

1993-05 A. Zündorf: A Heuristic for the Subgraph Isomorphism Problem in Executing PROGRES

1993-06 A. Kemper, D. Kossmann: Adaptable Pointer Swizzling Strategies in Object Bases: Design, Realization, and Quantitative Analysis

1993-07 * Graduiertenkolleg Informatik und Technik (Hrsg.): Graduiertenkolleg Informatik und Technik

1993-08 * Matthias Berger: k-Coloring Vertices using a Neural Network with Convergence to Valid Solutions

1993-09 M. Buchheit, M. Jeusfeld, W. Nutt, M. Staudt: Subsumption between Queries to Object-Oriented Databases

1993-10 O. Burkart, B. Steffen: Pushdown Processes: Parallel Composition and Model Checking

1993-11 * R. Große-Wienker, O. Hermanns, D. Menzenbach, A. Pollacks, S. Repetzki, J. Schwartz, K. Sonnenschein, B. Westfechtel: Das SUKITS-Projekt: A-posteriori-Integration heterogener CIM-Anwendungssysteme

1993-12 * Rudolf Mathar, Jürgen Mattfeldt: On the Distribution of Cumulated Interference Power in Rayleigh Fading Channels

1993-13 O. Maler, L. Staiger: On Syntactic Congruences for omega-languages

1993-14 M. Jarke, St. Eherer, R. Gallersdoerfer, M. Jeusfeld, M. Staudt: ConceptBase - A Deductive Object Base Manager

1993-15 M. Staudt, H.W. Nissen, M.A. Jeusfeld: Query by Class, Rule and Concept

1993-16 * M. Jarke, K. Pohl, St. Jacobs et al.: Requirements Engineering: An Integrated View of Representation Process and Domain

1993-17 * M. Jarke, K. Pohl: Establishing Vision in Context: Towards a Model of Requirements Processes

1993-18 W. Hans, H. Kuchen, St. Winkler: Full Indexing for Lazy Narrowing

1993-19 W. Hans, J.J. Ruz, F. Saenz, St. Winkler: A VHDL Specification of a Shared Memory Parallel Machine for Babel

1993-20 * K. Finke, M. Jarke, P. Szczurko, R. Soltysiak: Quality Management for Expert Systems in Process Control

1993-21 M. Jarke, M.A. Jeusfeld, P. Szczurko: Three Aspects of Intelligent Cooperation in the Quality Cycle

1994-01 Margit Generet, Sven Martin (eds.), Fachgruppe Informatik: Jahresbericht 1993

1994-02 M. Lefering: Development of Incremental Integration Tools Using Formal Specifications

1994-03 * P. Constantopoulos, M. Jarke, J. Mylopoulos, Y. Vassiliou: The Software Information Base: A Server for Reuse

1994-04 * Rolf Hager, Rudolf Mathar, Jürgen Mattfeldt: Intelligent Cruise Control and Reliable Communication of Mobile Stations

1994-05 * Rolf Hager, Peter Hermesmann, Michael Portz: Feasibility of Authentication Procedures within Advanced Transport Telematics

1994-06 * Claudia Popien, Bernd Meyer, Axel Kuepper: A Formal Approach to Service Import in ODP Trader Federations

1994-07 P. Peters, P. Szczurko: Integrating Models of Quality Management Methods by an Object-Oriented Repository

1994-08 * Manfred Nagl, Bernhard Westfechtel: A Universal Component for the Administration in Distributed and Integrated Development Environments

1994-09 * Patrick Horster, Holger Petersen: Signatur- und Authentifikationsverfahren auf der Basis des diskreten Logarithmusproblems

1994-11 A. Schürr: PROGRES, A Visual Language and Environment for PROgramming with Graph REwrite Systems

1994-12 A. Schürr: Specification of Graph Translators with Triple Graph Grammars

1994-13 A. Schürr: Logic Based Programmed Structure Rewriting Systems

1994-14 L. Staiger: Codes, Simplifying Words, and Open Set Condition

1994-15 * Bernhard Westfechtel: A Graph-Based System for Managing Configurations of Engineering Design Documents

1994-16 P. Klein: Designing Software with Modula-3

1994-17 I. Litovsky, L. Staiger: Finite acceptance of infinite words

1994-18 G. Hogen, R. Loogen: Parallel Functional Implementations: Graphbased vs. Stackbased Reduction

1994-19 M. Jeusfeld, U. Johnen: An Executable Meta Model for Re-Engineering of Database Schemas

1994-20 * R. Gallersdörfer, M. Jarke, K. Klabunde: Intelligent Networks as a Data Intensive Application (INDIA)

1994-21 M. Mohnen: Proving the Correctness of the Static Link Technique Using Evolving Algebras

1994-22 H. Fernau, L. Staiger: Valuations and Unambiguity of Languages, with Applications to Fractal Geometry

1994-24 * M. Jarke, K. Pohl, R. Dömges, St. Jacobs, H. W. Nissen: Requirements Information Management: The NATURE Approach

1994-25 * M. Jarke, K. Pohl, C. Rolland, J.-R. Schmitt: Experience-Based Method Evaluation and Improvement: A Process Modeling Approach

1994-26 * St. Jacobs, St. Kethers: Improving Communication and Decision Making within Quality Function Deployment

1994-27 * M. Jarke, H. W. Nissen, K. Pohl: Tool Integration in Evolving Information Systems Environments

1994-28 O. Burkart, D. Caucal, B. Steffen: An Elementary Bisimulation Decision Procedure for Arbitrary Context-Free Processes

1995-01 * Fachgruppe Informatik: Jahresbericht 1994

1995-02 Andy Schürr, Andreas J. Winter, Albert Zündorf: Graph Grammar Engineering with PROGRES

1995-03 Ludwig Staiger: A Tight Upper Bound on Kolmogorov Complexity by Hausdorff Dimension and Uniformly Optimal Prediction

1995-04 Birgitta König-Ries, Sven Helmer, Guido Moerkotte: An experimental study on the complexity of left-deep join ordering problems for cyclic queries

1995-05 Sophie Cluet, Guido Moerkotte: Efficient Evaluation of Aggregates on Bulk Types

1995-06 Sophie Cluet, Guido Moerkotte: Nested Queries in Object Bases

1995-07 Sophie Cluet, Guido Moerkotte: Query Optimization Techniques Exploiting Class Hierarchies

1995-08 Markus Mohnen: Efficient Compile-Time Garbage Collection for Arbitrary Data Structures

1995-09 Markus Mohnen: Functional Specification of Imperative Programs: An Alternative Point of View of Functional Languages

1995-10 Rainer Gallersdörfer, Matthias Nicola: Improving Performance in Replicated Databases through Relaxed Coherency

1995-11 * M.Staudt, K.von Thadden: Subsumption Checking in Knowledge Bases

1995-12 * G.V.Zemanek, H.W.Nissen, H.Hubert, M.Jarke: Requirements Analysis from Multiple Perspectives: Experiences with Conceptual Modeling Technology

1995-13 * M.Staudt, M.Jarke: Incremental Maintenance of Externally Materialized Views

1995-14 * P.Peters, P.Szczurko, M.Jeusfeld: Oriented Information Management: Conceptual Models at Work

1995-15 * Matthias Jarke, Sudha Ram (Hrsg.): WITS 95 Proceedings of the 5th Annual Workshop on Information Technologies and Systems

1995-16 * W.Hans, St.Winkler, F.Saenz: Distributed Execution in Functional Logic Programming

1996-01 * Jahresbericht 1995

1996-02 Michael Hanus, Christian Prehofer: Higher-Order Narrowing with Definitional Trees

1996-03 * W.Scheufele, G.Moerkotte: Optimal Ordering of Selections and Joins in Acyclic Queries with Expensive Predicates

1996-04 Klaus Pohl: PRO-ART: Enabling Requirements Pre-Traceability

1996-05 Klaus Pohl: Requirements Engineering: An Overview

1996-06 * M.Jarke, W.Marquardt: Design and Evaluation of Computer–Aided Process Modelling Tools

1996-07 Olaf Chitil: The Sigma-Semantics: A Comprehensive Semantics for Functional Programs

1996-08 * S.Sripada: On Entropy and the Limitations of the Second Law of Thermodynamics

1996-09 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP96 - Fifth International Conference on Algebraic and Logic Programming

1996-09-0 Michael Hanus (Ed.): Proceedings of the Poster Session of ALP 96 - Fifth International Conference on Algebraic and Logic Programming: Introduction and table of contents

1996-09-1 Ilies Alouini: An Implementation of Conditional Concurrent Rewriting on Distributed Memory Machines

1996-09-2 Olivier Danvy, Karoline Malmkjær: On the Idempotence of the CPS Transformation

1996-09-3 Víctor M. Gulías, José L. Freire: Concurrent Programming in Haskell

1996-09-4 Sébastien Limet, Pierre Réty: On Decidability of Unifiability Modulo Rewrite Systems

1996-09-5 Alexandre Tessier: Declarative Debugging in Constraint Logic Programming

1996-10 Reidar Conradi, Bernhard Westfechtel: Version Models for Software Configuration Management

1996-11 * C.Weise, D.Lenzkes: A Fast Decision Algorithm for Timed Refinement

1996-12 * R.Dömges, K.Pohl, M.Jarke, B.Lohmann, W.Marquardt: PRO-ART/CE* — An Environment for Managing the Evolution of Chemical Process Simulation Models

1996-13 * K.Pohl, R.Klamma, K.Weidenhaupt, R.Dömges, P.Haumer, M.Jarke: A Framework for Process-Integrated Tools

1996-14 * R.Gallersdörfer, K.Klabunde, A.Stolz, M.Eßmajor: INDIA — Intelligent Networks as a Data Intensive Application, Final Project Report, June 1996

1996-15 * H.Schimpe, M.Staudt: VAREX: An Environment for Validating and Refining Rule Bases

1996-16 * M.Jarke, M.Gebhardt, S.Jacobs, H.Nissen: Conflict Analysis Across Heterogeneous Viewpoints: Formalization and Visualization

1996-17 Manfred A. Jeusfeld, Tung X. Bui: Decision Support Components on the Internet

1996-18 Manfred A. Jeusfeld, Mike Papazoglou: Information Brokering: Design, Search and Transformation

1996-19 * P.Peters, M.Jarke: Simulating the impact of information flows in networked organizations

1996-20 Matthias Jarke, Peter Peters, Manfred A. Jeusfeld: Model-driven planning and design of cooperative information systems

1996-21 * G.de Michelis, E.Dubois, M.Jarke, F.Matthes, J.Mylopoulos, K.Pohl, J.Schmidt, C.Woo, E.Yu: Cooperative information systems: a manifesto

1996-22 * S.Jacobs, M.Gebhardt, S.Kethers, W.Rzasa: Filling HTML forms simultaneously: CoWeb architecture and functionality

1996-23 * M.Gebhardt, S.Jacobs: Conflict Management in Design

1997-01 Michael Hanus, Frank Zartmann (eds.): Jahresbericht 1996

1997-02 Johannes Faassen: Using full parallel Boltzmann Machines for Optimization

1997-03 Andreas Winter, Andy Schürr: Modules and Updatable Graph Views for PROgrammed Graph REwriting Systems

1997-04 Markus Mohnen, Stefan Tobies: Implementing Context Patterns in the Glasgow Haskell Compiler

1997-05 * S.Gruner: Schemakorrespondenzaxiome unterstützen die paargrammatische Spezifikation inkrementeller Integrationswerkzeuge

1997-06 Matthias Nicola, Matthias Jarke: Design and Evaluation of Wireless Health Care Information Systems in Developing Countries

1997-07 Petra Hofstedt: Taskparallele Skelette für irregulär strukturierte Probleme in deklarativen Sprachen

1997-08 Dorothea Blostein, Andy Schürr: Computing with Graphs and Graph Rewriting

1997-09 Carl-Arndt Krapp, Bernhard Westfechtel: Feedback Handling in Dynamic Task Nets

1997-10 Matthias Nicola, Matthias Jarke: Integrating Replication and Communication in Performance Models of Distributed Databases

1997-11 * R. Klamma, P. Peters, M. Jarke: Workflow Support for Failure Management in Federated Organizations

1997-13 Markus Mohnen: Optimising the Memory Management of Higher-Order Functional Programs

1997-14 Roland Baumann: Client/Server Distribution in a Structure-Oriented Database Management System

1997-15 George Botorog: High-Level Parallel Programming and the Efficient Implementation of Numerical Algorithms

1998-01 * Fachgruppe Informatik: Jahresbericht 1997

1998-02 Stefan Gruner, Manfred Nagel, Andy Schürr: Fine-grained and Structure-Oriented Document Integration Tools are Needed for Development Processes

1998-03 Stefan Gruner: Einige Anmerkungen zur graphgrammatischen Spezifikation von Integrationswerkzeugen nach Westfechtel, Janning, Lefering und Schürr

1998-04 * O. Kubitz: Mobile Robots in Dynamic Environments

1998-05 Martin Leucker, Stephan Tobies: Truth - A Verification Platform for Distributed Systems

1998-06 * Matthias Oliver Berger: DECT in the Factory of the Future

1998-07 M. Arnold, M. Erdmann, M. Glinz, P. Haumer, R. Knoll, B. Paech, K. Pohl, J. Ryser, R. Studer, K. Weidenhaupt: Survey on the Scenario Use in Twelve Selected Industrial Projects

1998-09 * Th. Lehmann: Geometrische Ausrichtung medizinischer Bilder am Beispiel intraoraler Radiographien

1998-10 * M. Nicola, M. Jarke: Performance Modeling of Distributed and Replicated Databases

1998-11 * Ansgar Schleicher, Bernhard Westfechtel, Dirk Jäger: Modeling Dynamic Software Processes in UML

1998-12 * W. Appelt, M. Jarke: Interoperable Tools for Cooperation Support using the World Wide Web

1998-13 Klaus Indermark: Semantik rekursiver Funktionsdefinitionen mit Striktheitsinformation

1999-01 * Jahresbericht 1998

1999-02 * F. Huch: Verifcation of Erlang Programs using Abstract Interpretation and Model Checking — Extended Version

1999-03 * R. Gallersdörfer, M. Jarke, M. Nicola: The ADR Replication Manager

1999-04 María Alpuente, Michael Hanus, Salvador Lucas, Germán Vidal: Specialization of Functional Logic Programs Based on Needed Narrowing

1999-05 * W. Thomas (Ed.): DLT 99 - Developments in Language Theory Fourth International Conference

1999-06 * Kai Jakobs, Klaus-Dieter Kleefeld: Informationssysteme für die angewandte historische Geographie

1999-07 Thomas Wilke: CTL+ is exponentially more succinct than CTL

1999-08 Oliver Matz: Dot-Depth and Monadic Quantifier Alternation over Pictures

2000-01 * Jahresbericht 1999

2000-02 Jens Vöge, Marcin Jurdzinski A Discrete Strategy Improvement Algorithm for Solving Parity Games

2000-03 D. Jäger, A. Schleicher, B. Westfechtel: UPGRADE: A Framework for Building Graph-Based Software Engineering Tools

2000-04 Andreas Becks, Stefan Sklorz, Matthias Jarke: Exploring the Semantic Structure of Technical Document Collections: A Cooperative Systems Approach

2000-05 Mareike Schoop: Cooperative Document Management

2000-06 Mareike Schoop, Christoph Quix (eds.): Proceedings of the Fifth International Workshop on the Language-Action Perspective on Communication Modelling

2000-07 * Markus Mohnen, Pieter Koopman (Eds.): Proceedings of the 12th International Workshop of Functional Languages

2000-08 Thomas Arts, Thomas Noll: Verifying Generic Erlang Client-Server Implementations

2001-01 * Jahresbericht 2000

2001-02 Benedikt Bollig, Martin Leucker: Deciding LTL over Mazurkiewicz Traces

2001-03 Thierry Cachat: The power of one-letter rational languages

2001-04 Benedikt Bollig, Martin Leucker, Michael Weber: Local Parallel Model Checking for the Alternation Free mu-Calculus
2001-05 Benedikt Bollig, Martin Leucker, Thomas Noll: Regular MSC Languages
2001-06 Achim Blumensath: Prefix-Recognisable Graphs and Monadic Second-Order Logic
2001-07 Martin Grohe, Stefan Wöhrle: An Existential Locality Theorem
2001-08 Mareike Schoop, James Taylor (eds.): Proceedings of the Sixth International Workshop on the Language-Action Perspective on Communication Modelling
2001-09 Thomas Arts, Jürgen Giesl: A collection of examples for termination of term rewriting using dependency pairs
2001-10 Achim Blumensath: Axiomatising Tree-interpretable Structures
2001-11 Klaus Indermark, Thomas Noll (eds.): Kolloquium Programmiersprachen und Grundlagen der Programmierung
2002-01 * Jahresbericht 2001
2002-02 Jürgen Giesl, Aart Middeldorp: Transformation Techniques for Context-Sensitive Rewrite Systems
2002-03 Benedikt Bollig, Martin Leucker, Thomas Noll: Generalised Regular MSC Languages
2002-04 Jürgen Giesl, Aart Middeldorp: Innermost Termination of Context-Sensitive Rewriting
2002-05 Horst Lichter, Thomas von der Maßen, Thomas Weiler: Modelling Requirements and Architectures for Software Product Lines
2002-06 Henry N. Adorna: 3-Party Message Complexity is Better than 2-Party Ones for Proving Lower Bounds on the Size of Minimal Nondeterministic Finite Automata
2002-07 Jörg Dahmen: Invariant Image Object Recognition using Gaussian Mixture Densities
2002-08 Markus Mohnen: An Open Framework for Data-Flow Analysis in Java
2002-09 Markus Mohnen: Interfaces with Default Implementations in Java
2002-10 Martin Leucker: Logics for Mazurkiewicz traces
2002-11 Jürgen Giesl, Hans Zantema: Liveness in Rewriting
2003-01 * Jahresbericht 2002
2003-02 Jürgen Giesl, René Thiemann: Size-Change Termination for Term Rewriting
2003-03 Jürgen Giesl, Deepak Kapur: Deciding Inductive Validity of Equations
2003-04 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Improving Dependency Pairs
2003-05 Christof Löding, Philipp Rohde: Solving the Sabotage Game is PSPACE-hard
2003-06 Franz Josef Och: Statistical Machine Translation: From Single-Word Models to Alignment Templates
2003-07 Horst Lichter, Thomas von der Maßen, Alexander Nyßen, Thomas Weiler: Vergleich von Ansätzen zur Feature Modellierung bei der Softwareproduktlinienentwicklung
2003-08 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp, Stephan Falke: Mechanizing Dependency Pairs
2004-01 * Fachgruppe Informatik: Jahresbericht 2003

2004-02 Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic

2004-03 Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting

2004-04 Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming

2004-05 Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming

2004-06 Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming

2004-07 Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination

2004-08 Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information

2004-09 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity

2004-10 Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules

2005-01 * Fachgruppe Informatik: Jahresbericht 2004

2005-02 Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: “Aachen Summer School Applied IT Security”

2005-03 Jürgen Giesl, René Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions

2005-04 Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem

2005-05 Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots

2005-06 Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information

2005-07 Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks

2005-08 Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut

2005-09 Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures

2005-10 Benedikt Bollig: Automata and Logics for Message Sequence Charts

2005-11 Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture

2005-12 Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems

2005-13 Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments

2005-14 Felix C. Freiling, Sukumar Ghosh: Code Stabilization

2005-15 Uwe Naumann: The Complexity of Derivative Computation

2005-16 Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)

2005-17 Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)

2005-18 Thomas von der Maßen, Klaus Müller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlußbericht des GI-Arbeitskreises "Features"

2005-19 Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers

2005-20 Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.

2005-21 Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited

2005-22 Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins

2005-23 Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves

2005-24 Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks

2006-01 * Fachgruppe Informatik: Jahresbericht 2005

2006-02 Michael Weber: Parallel Algorithms for Verification of Large Systems

2006-03 Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler

2006-04 Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation

2006-05 Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F

2006-06 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color

2006-07 Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group "Requirements Management Tools for Product Line Engineering"

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices

2006-16 Rafael Ballagas, Jan Borchers: Selexels: a Conceptual Framework for Pointing Devices with Low Expressiveness

2006-17 Eric Lee, Henning Kiel, Jan Borchers: Scrolling Through Time: Improving Interfaces for Searching and Navigating Continuous Audio Timelines

2007-01 * Fachgruppe Informatik: Jahresbericht 2006

2007-02 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, René Thiemann, and Harald Zankl: SAT Solving for Termination Analysis with Polynomial Interpretations

2007-03 Jürgen Giesl, René Thiemann, Stephan Swiderski, and Peter Schneider-Kamp: Proving Termination by Bounded Increase

* These reports are only available as a printed version.

Please contact biblio@informatik.rwth-aachen.de to obtain copies.