Handbook Of Graph Grammars And Computing By Graph Transformation Vol 1

This is likewise one of the factors by obtaining the soft documents of this Handbook Of Graph Grammars And Computing By Graph Transformation Vol 1 by online. You might not require more get older to spend to go to the book initiation as skillfully as search for them. In some cases, you likewise accomplish not discover the broadcast Handbook Of Graph Grammars And Computing By Graph Transformation Vol 1 that you are looking for. It will enormously squander the time.

However below, following you visit this web page, it will be therefore completely simple to get as competently as download guide Handbook Of Graph Grammars And Computing By Graph Transformation Vol 1

It will not receive many period as we tell before. You can accomplish it even if pretense something else at home and even in your workplace. suitably easy! So, are you question? Just exercise just what we manage to pay for below as skillfully as evaluation Handbook Of Graph Grammars And Computing By Graph Transformation Vol 1 what you subsequent to to read!

Graph Structure and Monadic Second-Order Logic Bruno Courcelle 2012-06-14 The study of graph structure has advanced in recent years with great strides: finite graphs can be described algebraically, enabling them to be constructed out of more basic elements. Separately the properties of graphs can be studied in a logical language called monadic second-order logic. In this book, these two features of graph structure are brought together for the first time in a presentation that unifies and synthesizes research over the last 25 years. The authors not only provide a thorough description of the theory, but also detail its applications, on the one hand to the construction of graph algorithms, and, on the other to the extension of formal language theory to finite graphs. Consequently the book will be of interest to graduate students and researchers in graph theory, finite model theory, formal language theory, and complexity theory.

Inductive Logic Programming Tamas Horváth 2003-10-24 This book constitutes the refereed proceedings of the 13th International Conference on Inductive Logic Programming, ILP 2003, held in Szeged, Hungary in September/October 2003. The 23 revised full papers presented were carefully reviewed and selected from 53 submissions. Among the topics addressed are multirelational data mining, complexity issues, theory revision, clustering, mathematical discovery, relational reinforcement learning, multirelational learning, inductive inference, description logics, grammar systems, and inductive learning.

Concurrency, Parallelism and Distribution Hartmut Ehrig 1999

Handbook of Cloud Computing Borko Furht 2010-09-11 Cloud computing has become a significant technology trend. Experts believe cloud computing is currently reshaping information technology and the IT marketplace. The advantages of using cloud computing include cost savings, speed to market, access to greater computing resources, high availability, and scalability. Handbook of Cloud Computing includes contributions from world experts in the field of cloud computing from academia, research laboratories and private industry. This book presents the systems, tools, and services of the leading providers of cloud computing; including Google, Yahoo, Amazon, IBM, and Microsoft. The basic concepts of cloud computing and cloud computing applications are also introduced. Current and future technologies applied in cloud computing are also discussed. Case studies, examples, and exercises are provided throughout. Handbook of Cloud Computing is intended for advanced-level students and researchers in computer science and electrical engineering as a reference book. This handbook is also beneficial to computer and system infrastructure designers, developers, business managers, entrepreneurs and investors within the cloud computing related industry.

Handbook of Graph Grammars and Computing by Graph Transformation Hartmut Ehrig 1999 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then, the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas, it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact, within the area of graph grammars, graph transformation is considered a fundamental computation paradigm where computation includes specification, programming, and implementation. Over the last three decades, graph grammars have developed at a steady pace into a theoretically attractive and important-for-applications research field. Volume 2 of the indispensable Handbook of Graph Grammars and Computing by Graph Transformations considers applications to functional languages, visual and object-oriented languages, software engineering, mechanical engineering, chemical process engineering, and images. It also presents implemented specification languages and tools, and structuring and modularization concepts for specification languages. The contributions have been written in a tutorial/survey style by the top experts in the corresponding areas. This volume is accompanied by a CD-Rom containing implementations of specification environments based on graphtransformation systems, and tools whose implementation is based on the use of graph transformation systems.

Unconventional Programming Paradigms Jean-Pierre Banatre 2005-07-15 Unconventional approaches to programming have long been developed, in various niches and out of curiosity, and they constitute a reservoir of alternative avenues to deal with unknown programming challenges. New paradigms of programming are currently experiencing a renewed period of interest and growth to cope with problems from specific application domains. This book constitutes the thoroughly refereed post-proceedings of the International Workshop on Unconventional Programming Paradigms, UPP 2004, held at Le Mont Saint Michel, France, in September 2004. The 26 revised full papers presented together with an invited paper on quantum computing were carefully reviewed for presentation in the book. The papers are organized in topical sections on chemical computing, amorphous computing, bio-inspired computing, autonomic computing, and generative programming.

Fundamentals of Algebraic Graph Transformation Hartmut Ehrig 2006-05-01 This is the first textbook treatment of the algebraic approach to graph transformation, based on algebraic structures and category theory. It contains an introduction to classical graphs. Basic and advanced results are first shown for an abstract form of replacement systems and are then instantiated to several forms of graph and Petri net transformation systems. The book develops typed attributed graph transformation and contains a practical case study.

Formal and Natural Computing Wilfried Brauer 2003-08-01 This book presents state of the art research in theoretical computer science and related ?elds. In particular, the following areas are discussed: automata theory, formal languages and combinatorics of words, graph transformations, Petri nets, concurrency, as well as natural and molecular computing. The articles are written by leading researchers in these areas. The writers were originally invited to contribute to this book but then the normal refereeing procedure was applied as well. All of the articles deal with some issue that has been under vigorous study during recent years. Still, the topics range from very classical ones to issues raised only two or three years ago. Both survey articles and papers attacking speci?c research problems are included. The book highlights some key issues of theoretical computer science, as they seem to us now at the beginning of the new millennium. Being a comprehensive overview of some of the most active current research in theoretical computer science, it should be of de?nite interest for all researchers in the areas covered. The topics range from basic decidability and the notion of information to graph grammars and graph transformations, and from trees and traces to aqueous algorithms, DNA encoding and self-assembly. Special e?ort has been given to lucid presentation. Therefore, the book should be of interest also for advanced students.

Handbook of Graph Grammars and Computing by Graph Transformation Grzegorz Rozenberg 1999

Lectures on Concurrency and Petri Nets Jörg Desel 2004-06-14 This tutorial volume originates from the 4th Advanced Course on Petri Nets, ACPN 2003, held in Eichsttt, Germany in September 2003. In addition to lectures given at ACPN 2003, additional chapters have been commissioned to give a well-balanced presentation of the state of the art in the area. This book will be useful as both a reference for those working in the area as well as a study book for the reader who is interested in an up-to-date overview of research and development in concurrent and distributed systems; of course, readers specifically interested in theoretical or applicational aspects of Petri nets will appreciate the book as well.

Graph Transformations Hartmut Ehrig 2004-11-11 This book constitutes the refereed proceedings of the Second International Conference on Graph Transformation, ICGT 2004, held in Rome, Italy, in September/October 2004. The 26 revised full papers presented together with three invited contributions and summaries of 2 tutorials and 5 workshops were carefully reviewed and selected from 58 submissions. The papers are organized in topical sections on integration technology, chemistry and biology, graph transformation concepts, DPO theory for high-level structures, analysis and testing, graph theory and algorithms, application conditions and logic, transformation of special structures, and object-orientation.

Graph Transformation Andrea Corradini 2003-06-30 ICGT 2002 was the ?rst International Conference on Graph Transformation following a series of six international workshops on graph grammars with - plications in computer science, held in Bad Honnef (1978), Osnabruc ? k (1982), Warrenton (1986), Bremen (1990), Williamsburg (1994), and Paderborn (1998). ICGT 2002 was held in Barcelona (Spain), October 7–12, 2002 under the a- pices of the European Association of Theoretical Computer Science (EATCS), the European Association of Software Science and Technology (EASST), and the IFIP Working Group 1.3, Foundations of Systems Speci?cation. The scope of the conference concerned graphical structures of various kinds (like graphs, diagrams, visual sentences and others) that are useful to describe complex structures and systems in a direct and intuitive way. These structures are often augmented by formalisms which add to the static description a further dimension, allowing for the modeling of the evolution of systems via all kinds of transformations of such graphical structures. The ?eld of Graph Transformation is concerned with the theory, applications, and implementation issues of such formalisms. The theory is strongly related to areas such as graph theory and graph - gorithms, formal language and parsing theory, the theory of concurrent and distributed systems, formal speci?cation and veri?cation, logic, and semantics.

Graph Transformations Hartmut Ehrig 2010-09-27 This book constitutes the proceedings of the 5th International Conference on Graph Transformations, ICGT 2010, held in Twente, The Netherlands, in September/October 2010. The 22 papers presented were carefully reviewed and selected from 48 submissions. These papers mirror the wide-ranged ongoing research activities in the theory and application of graph transformation. They are concerned with different kinds of graph transformation approaches, their algebraic foundations, composition and analysis, the relation to logic, as well as various applications, mainly to model transformation and distributed systems.

Foundations of Software Science and Computation Structures Spain) FOSSACS 2004 (2004 : Barcelona 2004-03-19 This book constitutes the refereed proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures, FOSSACS 2004, held in Barcelona, Spain in March/April 2004. The 34 revised full papers presented together with the abstracts of 2 invited talks were carefully reviewed and selected from over 130 submissions. Among the topics addressed are lambda calculus, cryptographic protocol analysis, graphs and grammar systems, decision theory, bisimulation, rewriting, normalization, specification, verification, process calculi, mobile code, automata, program semantics, dynamic logics, timed languages, security analysis, information-theoretical aspects.

Unifying Petri Nets Hartmut Ehrig 2014-03-12 Since their introduction nearly 40 years ago, research on Petri nets has diverged in many different directions. Various classes of Petri net, motivated either by theory or applications, with its own specific features and methods of analysis, have been proposed and studies in depth. These successful developments have led to a very heterogeneous landscape of diverse models, and this, in turn, has stimulated research on concepts and approaches that contribute to unifying and structuring the diverse landscape. This state-of-the-art survey presents the most relevant approaches to unifying Petri nets in a systematic and coherent way. The 14 chapters written by leading researchers are organized in topical sections on application-oriented approaches, unifying frameworks, and theoretical approaches.

Handbook of Graph Grammars and Computing by Graph Transformation H Ehrig 1999-10-20 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then, the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas, it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact, within the area of graph grammars, graph transformation is considered as a fundamental computation paradigm where computation includes specification, programming, and implementation. Over the last three decades, graph grammars have developed at a steady pace into a theoretically attractive and important-for-applications research field. Volume 2 of the indispensable Handbook of Graph Grammars and Computing by Graph Transformations considers applications to functional languages, visual and object-oriented languages, software engineering, mechanical engineering, chemical process engineering, and images. It also presents implemented specification languages and tools, and structuring and modularization concepts for specification languages. The contributions have been written in a tutorial/survey style by the top experts in the corresponding areas. This volume is accompanied by a CD-Rom containing implementations of specification environments based on graph transformation systems, and tools whose implementation is based on the use of graph transformation systems. Contents:Term Rewriting and Functional LanguagesVisual and Object-Oriented LanguagesApplications to Software EngineeringApplications to Engineering DisciplinesApplications to PicturesImplemented Specification Languages and ToolsStructuring and Modularization Concepts Readership: Students and researchers interested in modern developments in computer science, and in particular in the use of modern formal methods in applied computer science.

Hyperedge Replacement: Grammars and Languages Annegret Habel 1992-12-08 The area of graph grammars is theoretically attractive and well motivated byvarious applications. More than 20 years ago, the concept of graph grammars was introduced by A. Rosenfeld as a formulation of some problems in pattern recognition and image processing, as well as by H.J. Schneider as a method for data type specification. Within graph-grammar theory one maydistinguish the set-theoretical approach, the algebraic approach, and the logical approach. These approaches differ in the method in which graph replacement is described. Specific approaches, node replacement and hyperedge replacement, concern the basic units of a hypergraph, nodes and hyperedges. This monograph is mainly concerned with the hyperedge-replacement approach. Hyperedge-replacement grammars are introduced as a device for generating hypergraph languages including graph languages and string languages. The concept combines a context-free rewriting with a comparatively large generative power. The volume includes a foreword by H. Ehrig.

Handbook of Graph Grammars and Computing by Graph Transformation Hartmut Ehrig 1999 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then, the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas, it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact, within the area of graph grammars, graph transformation is considered a fundamental computation paradigm where computation includes specification, programming, and implementation. Over the last three decades, graph grammars have developed at a steady pace into a theoretically attractive and important-for-applications research field. Volume 3 of the 'indispensable Handbook of' Graph Grammars and Computing by Graph Transformations presents the research on concurrency, parallelism, and distribution -- important paradigms of modern science. The topics considered include semantics for concurrent systems, modeling of concurrency, mobile and coordinated systems, algebraic specifications, Petri nets, visual design of distributed systems, and distributed algorithms. The contributions have been written in a tutorial/survey style by the top experts.

Applications, Languages and Tools Hartmut Ehrig 1999

Handbook of Graph Grammars and Computing by Graph Transformation Grzegorz Rozenberg 1997-02-27 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact within the area of graph grammars, graph transformation is considered a fundamental programming paradigm where computation includes specification, programming, and implementation. Over the last 25-odd years graph grammars have developed at a steady pace into a theoretically attractive and well-motivated research field. In particular, they are now based on very solid foundations, which are presented in this volume. Volume 1 of the indispensable Handbook of Graph Grammars

and Computing by Graph Transformations includes a state-of-the-art presentation of the foundations of all the basic approaches to rule-based graph specification and transformation: algebraic approach, logic approach, node-based rewriting, (hyper)edge-based rewriting, programmed graph rewriting, and 2-structures. The book has been written in a tutorial/survey style to enhance its usefulness. Contents:Node Replacement Graph Grammars (J Engelfriet & G Rozenberg)Hyperedge Replacement Graph Grammars (F Drewes et al.)The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic (B Courcelle)Algebraic Approaches to Graph Transformation — Part I: Basic Concepts and Double Pushout Approach (A Corradini et al.)Algebraic Approaches to Graph Transformation — Part II: Single Pushout Approach and Comparison with Double Pushout Approach (H Ehrig et al.)2-Structures — A Framework for Decomposition and Transformation of Graphs (A Ehrenfeucht et al.)Programmed Graph Replacement Systems (A Schürr) Readership: Computer scientists and mathematicians. keywords:

CONCUR 2004 -- Concurrency Theory Philippa Gardner 2004-08-19 assisted us in the evaluation of the submitted papers.

Graph Transformations in Computer Science Hans J. Schneider 2014-01-15

Handbook of Graph Grammars and Computing by Graph Transformation: Concurrency, parallelism, and distribution Grzegorz Rozenberg 1997

Formal Methods in Software and Systems Modeling Hans-Jörg Kreowski 2005-02-09 By presenting state-of-the-art research results on various aspects of formal and visual modeling of software and systems, this book commemorates the 60th birthday of Hartmut Ehrig. The 24 invited reviewed papers are written by students and collaborators of Hartmut Ehrig who are established researchers in their fields. Reflecting the scientific interest and work of Hartmut Ehrig, the papers fall into three main parts on graph transformation, algebraic specification and logic, and formal and visual modeling.

Graph Grammars and Their Application to Computer Science Janice Cuny 2014-01-15

Handbook of Graph Grammars and Computing by Graph Transformation Grzegorz Rozenberg 1997-01-01 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact within the area of graph grammars, graph transformation is considered a fundamental programming paradigm where computation includes specification, programming, and implementation.

Foundations Grzegorz Rozenberg 1997-01 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, visual languages, and many others.The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact within the area of graph grammars, graph transformation is considered a fundamental programming paradigm where computation includes specification, programming, and implementation.Over the last 25-odd years graph grammars have developed at a steady pace into a theoretically attractive and well-motivated research field. In particular, they are now based on very solid foundations, which are presented in this volume. Volume 1 of the indispensable Handbook of Graph Grammars and Computing by Graph Transformations includes a state-of-the-art presentation of the foundations of all the basic approaches to rule-based graph specification and transformation: algebraic approach, logic approach, node-based rewriting, (hyper)edge-based rewriting, programmed graph rewriting, and 2-structures. The book has been written in a tutorial/survey style to enhance its usefulness.

Handbook of Graph Grammars and Computing by Graph Transformation H Ehrig 1999-08-30 Graph grammars originated in the late 60s, motivated by considerations about pattern recognition and compiler construction. Since then, the list of areas which have interacted with the development of graph grammars has grown quite impressively. Besides the aforementioned areas, it includes software specification and development, VLSI layout schemes, database design, modeling of concurrent systems, massively parallel computer architectures, logic programming, computer animation, developmental biology, music composition, visual languages, and many others. The area of graph grammars and graph transformations generalizes formal language theory based on strings and the theory of term rewriting based on trees. As a matter of fact, within the area of graph grammars, graph transformation is considered as a fundamental computation paradigm where computation includes specification, programming, and implementation. Over the last three decades, graph grammars have developed at a steady pace into a theoretically attractive and important-for-applications research field. Volume 3 of the indispensable Handbook of Graph Grammars and Computing by Graph Transformations presents the research on concurrency, parallelism, and distribution — important paradigms of modern computer science. The topics considered include semantics for concurrent systems, modeling of concurrency, mobile and coordinated systems, algebraic specifications, Petri nets, visual design of distributed systems, and distributed algorithms. The contributions have been written in a tutorial/survey style by the top experts. Contents:Graph Relabelling Systems and Distributed Algorithms (I Litovsky et al.)Actor Grammars and Local Actions (D Janssens)Concurrent Semantics of Algebraic Graph Transformations (P Baldan et al.)Modeling Concurrent, Mobile and Coordinated Systems via Graph Transformations (U Montanari et al.)Distributed Graph Transformation with Application to Visual Design of Distributed Systems (I Fischer et al.)High-Level Replacement Systems Applied to Algebraic Specifications and Petri Nets (H Ehrig et al.)Describing Systems of Processes by Means of High-Level Replacement (H J Schneider) Readership: Students and researchers interested in modern developments in computer science and in particular in three modern paradigms of computer science — concurrency, parallelism, and distribution. Keywords:

Current Trends in Theoretical Computer Science Gheorghe P?un 2001 The scientific developments at the end of the past millennium were dominated by the huge increase and diversity of disciplines with the common label "computer science". The theoretical foundations of such disciplines have become known as theoretical computer science. This book highlights some key issues of theoretical computer science as they seem to us now, at the beginning of the new millennium. The text is based on columns and tutorials published in the Bulletin of the European Association for Theoretical Computer Science in the period 1995 -- 2000. The columnists themselves selected the material they wanted for the book, and the editors had a chance to update their work. Indeed, much of the material presented here appears in a form quite different from the original. Since the presentation of most of the articles is reader-friendly and does not presuppose much knowledge of the area, the book constitutes suitable supplementary reading material for various courses in computer science.

Theory and Application of Graph Transformations Hartmut Ehrig 2014-01-15

Fundamentals of Algebraic Specification 2 Hartmut Ehrig 2011-12-10 Since the early seventies concepts of specification have become central in the whole area of computer science. Especially algebraic specification techniques for abstract data types and software systems have gained considerable importance in recent years. They have not only played a central role in the theory of data type specification, but meanwhile have had a remarkable influence on programming language design, system architectures, and software tools and environments. The fundamentals of algebraic specification lay a basis for teaching, research, and development in all those fields of computer science where algebraic techniques are the subject or are used with advantage on a conceptual level. Such a basis, however, we do not regard to be a synopsis of all the different approaches and achievements but rather a consistently developed theory. Such a theory should mainly emphasize elaboration of basic concepts from one point of view and, in a rigorous way, reach the state of the art in the field. We understand fundamentals in this context as: 1. Fundamentals in the sense of a carefully motivated introduction to algebraic specification, which is understandable for computer scientists and mathematicians. 2. Fundamentals in the sense of mathematical theories which are the basis for precise definitions, constructions, results, and correctness proofs. 3. Fundamentals in the sense of concepts from computer science, which are introduced on a conceptual level and formalized in mathematical terms.

Term Graph Rewriting M. R. Sleep 1993-06-08 Charting the progress of the theory, implementations and applications of rewriting models of computation, this volume represents a key resource for researchers. Much of modern computer science - particularly 'new generation languages' - relies heavily on rewriting systems, and Term Graph Rewriting offers an insight into the very foundations. A major feature of the book is the presentation of theoretical advances, such as: a new theory of transfinite term rewriting, a single pushout categorical model of graph rewriting, and an abstract interpretation for term graph rewriting. In order to fully appreciate the practical but novel benefits offered by term and graph rewriting techniques, readers are here presented with the most comprehensive study conducted to date. A crucial theme throughout is the relationship between term and graph rewriting which provides a useful context for considering implementations - ranging from high performance sequential to parallel distributed memory, and applications - specifically to logic and functional programming.

Graph Grammars and Their Application to Computer Science Hartmut Ehrig 2014-01-15

Transformation of Knowledge, Information and Data Patrick van Bommel 2005-01-01 This book considers transformations within the context of computing science and information science, as they are essential in changing organizations. It not only considers transformations of structured models, rather, the transformation of instances (i.e. the actual contents of those structures) is addressed as well.

Recent Trends in Algebraic Development Techniques José Luiz Fiadeiro 2007-06-20 This book constitutes the thoroughly refereed postproceedings of the 18th International Workshop on Algebraic Development Techniques, WADT 2006, held in La Roche en Ardenne, Belgium, June 2006. The 10 revised full papers focus on the algebraic approach to the specification and development of systems and address topics such as formal methods for system development, specification languages and methods, and distributed and mobile systems.

Graph Transformations Hartmut Ehrig 2014-01-15

Context Free Hypergraph Grammars Renate Klempien-Hinrichs 2002-01

Handbook of Graph Grammars and Computing by Graph Transformation: Applications, languages and tools Grzegorz Rozenberg 1997

Applications of Graph Transformations with Industrial Relevance Manfred Nagl 2003-07-31 This book constitutes the thoroughly refereed post-proceedings of the International Workshop on Graph Transformation with Industrial Relevance, AGTIVE'99, held in Kerkrade, The Netherlands, in June 1999. The 28 revised full papers presented went through an iterated process of reviewing and revision. Also included are three invited papers, 10 tool demonstrations, a summary of a panel discussion, and lists of graph transformation systems and books on graph transformations. The papers are organized in sections on modularization concepts, distributed systems modeling, software architecture: evolution and reengineering, visual graph transformation languages, visual language modeling and tool development, knowledge modeling, image recognition and constraint solving, process modeling and view integration, and visualization and animation tools.

Foundation of Software Science and Computation Structures Jerzy Tiuryn 2000-03-15 ETAPS2000wasthethirdinstanceoftheEuropeanJointConferencesonTheory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprisedv e conferences (FOSSACS, FASE, ESOP,CC, TACAS), ve satellite workshops (CBS, CMCS, CoFI, GRATRA, INT), seven invited lectures, a panel discussion, and ten tutorials. The events that comprise ETAPS address various aspects of the system - velopmentprocess,includingspeci cation,design,implementation,analysis,and improvement. The languages, methodologies, and tools which support these - tivities are all well within its scope. Die rent blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive. ETAPS is a loose confederation in which each event retains its own identity, with a separate program committee and independent proceedings. Its format is open-ended, allowing it to grow and evolve as time goes by. Contributed talks and system demonstrations are in synchronized parallel sessions, with invited lectures in plenary sessions. Two of the invited lectures are reserved for \u- fying" talks on topics of interest to the whole range of ETAPS attendees.