Fourth Annual Progress Report of the GETGRATS Research Network Contract ERB FMRXCT960061

Andrea Corradini
Dipartimento di Informatica, Pisa
andrea@di.unipi.it

20th November 2000

Contents

1 Progress of the Network during the Fourth year

1.1 Comparison of accomplished work and objectives in the Project Programme

On February 18, 2000, the Network Coordinator submitted to the EC officer the request for a further one-year prolongation of the Network, motivated by the difficulties found in recruiting a sufficient number of network fellows, and by the possibility of better organizing the remaining GETGRATS events. In June 2000 the prolongation was formalized with a contract amendment, therefore the Network will end on August 31, 2001.

Because of the prolongation, a few changes to the schedule of the events were decided. More precisely, the Focus Area Meeting of Focus Area (a) has been postponed to mid-year 2001, and as the closing GETGRATS event it will be larger than the other Focus Area Meetings. Furthermore, GETGRATS will sponsor (as an additional event with respect to Project Programme) the UNIGRA workshop, which will be held in Genova (I) on March 31 and April 1, 2001, as satellite event of the International Conference ETAPS'01.

The remaining events and organizational activities accomplished during the fourth year of the Network meet the objectives, schedule and milestones set out in the Project Programme (Annex I of the contract), and further detailed in the Third Year Progress Report.

Significant progress has been made in the research activity in all the seven Focus Areas in which the research themes of the project are divided, as described in the next subsection and in the Appendix (Section 4).

As described in Section 1.3, a change of the distribution of the Network budget among the partners has been decided during the Fifth Team Leader Meeting, with the agreement of all Team Leaders. As the changes did not affect any partner for more than 50% of his budget, no contract amendment was needed, according to Article 14.4 of Annex II of the contract.

1.2 Scientific Highlights

A quite detailed description of the results of the scientific research carried on during the fourth year of the project is presented in the Appendix (Section 4).

Significant contributions have been made to research themes of all the seven Fous Areas. Among them we recall the proposal of using Monadic Second Order Logic (MSOL) formulas for expressing properties of graph drawings; the study of graph (de)composition mechanisms based on pullbacks; the analysis of the power of context-free graph grammars with respect to the generated hypergraph languages; the progress in the investigation of dynamic labelled 2-structures as an algebraic model of graph transformation; the analysis of the complexity of deciding whether a tree transducer yields exponentially large output trees; the strengthening of the expressivity of the process notion for LAS obtained by adding a temporal order on actions; the formalization of agent-based systems by use of GTS semantics; the progress in the analysis of distributed algorithms based on graph relabelling systems; the application of distributed graph transformations to dynamic architecture configuration for distributed systems and viewpoint-oriented software development; the study of rule-based tranformation of hierarchically structured (hyper)graphs; the definition of a notion of transformation unit with interleaving semantics, and of transformation modules; the progress in the study of refinement of GTSs, as well as in the use of GTS for the specification of software architecture styles.

Another scientific highlight is the formulation in terms of graph transformation of the main features of the gene assembly in ciliates. Using this formulation one can reason about the strategies for gene assembly in terms of graph rewriting systems. This is a major point for the area of DNA Computing in vivo, because gene assembly in ciliates is one of the most sophisticated DNA processing known in living organisms.

During the reported period, three Ph.D. theses were completed by young researchers whose resarch activity was partly supported by the Network. In Marta Simeoni's thesis [Sim00], an abstract categorical theory of modularization of graph transformation systems has been developed which is formalized independently of any particular approach. In Paolo Baldan's thesis [Bal00], some classical approaches to the functorial semantics of Petri nets were generalized to graph trasnformation systems, as well as to two enriched models of nets, namely contextual nets and nets with inhibitor arcs. The thesis by Sabine Kuske [Kus00b] presents a thorough study of the theory of transformation units, an approach to construct large graph transformation systems from small ones in a structured and systematic way.

1.3 Principal Networking and Coordination Activities

Here is a short description of the main events organized within the network during the fourth year:

1. The Focus Area Meeting on *Analysis and Verification Techniques* (Focus Area (c)) was held in Bremen (D), on December 9–10, 1999. There where 16 participants from four sites involved in the Focus Area, namely UIA, TUB, UNIBREMEN and LaBRI. Different approaches to the verification of graph transformation were proposed during the talks and discussed in a general discussion.

2. The *International Workshop on Graph Transformation and Applications (GRATRA 2000)* was held in Berlin on March 25-27, 2000, as a joint workshop of the GETGRATS Network and of the ESPRIT working group *Applications of Graph Transformation (APPLIGRAPH).* The workshop focus was on both theory and practice of graph grammars and graph transformation systems, and it was organized as a satellite event of ETAPS 2000, the joint *European Conference on Theory and Practice of Software.*

The workshop was intended to bring together not only researchers of the area of graph transformation, but also researchers from other areas, potentially interested in graph transformation, and attending the ETAPS 2000 conferences. For this reason, the workshop was open to anybody, and indeed there were about 70 participants, among which about 35 GETGRATS members.

The program committee selected 28 papers to be presented at the workshop, and it assigned two best paper awards (one to for the best theoretical contribution, and one for the best paper on applications). Selected papers of the workshop will be published in two special issues of journals: one in *Mathematical Structures in Computer Science* and the other in *Science of Computer Programming*. There were invited lectures by G. Engels on the role of graph transformation in software engineering and by G. Rozenberg on DNA computing and graph transformation. Furthermore, a panel discussion took place. All contributions to the workshop are collected in [ET00].

3. The Fifth Team Leader Meeting of GETGRATS took place in Berlin on March 26, 2000, during the GRATRA 2000 workshop. There were 13 participant, representing all the sites of the Network. The main topics of discussion were the current status of the felloships, the budgetary situation of the Network, and the planning of the remaining events of the network.

   At the time of the Meeting the further one-year prolongation of the Network was not yet formalized, but was considered very likely to be accepted. Therefore some events have been planned for the period September 2000 – August 2001 as well. In particular, the following events were planned (the first three already took place, and are described later):

   (a) FAM for Focus Area (d) on "Abstract Semantics", in Antwerpen on April 28, 2000.
   (b) FAM for Focus Area (b) on "Classification and Expressive", in Leiden on May 27–28, 2000.
   (c) Workshop on Graph Transformation and Visual Modeling Techniques, July 14–15, 2000, satellite of ICALP 2000 in Geneve.
   (d) Workshop on Uniform Approaches to Graphical Process Specification Techniques (UNIGRA), March 31 – April 1, 2001, satellite of ETAPS 2001 in Genova.
   (e) FAM for Focus Area (a) on "Foundation, Unification, Combination and Comparison", in Bordeaux, approximatively in June 2001. Jointly with this FAM, a "Closing Event" of the GETGRATS project will be organized, where summary presentations of all the activities performed during the project will be given.

   Concerning the Estimated Breakdown of Allowable Costs among partners, some changes were agreed upon. As a consequence, also the total number of months for which the partners will employ young researchers financed by the project is changed. The following table shows for each partner the new total budget and the new number of months.

PARTNER	**EURO**	**months**
UNIPISA	162,000	36
UIA	82,120	23
TUB	150,600	41
LaBRI	134,040	33
UNIBREMEN	132,000	36
RUL	134,520	37
UNIROMA1	100,720	28
TOTAL	**896,000**	**234**

4. The Focus Area Meeting on *Abstract Semantics* (Focus Area (d)) was held in Antwerpen (BE), on April 27–28, 2000. There where 6 participants from five sites of the network, namely UIA, TUB, UNIROMA1, UNIBREMEN, and UNIPISA. Besides technical presentations by the participants, there has been a general discussion about the role of Graph Transformation Systems in semantics. Among the various issues, the following were discussed: 1) the possibility to develop a GTS semantics for UML; 2) the need of a common exchange format for different types of GTS; 3) pro's and con's of an integration of the GraGra community with research communities of related fields, such as Abstract State Machines.

5. The Focus Area Meeting on *Classification and Expressive Power* (Focus Area (b)) was held in Leiden (NL), on May 26–27, 2000. There were 11 participants from five sites, namely UIA, RUL, LaBRI, UNIBREMEN, UNIPISA. Besides technical presentations by the participants, there has been a general discussion about the presentations themselves and about organizational aspects of the network.

6. The Workshop on Graph Transformation and Visual Modeling Techniques (GT-VMT) was held in Geneva, Switzerland, on July 14 and 15, 2000, as a satellite event to ICALP 2000. The workshop was organized by A. Corradini and R. Heckel (former GETGRATS fellow, now at the University of Paderborn). Many GETGRATS Team Leaders were involved either as Program Committee members or as invited speakers.

   The aim of the workshop was to bring together scientists and researchers interested in discussing formal methodologies for the definition of syntax and semantics of visual modelling techniques, with a particular emphasis on techniques based on Graph Transformation. There were about 30 participants, out of which about 15 from the network. The 11 contributions selected for presentation are collected in the proceeding book [CH00].

1.4 Progress with the Appointment of Young Researchers

During the first four years of the Network duration (i.e., from September 1, 1996, to August 31, 2000) about 41 grants have been assigned, financing a total of about 215 months (52 months during the fourth year), out of a total of 234 for the whole duration of the project. The last call for applications was published in December 1999, an it will remain open till all the remaining grants are assigned.

1.5 Difficulties

As reported in Section 1.1, at the beginning of the fourth year of the Network we realized that it would have been very hard to find enough network fellows to assign grants for the remaining man-months; furthermore the use of the networking budget was unevenly distributed among the partners. Therefore on the one hand we asked for a further prolongation of the Network, which has been formalized with a contract amendment, on the other hand we decided a redistribution of the project budget.

No other significant difficulties arose during the fourth year of the project.

2 Information on each Participant

The following pages are copies of the form "Information on each Participant" filled in by each participant.

3 Joint Publications

Here is the list of joint publications resulting from the joint activities of the network. The participants involved are indicated in bold at the end of each bibliographic item, using the following achronyms:

N.	Achronym	Site	C.	Scienties-in-Charge
1	UNIPISA	Dipartimento di Informatica, Pisa	IT	U. Montanari
2	UIA	University of Antwerp	BE	D. Janssens
3	TUB	Technische Universität Berlin	DE	H. Ehrig
4	LaBRI	Laboratorie Bordelais de Recherche en Informatique	FR	M. Bauderon
5	UNIBREMEN	Universität Bremen	DE	H.J. Kreowski
6	RUL	University of Leiden	NL	G. Rozenberg
7	UNIROMA1	Università di Roma La Sapienza	IT	F. Parisi Presicce

3.1 Joint papers presented at Conferences

[**BtH00**] Giorgio Busatto and Pieter Jan 't Hoen. A graph-grammar based approach for the specification of hypermedia application dynamics. In *Workshop on Graph Transformation and Visual Modelling, see [CH00]*, pages 403–409, 2000. **RUL, UNIBREMEN.**

[**GPS00**] M. Große–Rhode, F. Parisi Presicce, and M. Simeoni. Refinements of Graph Transformation Systems via Rule Expressions. In H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors, *Proc. 6th International Workshop on Theory and Application of Graph Transformation (TAGT'98)*, Springer LNCS 1764, pages 368–382, 2000. **TUB, UNIROMA1.**

[**GPST00**] M. Große–Rhode, F. Parisi Presicce, M. Simeoni, and G. Taentzer. Modeling Distributed Systems by Modular Graph Transformation based on Refinement via Rule Expressions. In M. Nagl, A. Schürr, and M. Münch, editors, *Proc. Applications of Graph Transformations with Industrial Relevance (AGTIVE'99), Kerkrade, The Netherlands, September 1999*, Springer LNCS 1779, pages 31–45, 2000. **TUB, UNIROMA1.**

[**FKT00**] I. Fischer, M. Koch, and G. Taentzer. Local Views on Distributed Systems and their Communications. In H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg (eds.), *Theory and Application of Graph Transformations*, volume 1764, pages 164–178. Springer Verlag, 2000. **TUB, UNIROMA1.**

4 Appendix - Scientific Contributions

This section describes the main scientific contributions of the research activity carried on during the fourth year of the GETGRATS project. It is divided in seven subsections, one for each of the Focus Areas into which the research themes of the network are divided (as listed in the Annex I of the contract), followed by a list of references.

4.1 (a) Foundation, Unification, Combination and Comparison

Foundations In [CM00], a common framework for HR and VR context-free graph and hypergraph grammars is presented, based on hypergraph operations defined by quantifier free formulas and also, on the fusion of all vertices with same label. Concerning graph decompositions based on graph operations defining VR grammars, the corresponding complexity measure is called clique-width. The case of infinite graphs raises specific problems. Progress has been made for applications to low degree algorithms for problems specified in Monadic Second-Order logic ([CMR00, CO00, CV00])

Graph drawings The papers [Cou00a, Cou00b] consider the formalization with relational structures of graph drawings, with or without edge crossings. Properties of drawings are formalized with monadic second order logic.

Pullback rewriting Pullback of graphs was introduced as a categorical formalism to describe node rewriting in graphs and hypergraphs. But it may be considered the other way round as a (de)composition mechanism. In this approach, we show ([BC00]) how some types of graphs may be composed in two "orthogonal components", whose pullback is the original graph. Iterating the process, these graphs may be written as a composition of some basic buildings blocks. This result gives an easy description of several types of graphs who look very regular (like grids), although there are not, in any language theoretic sense.

SPO-Approach. In [Gru00a] the typed SPO-approach has been extended by meta-typing facilities. The main result of this contribution is that the presented concept of meta-typing is compatible with the SPO theory and can be implemented into the SPO-based graph grammar environment AGG without spoiling its formal basis. This SPO-based approach to meta-typing is compared with the algorithmic approach to meta-typing followed in PROGRES.

4.2 (b) Classification and Expressive Power

With respect to generative power, C-edNCE grammars – which rewrite nodes – are known to provide the most powerful context-free approach to graph rewriting. Considering the context-free rewriting of hypergraphs instead of graphs, it is shown in [Kle99] that confluent node rewriting allows to specify all hyperedge-rewriting and separated handle-rewriting languages, and even languages beyond that union. Still, there are some normal forms for these C-hNCE grammars which allow to deduce that the generated graph languages can already be specified by C-edNCE grammars [Kle00].

Paper [DK00] can be considered as a case study on (extensions of) context-free collage grammars, showing how various types of celtic knotwork can be generated using this type of grammar, but also indicating some limitations.

In [BtH00] a graph-based hypermedia model is proposed, where hypermedia dynamics is specified through graph transformation. This approach addresses the lack of an appropriate language for specifying transformations of hypermedia networks in known methodologies to hypermedia development.

A model of gene assembly in ciliates is formulated in [EPPR00b] and [EPPR00a], which is based on graph transformation. The model has turned out to be successful in the sense that it has provided a uniform explanation to all experimentally known cases of gene assembly in ciliates (for many different genes in many different species of ciliates). The value of this work for the theory of graph transformation is that it connects graph transformation with one of the most modern and exciting developments in computer science, viz. DNA Computing, and at the same time it leads to novel, well-motivated, sorts of gene transformation.

The investigation of dynamic labelled 2-structures as an algebraic (group based) model of graph transformations has been continued. Dynamic labelled 2-structures also play an important role in graph theory where they are known as switching classes of gain graphs. The investigation concerned the issues of complexity, pancyclicity, size, and membership. As mentioned in the Third Annual Report, the results appear in [Hag99, EHHR00a, EHHR00b]. Work is in progress on some combinatorial problems, using forbidden subgraphs; first results appear in [HH00a]. A complete introduction to the theory of 2-structures has been published in the book [EHR99].

It is well known that context-free graph languages can be obtained from regular tree languages by means of MSO-definable tree-to-graph transductions (where MSO abbreviates Monadic Second-Order Logic). In [EM00a] the MSO-definable tree-to-tree transductions are characterized by macro tree transducers of which the size of the output tree is linear in the size of the input tree. Moreover, it is shown that it is decidable whether a macro tree transduction is MSO-definable.

The following papers were described in one of the previous Annual Reports, and the corresponding items have been updated:

- [DKK00] (was [DKK99]), to appear in Proc. DLT'99
- [HH00b] (was [HH97]), appeared in Discrete Mathematics
- [Hag99] (was [Hag98]), appeared in Fundamenta Informaticae
- [EHHR00a] (was [EHHR98]), appeared in Proc. TAGT'98
- [EHHR00b] (was [EHHR99]), appeared in Information Processing Letters

- [BE00] (was [BE98]), appeared in Journal of Computer and System Sciences
- [EM99] (was [EM98a]), appeared in Information and Computation
- [EM00b] (was [EM98b]), appeared in Proc. TAGT'98

4.3 (c) Analysis and Verification Techniques

Tree transducers are useful as a tool for the transformation of hierarchically decomposable graphs. The properties of graph transformations can then be studied by looking at the properties of the tree transducers used. In [Dre00a], the long version of [Dre00b], the complexity of deciding whether a top-down or bottom-up tree transducer yields exponentially large output trees is studied. It turns out to be LOGSPACE-complete for total top-down tree transducers, but DEXPTIME-complete in general ones, and P-complete in the bottom-up case.

System verification in the broadest sense deals with those semantic properties that can be decided or deduced by analyzing a syntactic description of the system. In [KV00] the semantic properties of redundancy and subsumption are carried over from rule-based systems to high-level replacement systems, which in turn generalize graph and hypergraph grammars. The main results are a characterization of subsumption and a sufficient condition for redundancy, which involves composite productions.

The definition of SOS formats ensuring that bisimilarity on closed terms is a congruence has received much attention in the last two decades. For dealing with open terms, the congruence is usually lifted from closed terms by instantiating the free variables in all possible ways; the only alternatives considered in the literature are Larsen and Xinxin's context systems and Rensink's conditional transition systems. Papers [BdFEMOM00a, BdFEMOM00b] present an approach based on Tile Logic, where closed and open terms are managed uniformly, and study the 'bisimilarity as congruence' property for several tile formats, accomplishing different concepts of open system.

4.4 (d) Abstract Semantics

On the one hand the study of semantics concepts for Graph Transformation Systems has continued in the directions sketched in the previous reports, and on the other hand its relationship to other semantic concepts, such as agent-based systems, has been investigated.

The process semantics of Local Action Systems (LAS) was further developed along the lines set out in [JV99]. It has been shown that a number of simple operations may be defined on dynamic structures (a dynamic structure is the component of a process that describes the local actions), and that their algebraic properties provide a useful tool for reasoning about the relationship between processes and their configurations, which is of crucial importance if one wants to connect static properties of a system with dynamic ones. The potential of the technique has been illustrated by providing an equational proof of the so-called coherence theorem, which indeed expresses a relation between a structural property of processes (being the sequential composition of subprocesses) and their configurations (in particular, their result graphs). In [JV00], the expressivity of the process notion for LAS is strengthened by adding explicitly a temporal order on the local actions, and it has been shown that the behavior of NLC graph rewriting, despite being non-confluent, can be described by these so-called ordered processes. It turns out that there is a trade-off between algebraic properties of the alphabet of local actions and the amount of ordering information needed to obtain a process notion that is adequate in the sense that it allows a characterization the intermediate configurations. Alternative ways of increasing the expressivity of the process semantics are also being investigated; preliminary results indicate that it is possible to generate e.g. Message Sequence Charts (MSCs) in this way.

Reactive systems perform their tasks through interaction with their users or with other systems (as parts of a bigger systems). An essential requirement for modeling such systems is the ability to express this kind of interaction. Classical rule-based approaches like Petri nets and graph transformation are not suited for this purpose because they assume to have complete control about the state and its transformation. Therefore, in [EHL+00], a general framework is proposed which extends a given rule-based approach by a loose semantics where the rules of the system (e.g. graph productions or Petri net transitions) are considered as incomplete descriptions of the transformations to be performed: they still determine the changes to the matched substructure but for the remaining part (the context) unspecified changes are possible representing the application of (unknown) rules from the environment. The framework is applied to graph transformation systems in the double-pushout approach as well as place-transition Petri nets leading, respectively, to the concepts of graph transition and open step.

An abstract categorical theory on modularization of graph transformation systems has been developed by M. Simeoni [Sim00]; it is formalized independently of any particular approach on graph transformation. The instantiation of the abstract theory over a concrete approach follows automatically by providing formal proofs to the generic categorical requirements pointed out within the abstract theory. In particular an instantiation for the Local Action Systems approach has been demonstrated.

The use of GTS semantics for the formalization of agent-based systems has been investigated by P. Knirsch first in [KK00a]. An agent system consists of a set of agents and an environment they can act on. In a graph transformation based framework, the environment of course is a graph. In the case of distributed problem solving it represents the problem, e.g. a map if we consider a shortest paths algorithm, that should be solved by the agents. It is also supposed to be a blackboard for asynchronous communication purposes. In this approach an agent comprises a set of rules and a finite automaton. Following the line of transformation units (e.g. [KK99]) this framework is also independent of a certain graph transformation approach. The rules are the agent's basic operational entities to modify its environment. The finite automaton, here called capacity, restricts the order in which the rules can be applied. Two kinds of semantics and their relationships are then considered. Both are based on the idea of applying parallel rules in order to model concurrency. The first one is an operational one based on the idea of "enabled" derivations. A derivation is enabled if for each agent the derivation starting from an initial environment restricted to those rules of the respective agent's set of rules is accepted by the agent's capacity. The second type of semantics is based on the notion of system states, i.e. configurations: an environment combined with a vector of all agents' control automata states constitutes a configuration of the agent system. From a certain configuration, an arbitrary number of agents may apply rules in parallel to derive a new environment, and the active agents reach new states provided that each agent's automaton admits the transition from the actual state to the new state via the applied rule. This is formally described by a transition system using these configurations as states. Based on this approach, it is shown that essential characteristics of agent systems like reactivity, pro-activity and cooperation can easily be modeled mainly using one formalism, i.e. graph transformation.

4.5 (e) Concurrency Aspects

Graph relabelling systems and distributed algorithms Paper [BM98] deals with the delimitation of the power of local computations on graphs with or without some initial knowledge. For this, three classical problems are focussed: the recognition problem, the election problem and the termination detection problem. In [GMM00] it has been proved previously that properties like planarity cannot be determined using local computations without initial knowledge. Moreover recognizability with the knowledge of the size is significantly more powerful than recognizability without initial knowledge. Indeed, the class of covering-minimal graphs, the class of graphs having an odd number of vertices or the class of graphs having exactly one vertex with the label A cannot be recognized without initial knowledge and can be recognized if we know the size of the graphs.

A characterization of families of graphs has been obtained in [MT00] which enables the local detection of the termination of distributed algorithm.

Finally in [MSZ00] and [MSZ00b], randomized algorithms have been proposed and analyzed to implement local synchronizations in an asynchronous anonymous graph. In this work a measure is defined and studied to evaluate the efficiency of these algorithms.

Distributed Graph Transformation. Distributed algorithms and dynamic distributed data structures are basic aspects of distributed systems which usually are modelled separately by different specification techniques. In the running example of the mutual access to file systems, typical problems are shown in [TE00] which are not handled by separate modelling. This paper discusses how these problems can be solved by integrated specification of distributed algorithms and dynamic data structures by graph transformation. From the component point of view it is even more adequate to use distributed graph transformation for the semantics of distributed system specification.

Distributed graph transformation has been used to specify static as well as dynamic aspects of distributed systems. To support distributed designs by different developers, local views are introduced in [FKT00]. A local view on a distributed system consists of one local system, its import and export interfaces, and connected remote interfaces. The behavior of a local system is specified by a set of graph rules that are applicable only to the local view of the local system. Local systems communicate either synchronously or asynchronously via their import and export interfaces. Asynchronous communication is modeled by sequential application of graph rules, synchronous communication by the amalgamation of graph rules. We compose a distributed system grammar from the rule sets for local systems. The operational semantics of the distributed system is given by distributed transformation sequences.

In [TGM99], [TGM00], [GEMT00a] and [GEMT00b] several different applications of distributed graph transformation to dynamic architecture configuration for distributed systems and view point-oriented software development have been developed.

The problem of specifying dynamic change in distributed systems is addressed in [TGM99] and [TGM00]. Change in distributed systems is related to at least two levels. One is the management of change in local nodes of the distributed system and how such a change is then propagated to those nodes which need to know about the change. The other aspect is changing the structure of the distributed system itself. This implies e.g. to add and/or remove a local node or an entire subsystem to/from the distributed system. In some important application areas such operations must be done during runtime without disturbing the unmodified rest of the distributed computing system. In [TGM00], first an overview of the model of change is given and explained how the two aspects of change interact. It is discussed how distributed graph transformation is used to realize the change model. An example - a ring database - shows how the approach can be applied to a small but non-trivial distributed system. This example shows nicely how the two aspects can be defined in an adequate way.

In [GEMT00a] and [GEMT00b], the integration of multiple perspectives in view point-oriented software development is described by distributed graph transformation. In order to support multiple perspectives in software development one needs a scheme which expresses explicitly all the views held by the various stake-holders like requirement engineers, software architects, clients, users, etc. The ViewPoint framework has been developed in the past as a conceptual model for expressing such a multiple persective steeing in software development projects. In [GEMT00a], this framework is formally described by distributed graph transformation and the applicability of this approach is demonstrated at a non-trivial sample system. In [GEMT00b], the ViewPoint Tool is presented which supports the definition and usage of multiple views based on the above mentioned approach using graph transformation.

Coupled Graph Grammars. In [Gru00], a specification method for distributed data modeling based on coupled graph grammars is presented. This method is able to uniformly describe legal domain configurations used in distributed modeling as well as re-engineering tasks. The specification method supports the development of proper (re)design tools, and the method is tool-supported itself.

Local Action Systems Progress on Local Action Systems (LAS) is reported in Section (d) on abstract semantics.

4.6 (f) Modularity Aspects

If systems are specified by graph transformation, large graphs should be structured in order to be comprehensible. The papers [DHP00a, DHP00b] presents an approach to the rule-based transformation of hierarchically structured (hyper)graphs. In these graphs, distinguished hyperedges contain graphs that can be hierarchical again. This framework extends the well-known double-pushout approach from flat to hierarchical graphs. The paper shows how pushouts and pushout complements of hierarchical graphs can be constructed recursively. Moreover, rules are made more expressive by introducing variables which allow to copy and to remove hierarchical subgraphs in a single rule application.

In [KK99] the notion of a transformation unit is introduced together with its interleaving semantics. It is studied as a means to construct large graph transformation systems from small ones in a structured and systematic way. A transformation unit comprises a set of rules, descriptions of initial and terminal graphs, and a control condition. Moreover, it may import other transformation units for structuring purposes. Its semantics is a binary relation between initial and terminal graphs which is given by interleaving sequences. The introduced framework is independent of a particular graph transformation approach. A thorough study of the theory of transformation units is presented in the thesis [Kus00b].

Putting transformation units together yields transformation modules. A module may have an import interface consisting of formal parameter units and an export interface consisting of the units that are made available to the environment. The formal parameter units can be instantiated by exported units of other modules. In [KK00b] transformation units and transformation modules are generalized in such a way that they can be used for the structuring of arbitrary rule-based systems. Moreover, the interleaving semantics is relaxed which means that the imported units may also operate on local components of actual configurations rather than on the global objects.

In [tHBG99] the concept of an instance level module is introduced to encapsulate objects in an object-oriented model. This concept is based on *packages* from the industry standard UML. This paper sees the general issue of structuring graphs from the perspective of object orientation.

In [GPS00], graph transformation systems are considered as formal models of computational systems, specified by rules which describe the atomic steps of a system. A refinemenet of a graph transformation system is given by associating with each of its rules a composition of rules of a refining system, that has the same visible effect as the original rule. The basic composition operations on graph transformation rules are sequential and parallel composistion, corresponding to temporal and spatial refinement respectively. Syntactically, refinements are represented by rule expressions that describe how the refining rules shall be composed.

For the development of larger systems, it is important that a modeling technique offers a stringent module concept. Each module has to support the encapsulation of data structure as well as functionality also at runtime. Modular graph transformation, presented in [GPST00], supports these features. Modules are built up of specifications where attributed graphs describe the static data structures, whereas the dynamic behavior is modeled by the controlled application of graph rules. Rule expressions are used to formulate the control flow.

Within one module, it is stated that a (weak) preservation of export and import behavior wrt. the local behavior in the module's body in the sense that an interface derivation is subsumed by a local derivation if it can be performed. Modules may use each other, meaning that each import interface has to be connected with an export interface in a way that the import behavior is subsumed by the export behavior.

An important issue in the area of software architecture is the specification of reconfiguration and mobility of systems. Papers [HIM99] and [HIM00] presents an approach for the specification of software architecture styles using hyperedge replacement systems and for their dynamic reconfiguration using constraint solving. A system architecture is represented as a graph where edges are components and nodes are ports of communication. Then, a style is represented as a graph grammar where the instances of the style are the graphs generated by the corresponding grammar. The construction and dynamic evolution of the style are represented as context-free productions and graph rewriting. To model reconfigurations we allow the declaration, creation and matching of new nodes (i.e. ports of communication) and use constraint solving over the productions of the style grammar for achieving synchronization. In this way complex evolutions can be specified in a more expressive and compact form than using π-calculus style languages for mobility.

A similar approach is proposed in [HM00], where graphs, productions and derivations are characterized as certain terms of a simply typed λ-calculus, and hyperedge replacement can be efficiently implemented as β-conversion.

The contents of the papers [HHKK00], [Kus00a], and [DKKK00] have been described in previous interim reports.

4.7 (g) Morphisms, Transformations and Operations

The broad objective of the PhD thesis [Bal00] is to consolidate the foundations of the concurrency theory for DPO graph transformation systems, by extending to this more general setting some fundamental approaches proper of Petri net theory. Following the guidelines traced by Nielsen, Plotkin and Winskel in their seminal work on safe nets, the thesis defines a functorial construction which associates to each graph grammar a nondeterministic occurrence grammar describing its behaviour. The unfolding is then abstracted to an (appropriate kind of) event structure and finally to a finitary prime algebraic domain. The natural notion of deterministic process emerging from the above theory coincides with an earlier proposal in the literature. Building on this work, *concatenable graph processes* are defined, which are shown to provide a description of the truly concurrent behaviour of the grammar equivalent to the classical truly concurrent model of computation based on traces. This result is used to show that the event structure obtained via the unfolding coincides both with the one defined by Corradini et al. via a comma category construction on the category of traces, and with the one proposed by Schied, based on a deterministic variant of the DPO approach. These results, besides confirming the appropriateness of the proposed unfolding construction, unify the various event structure semantics for the DPO approach to graph transformation.

The first part of the thesis lays the conceptual basis for the treatment of grammars by developing an analogous theory for two generalizations of Petri nets in the literature, which reveal a close relationship with graph transformation systems, namely *contextual nets* (also called nets with read, activator or test arcs) and *inhibitor nets* (or nets with inhibitor arcs). In the treatment of contextual nets a basic role is played by *asymmetric event structures*, which generalize prime event structures by allowing a non-symmetric conflict relation. The generalization to inhibitor nets, requires the definition of a new, very general, notion of event structure, called *inhibitor event structure*, to faithfully express the dependency among events arising from the non-monotonicity of the enabling. Inhibitor event structures turns out to be expressive enough to model graph grammar computations and the theory developed for graph grammars can be presented as a lifting of the corresponding theory for inhibitor nets. A refined version of the work on contextual nets appears also in [BCM00a]. The unfolding semantics of inhbitor nets has been presented also in [BBCP00a, BBCP00b]. The unfolding of graph grammars is described also in [BCM00b].

The work on the functorial semantics of multi-algebras and partial algebras already described in the third progress report (see [CG00], which has been accepted for publication in TCS) is continued in [CGK00], where the term graph syntax for multi-algebras is exploited to design an inference calculus for term-graph equational deduction. The soundness of the calculus is proved, but its completeness remains an open problem.

Paper [BS00] extends the algebraic approach of Meseguer and Montanari from ordinary place/transition Petri nets to contextual nets, covering both the collective and the individual token philosophy uniformly along the two interpretations of net behaviors.

Papers [CG99b] and [CG99a], already described in the previous reports, where published during the last year.

References

[Bal00] P. Baldan. *Modelling concurrent computations: from contextual Petri nets to graph grammars.* PhD thesis, Department of Computer Science, University of Pisa, 2000. Available as technical report n. TD-1/00.

[BBCP00a] P. Baldan, N. Busi, A. Corradini, and G.M. Pinna. Domain and event structure semantics for Petri nets with read and inhibitor arcs. Technical Report TR-05-00, Department of Computer Science, University of Pisa, 2000.

[BBCP00b] P. Baldan, N. Busi, A. Corradini, and G.M. Pinna. Functorial concurrent semantics for Petri nets with read and inhibitor arcs. In C. Palamidessi, editor, *CONCUR'00 Conference Proceedings*, volume 1877 of *LNCS*, pages 442–457. Springer Verlag, 2000.

[BC00] M. Bauderon and F. Carrere. Orthogonal graph decompositions. In *GRATRA'2000, see [ET00]*, pages 197–204, 2000.

[BCM00a] P. Baldan, A. Corradini, and U. Montanari. Contextual Petri nets, asymmetric event structures and processes. To appear in *Information and Computation*. (Also Technical Report TR-99-18 of the Department of Computer Science, University of Pisa), 2000.

[BCM00b] P. Baldan, A. Corradini, and U. Montanari. Unfolding of double-pushout graph grammars is a coreflections. In *TAGT'98, see [EEKR00]*, pages 145–163, 2000.

[BdFEMOM00a] R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, and U. Montanari. Bisimilarity congruences for open terms and term graphs via tile logic. In C. Palamidessi, editor, *CONCUR 2000, 11th International Conference on Concurrency Theory*, volume 1877 of *LNCS*, pages 259–274. Springer Verlag, 2000.

[BdFEMOM00b] R. Bruni, D. de Frutos-Escrig, N. Martí-Oliet, and U. Montanari. Tile bisimilarity congruences for open terms and term graphs. Technical Report TR-00-06, Computer Science Department, University of Pisa, 2000.

[BE00] R. Bloem and J. Engelfriet. A comparison of tree transductions defined by monadic second order logic and by attribute grammars. *Journal of Computer and System Sciences*, 61:1–50, 2000.

[BM98] A. Bottreau and Y. Métivier, *Minor searching, normal forms of graph relabelling: two applications based on enumeration by graph relabelling,* In: Foundations of System Specification and Computation Structures (FoSSaCS'98) Lecture Notes in Comput. Sci. **1378** (1998) 110–124.

[BS00] R. Bruni and V. Sassone. Algebraic models for contextual nets. In U. Montanari, J.D.P. Rolim, and E. Welzl, editors, *ICALP2000, 27th Int. Coll. on Automata, Languages and Programming*, volume 1853 of *LNCS*, pages 175–186. Springer Verlag, 2000.

[BtH00] Giorgio Busatto and Pieter Jan 't Hoen. A graph-grammar based approach for the specification of hypermedia application dynamics. In *Workshop on Graph Transformation and Visual Modelling, see [CH00]*, pages 403–409, 2000.

[Cou00a] B. Courcelle: The monadic second-order logic of graphs XII: Planar graphs and planar maps. Theoret. Comput. Sci. 237 (2000) 1-32.

[Cou00b] B. Courcelle: The monadic second-order logic of graphs XIII: Graph drawings with edge crossings.Theoretical Computer Science, 244 (2000) 63-94.

[CG99a] A. Corradini and F. Gadducci. An algebraic presentation of term graphs, via gs-monoidal categories. *Applied Categorical Structures*, 7:299–331, 1999.

[CG99b] A. Corradini and F. Gadducci. Rewriting on cyclic structures: Equivalence between the operational and the categorical description. *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 33:467–493, 1999.

[CG00] A. Corradini and F. Gadducci. A functorial semantics for multi-algebras and partial algebras, with applications to syntax. *Theoretical Computer Science*, 2000. to appear.

[CGK00] A. Corradini, F. Gadducci, and W. Kahl. Term graph syntax for multi-algebras. Technical Report TR-00-04, Dipartimento di Informatica, 2000.

[CH00] A. Corradini and R. Heckel. Graph transformation and visual modelling techniques. In J.D.P. Rolim, A.Z. Broder, A. Corradini, R. Gorrieri, R. Heckel, J. Hromkovic, U. Vaccaro, and J.B. Wells, editors, *ICALP Workshops 2000*, volume 8 of *Proceedings in Informatics*, pages 357–486. Carleton Scientific, 2000.

[CM00] B. Courcelle and J. Makowsky. Operations on relational structures and their compatibility with monadic second-order logic. In *GRATRA 2000, see [ET00]*, pages 206–214, 2000.

[CMR00] B. Courcelle, J. Makowsky and U. Rotics (Technion, Haifa, Israel): Linear time solvable optimization problems on certain structured graph families, Theory of Computer Science (formerly Mathematical Systems Theory) 33 (2000) 125-150.(Final version of [CMR99a], cited in 1999 report)

[CO00] B. Courcelle, S. Olariu: Upper bounds to the clique-width of graphs, Discrete Applied Mathematics 101 (2000) 77-114. (cited as [CO99] in the 1999 report).

[CV00] B. Courcelle, R. Vanicat, Query efficient implementations of long words and linear structured graphs, June 2000, Submitted for publication.

[DHP00a] Frank Drewes, Berthold Hoffmann, and Detlef Plump. Hierarchical graph transformation. In *Proc. Foundations of Software Science and Computation Structures (FOSSACS 2000)*, volume 1784 of *LNCS*, pages 98–113. Springer-Verlag, 2000.

[DHP00b] Frank Drewes, Berthold Hoffmann, and Detlef Plump. Hierarchical graph transformation. Informatik-Bericht 1/2000, Fachbereich Mathematik und Informatik, Universität Bremen, 2000. 33 pages.

[DK00] Frank Drewes and Renate Klempien-Hinrichs. Picking knots from trees. The syntactic structure of celtic knotwork. In *Proc. Diagrams 2000*, Lecture Notes in Artificial Intelligence, 2000. To appear.

[DKK00] Frank Drewes, Renate Klempien-Hinrichs, and Hans-Jörg Kreowski. Table-driven and context-sensitive collage languages. In G. Rozenberg and W. Thomas, editors, *Proc. Developments in Language Theory (DLT'99)*. World Scientific, 2000. To appear.

[DKKK00] Frank Drewes, Peter Knirsch, Hans-Jörg Kreowski, and Sabine Kuske. Graph transformation modules and their composition. In *AGTIVE'99, see [NSM00]*, pages 15–30, 2000.

[Dre00a] Frank Drewes. The complexity of the exponential output size problem for top-down and bottom-up tree transducers. *Information and Computation*, 2000. To appear.

[Dre00b] Frank Drewes. The complexity of the exponential output size problem for top-down tree transducers. In G. Ciobanu and Gh. Phaun, editors, *Proc. Fundamentals of Computation Theory (FCT'99)*, volume 1684 of *Lecture Notes in Computer Science*, pages 234–245, 2000.

[EEKR00] H. Ehrig, G. Engels, H.-J. Kreowski, and G. Rozenberg, editors. *Proceedings of Workshop on Theory and Application of Graph Transformations (TAGT'98), Paderborn, Germany, November 1998*, volume 1764 of *LNCS*. Springer Verlag, 2000.

[EHHR00a] A. Ehrenfeucht, J. Hage, T. Harju, and G. Rozenberg. Complexity issues in switching of graphs. In *TAGT'98, see [EEKR00]*, pages 59–70, 2000.

[EHHR00b] A. Ehrenfeucht, J. Hage, T. Harju, and G. Rozenberg. Pancyclicity in switching classes. *Information Processing Letters*, 73(5–6):153 – 156, 2000.

[EHL+00] H. Ehrig, R. Heckel, M. Llabres, F. Orejas, J. Padberg, and G. Rozenberg. Double-Pullback Graph Transitions: A Rule-Based Framework with Incomplete Information. In *TAGT'98, see [EEKR00]*, pages 85–102, 2000.

[EHR99] A. Ehrenfeucht, T. Harju, and G. Rozenberg. The Theory of 2-Structures: A Framework for Decomposition and Transformation of Graphs. World Scientific, 1999.

[EM99] J. Engelfriet and S. Maneth. Macro tree transducers, attribute grammars, and MSO definable tree translations. *Information and Computation*, 154:34–91, 1999.

[EM00a] J. Engelfriet and S. Maneth. Characterizing and deciding MSO-definability of macro tree transductions. In H.Reichel and S.Tison, editors, *STACS'2000*, volume 1770 of *Lecture Notes in Computer Science*, pages 542–554, Berlin, 2000. Springer Verlag.

[EM00b] J. Engelfriet and S. Maneth. Tree languages generated by context-free graph grammars. In *TAGT'98, see [EEKR00]*, pages 15–29, 2000.

[EPPR00a] A. Ehrenfeucht, I. Petre, D.M. Prescott, and G. Rozenberg. String and graph reduction systems for gene assembly in ciliates. 2000.

[EPPR00b] A. Ehrenfeucht, I. Petre, D.M. Prescott, and G. Rozenberg. Universal and simple operations for gene assembly in ciliates. To appear in Words, Sequences, Languages: Where computer science, biology and linguistics come across, Mitrana, V., Martin-Vide, C. (eds.), 2000.

[ET00] H. Ehrig and G. Taentzer, editors. *Proceedings of Joint APPLIGRAPH/GETGRATS Workshop on Graph Transformation (GRATRA 2000).* Technical Report No. 2000-2, FB Informatik, TU Berlin, April 2000.

[FKT00] I. Fischer, M. Koch, and G. Taentzer. Local Views on Distributed Systems and their Communications. In *TAGT'98, see [EEKR00]*, pages 164–178, 2000.

[GEMT00a] M. Goedicke, B. Endres, T. Meyer, and G. Taentzer. Tool Support for ViewPoint-Oriented Software Development: Towards Integration of Multiple Perspectives by Distributed Graph Transformation. In *AGTIVE'99, see [NSM00]*, pages 369 – 3780, 2000.

[GEMT00b] M. Goedicke, B. Endres, T. Meyer, and G. Taentzer. ViewPoint-Oriented Software Development: Tool Support for Integrating Multiple Perspectives by Distributed Graph Transformation. In H. Ehrig,

G. Engels, H.-J. Kreowski, and G. Rozenberg (eds.), *6th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems, Berlin, Germany*, volume 1785, pages 43 – 47. Springer Verlag, 2000.

[GMM00] E. Godard, Y. Métivier and A. Muscholl. The power of local computations in graphs with initial knowledge. In *TAGT'98, see [EEKR00]*, pages 71–84, 2000.

[GPS00] M. Große–Rhode, F. Parisi Presicce, and M. Simeoni. Refinements of Graph Transformation Systems via Rule Expressions. In *TAGT'98, see [EEKR00]*, pages 368–382, 2000.

[GPST00] M. Große–Rhode, F. Parisi Presicce, M. Simeoni, and G. Taentzer. Modeling Distributed Systems by Modular Graph Transformation based on Refinement via Rule Expressions. In *AGTIVE'99, see [NSM00]*, pages 31–45, 2000.

[Gru00] S. Gruner. A Combined Graph Schema and Graph Grammar approach to Consistency in Distributed Data Modeling. In *AGTIVE'99, see [NSM00]* , 2000.

[Gru00a] S. Gruner. Meta-typing is compatible to the typed SPO approach. In *GRATRA 2000, see [ET00]*, pages 50–58, 2000.

[Hag99] J. Hage. The membership problem for switching classes with skew gains. *Fundamenta Informaticae*, 39(4):375–387, 1999.

[HH00a] J. Hage and T. Harju. A characterization of acyclic switching classes using forbidden subgraphs. Technical Report 5, Leiden University, Department of Computer Science, 2000. Submitted to European J. Combin.

[HH00b] J. Hage and T. Harju. The size of switching classes with skew gains. *Discrete Mathematics*, 215:81 – 92, 2000.

[HHKK00] Reiko Heckel, Berthold Hoffmann, Peter Knirsch, and Sabine Kuske. Simple modules for GRACE. In *TAGT'98, see [EEKR00]*, pages 383–395, 2000.

[HIM99] D. Hirsch, P. Inverardi, and U. Montanari. Modeling software architectures and styles with graph grammars and constraint solving. In *Proc. Working IFIP Conference on Software Architecture*, 1999.

[HIM00] D. Hirsch, P. Inverardi, and U. Montanari. Reconfiguration of software architecture styles with name mobility. In A. Porto and G.-C. Roman, editors, *Coordination 2000*, volume 1906 of *LNCS*, pages 148–163. Springer, 2000.

[HM00] D. Hirsch and U. Montanari. Higher-order hyperedge replacement systems and their transformations. In *GRATRA 2000, see [ET00]*, pages 215–223, 2000.

[JV99] D. Janssens and N. Verlinden. Algebraic properties of LAS processes. Technical report, Universiteit Antwerpen, UIA, 1999.

[JV00] D. Janssens and N. Verlinden. A framework for NLC and ESM: Local Action Systems. In *TAGT'98, see [EEKR00]*, pages 194–214, 2000.

[KK99] Hans-Jörg Kreowski and Sabine Kuske. Graph transformation units with interleaving semantics. *Formal Aspects of Computing*, 11(6):690–723, 1999.

[KK00a] Peter Knirsch and Hans-Jörg Kreowski. A note on modeling agent systems by graph transformation. In *AGTIVE'99, see [NSM00]*, pages 411–417, 2000.

[KK00b] Hans-Jörg Kreowski and Sabine Kuske. Note on approach-independent structuring concepts for rule-based systems. In *GRATRA 2000, see [ET00]*, pages 41–49, 2000.

[Kle99] Renate Klempien-Hinrichs. The generative power of context-free node rewriting in hypergraphs. *Grammars*, 2:211–221, 1999.

[Kle00] Renate Klempien-Hinrichs. Normal forms for context-free node-rewriting hypergraph grammars. In *GRATRA 2000, see [ET00]*, pages 224–231, 2000.

[Kus00a] Sabine Kuske. More about control conditions for transformation units. In *TAGT'98, see [EEKR00]*, pages 323–337, 2000.

[Kus00b] Sabine Kuske. *Transformation Units – A Structuring Principle for Graph Transformation Systems.* PhD thesis, Universität Bremen, 2000.

[KV00] Hans-Jörg Kreowski and Gabriel Valiente. Redundancy and subsumption in high-level replacement systems. In *TAGT'98, see [EEKR00]*, pages 215–227, 2000.

[MSZ00] Y. Métivier, N. Saheb and A. Zemmari, *Randomized rendezvous,* In: Colloquium on mathematics and computer science: Algorithms, trees, combinatorics and probabilities, Trends in mathematics, Birkhauser (2000).

[MSZ00b] Y. Métivier, N. Saheb and A. Zemmari, *On the efficiency of randomized local synchronizations,* submitted, 2000.

[MT00] Y. Métivier and G. Tel, *Termination detection and universal graph reconstruction,* in: 7th International Colloquium on Structural Information and Communication Complexity, Carleton Scientific, 2000, 237-251.

[NSM00] M. Nagl, A. Schürr, and M. Münch, editors. *Proceedings of Workshop on Applications of Graph Transformation with Industrial Relevance (AGTIVE'99), Kerkade, The Netherlands, September 1999*, volume 1779 of *LNCS*, 2000.

[Sim00] M. Simeoni. *A Categorical Approach to Modularization of Graph Transformation Systems using Refinements.* PhD thesis, Dipartimento di Informatica, Università di Roma "La Sapienza", 2000.

[tHBG99] Pieter Jan 't Hoen, Giorgio Busatto, and Luuk Groenewegen. Instance level packages for UML. Technical Report 99-06, Leiden Institute of Advanced Computer Science, Vakgroep Informatica, Leiden, NL, 1999.

[TE00] G. Taentzer and H. Ehrig. Semantics of Distributed System Specifications based on Graph Transformation. In M. Wirsing, M. Gogolla, H.-J. Kreowski, T. Nipkow, W. Reif (eds.): *"Rigorose Entwicklung software-intensiver Systeme"*, techn. report LMU München, Institut f. Informatik, no. 0005, Sept. 2000.

[TGM99] G. Taentzer, M. Goedicke, and T. Meyer. Dynamic Accomodation of Change: Automated Architecture Configuration of Distributed Systems. In *Automated Software Engineering'99, Cocoa Beach, Florida, USA*. IEEE Computer Society, 1999.

[TGM00] G. Taentzer, M. Goedicke, and T. Meyer. Dynamic Change Management by Distributed Graph Transformation: Towards Configurable Distributed Systems. In *TAGT'98, see [EEKR00]*, pages 179–193, 2000.