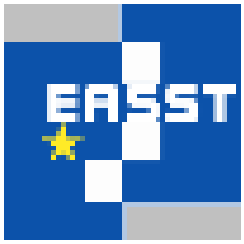
THE EASST NEWSLETTER

Volume 5

March 2003



European Association of Software Science and Technology

EASST **Board:**
Prof. Dr. Herbert Weber (President),
email : herbert.weber@isst.fhg.de
Prof. Dr. Hartmut Ehrig (Vice-President, ETAPS 2000 organizer),
email : ehrig@cs.tu-berlin.de
Prof. Dr. Peter Pepper (Treasurer),
email : pepper@cs.tu-berlin.de
Dr. Julia Padberg (Secretary),
email : padberg@tzi.de
Prof. Dr. Marie-Claude Gaudel (representative in the ETAPS steering committee),
email : Marie-Claude.Gaudel@lri.fr
Prof. Dr. Egidio Astesiano (representative in the ETAPS steering committee),
email : astes@disi.unige.it
Dr. Michel Wermelinger,
email : mw@di.fct.unl.pt
Dr. Tiziana Margaria (Tool Column Editor),
email : tmargaria@metaframe.de

Homepage of EASST**:**
http://www.easst.org

Subscription:
EASST NEWSLETTER is distributed among the members of EASST, the *European Association of Software Science and Technology*. If you are not yet a member of EASST, but you wish to receive the EASST NEWSLETTER, then you are kindly invited to become a member! Note, there are **no** membership fees. The application form can be found on the last page.
Or apply online at http://www.easst.org

Editor:
Dr. Julia Padberg
Universität Bremen On leave from TU Berlin
FB3, Computer Science Department
Postfach 33 04 40
28334 Bremen

E-mail : padberg@cs.tzi.de
URL : http://tfs.cs.tu-berlin.de/~padberg/
Tel : +49-421 218 4202
FAX : +49-421 218 8661

Contents:

Welcome!

Dear EASST Members,

this newsletter comes to you just before the next ETAPS conference in Warsaw, Poland. So, we have included the FASE program and the agenda of the next General Assembly of EASST. The General Assembly takes places on April 10, 2003 from 12:30 to 13:30. Please join the General Assembly and help deciding the future development of EASST.

Again, there is the TOOL COLUMN by T. Margaria with an contribution by Thomas Arts on the requirements for software tools. This contribution is based on five years of industrial experience with formal method tools. It is a topic of highest interest for our community as it deals with the impact of formal techniques to software development in the practice.

Then we have included the article by R. Heckel and M. Chouikha on control synthesis for discrete event systems that has already been accepted for publication by the *Journal of Design & Process science.* This contribution gives a semantic framework based on open Petri nets. We think this is very interesting paper that combines theory and practice in a particularly nice way.

There will be a lot of participants at ETAPS and especially at FASE and I sincerely hope, we will have a fruitful, encouraging and controversial, lively ETAPS. Maybe we meet at some session, in of the discussions, or at the General Assembly.

Yours sincerely,

Julia Padberg
(EASST-Secretary)

Control Synthesis for Discrete Event Systems: A Semantic Framework Based on Open Petri Nets

Reiko Heckel* and Mourad Chouikha**
*Faculty of Computer Science, Electrical Engineering, and Mathematics
University of Paderborn, Paderborn, Germany
reiko@uni-paderborn.de
**EXTESSY AG and Technical University of Braunschweig
Braunschweig, Germany
m.chouikha@extessy.com

Open nets are place-transition Petri nets with interfaces, which support a notion of composition and a corresponding compositional interpretation of the concurrent behaviour of nets. The control synthesis problem of generating a controller for a given plant from an abstract specification of the controller's behaviour can be formulated in terms of open nets by modelling the plant as an open net whose interfaces correspond to the sensors and actuators of the controller and specifying the desired behaviour as a set of processes for this net. Then, the problem consists in synthesising a controller net which, when composed with the net modelling the plant, leads to the specified restriction of the plant's processes.

Based on this observation, which provides an abstraction of the actual synthesis algorithm, we study the problem of generating controllers consisting of several components. In particular, we analyse requirements for the logic used for specifying the controller in order to allow for a compositional, component-wise synthesis.

1 Introduction

Petri nets are widely used in engineering and computer science. They provide an intuitive visual, yet formally based language with a direct representation of concurrency, powerful analysis methods and tool support. Besides visual specifications, also the behaviour of a net can be represented graphically by means of processes, that is, deterministic and acyclic nets that represent concurrent computations. This feature makes it possible to present the concurrent semantics of nets in a semi-formal way, e.g., in order to explain it to domain experts. A well-known drawback, however, of Petri nets is that, in contrast to more "syntactic" methods like process calculi, they do not provide an easy notion of composition which would allow for a compositional interpretation of their behaviour.

Practical methodologies in engineering and computer science take a structured approach, designing systems from smaller subsystems and components, which can be combined and reused. For example, in

workflow specification, one important application area of Petri nets (Aalst, 1998), a common problem is the integration or coupling of workflows of different departments or enterprises (Aalst, 1999). Also technical system are designed from smaller components. Using Petri nets for modelling such systems, it is important that they support a notion of composition and a notion of behaviour which allows to understand and reason about individual components without requiring complete knowledge of their environment.

Most approaches that allow the composition of nets based on some notion of interface consider semantics only globally, or they do not consider a semantics based on processes, but on abstract mathematical models like partial orders, event structures, or traces (Best *et al.*, 1992; Nielsen *et al.*, 1995; Kindler, 1997; Basten, 2000). The approach of open Petri nets is intended to solve the problem of compositionality in nets without loosing the benefits of a visual concurrent semantics. For this purpose, place-transition Petri nets are endowed with an interface of open places which represent the gluing points of the net with some unknown context. Then a corresponding notion of open processes is defined, which behave compositionally in the sense that the open processes of a composed net can be obtained by pairwise merging of open processes of the components nets (Baldan *et al.*, 2001).

Another problem with the application of Petri nets is the gap between Petri nets in theory and Petri net-based languages in practical use. In fact, the latter often combine features like abstract data types or object-oriented concepts, time, read and inhibitor arcs, etc., for which no relevant theory exists. Also there is a lack of theoretical support for domain-specific abstractions and methodologies in application-oriented languages.

The aim of this paper is to shorten this gap by analysing, from a theoretical point of view, a practical application of nets in control engineering and mapping the concepts found in this application to constructions and results in the theory of open nets. In particular, we shall employ open nets to formalise a component-based modelling approach and to provide an abstract formulation of the methodology of control synthesis which allows to generate a controller model for an automation system from a declarative specification. Then, based on the correspondence between theory and application concepts, we derive requirements for the further development of the theory to improve the support of practical problems.

The paper is organised as follows. First, we review the approach to control synthesis and discuss the various places where decomposition of nets is relevant in this context. Then, in Section 3 we present the basic concepts of open Petri nets, their composition operator and their compositional process semantics. Section 4 is devoted to the formulation of the control synthesis problem in terms of open nets. Moreover, in this section, a further development of the approach towards a compositional control synthesis is proposed. Section 5 concludes the paper by a summary and discussion of extensions to the theory that would be needed to provide full support for the control synthesis approach.

2 Model-based Control Synthesis

The control theoretic approach to the synthesis of automation systems is based on a decomposition into a controller and plant which resembles a control loop in control engineering (Fig. 1). In (Chouikha *et al.*, 1998) a Petri net-based methodology is presented which supports this approach in four steps.

1. Modelling of the plant (the subsystem that shall be controlled) by means of a Petri net,

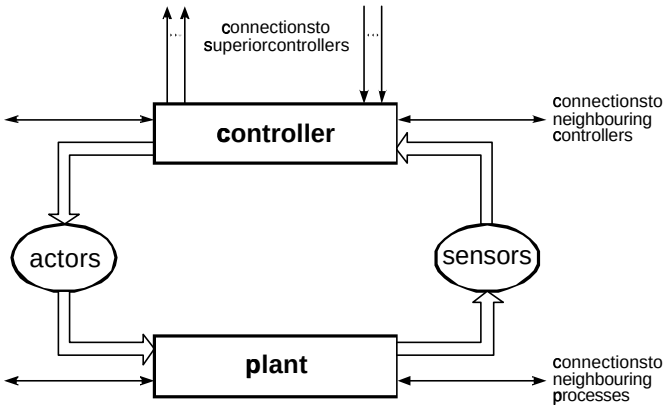


Figure 1: Structure of automation systems

2. Specification of the controller (the component to be developed) using some form of temporal logic
3. Synthesis of a controller model (i.e., a Petri net) from the temporal specification, and
4. Generation of the controller code from the Petri net model

In the following, we shall review the first three steps of this methodology (see also (Lemmer *et al.*, 1995; Chouikha, 1999)) based on a small example, adapted from (Chouikha *et al.*, 1998).

2.1 Modelling of the plant

The model-based methodology for controller synthesis will be explained by means of a simplified stamping process (see Fig. 2). In this process a pusher moves coins from a store into a mould. Then the coins are stamped by a die, and afterwards ejected by a second pusher and blown out of the process by compressed air.

This process is modelled by the Petri net shown in Fig. 3. The net is structured into several components (or objects) representing resources like pushers or the die, or the workpiece to be processed. To support reuse, such objects are stored in libraries, from where they are imported, composed and configured using a design tool (Chouikha *et al.*, 1998).

The plant model consists of several layers, which represent different aspects of the plant. The first layer describes the state changes of the processed objects, the coins. The second layer represents the material flow. Here, the possible locations and the corresponding transportation processes of the workpiece are described. The third layer represents the dynamics of the process resources which influence the state and the location of the workpiece.

The actuators by means of which the resources shall be controlled are modelled as places, shown with bold borders. Sensors, which shall allow to detect the current state of resources or the location and state of a workpiece, are filled grey. The net contains read-arcs, shown as bidirectional arrows, inhibitor arcs, distinguished by a small black circle at the transition end, and two different dependencies between

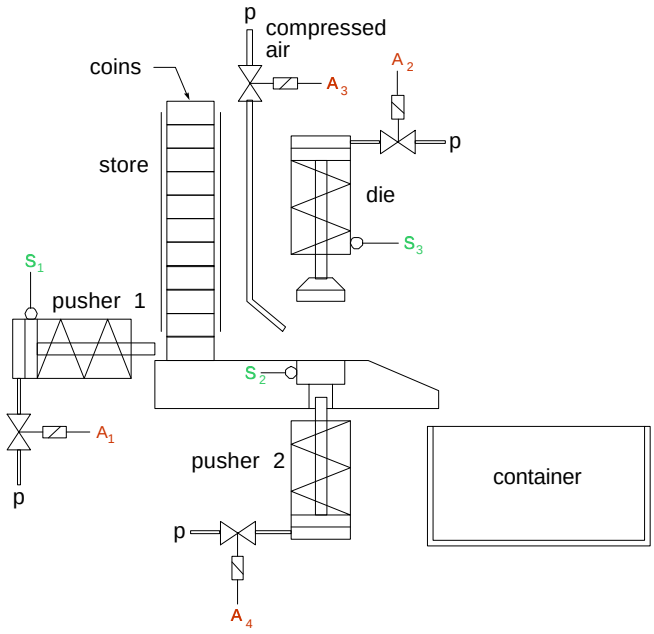


Figure 2: Sample process: stamping the Euro coins

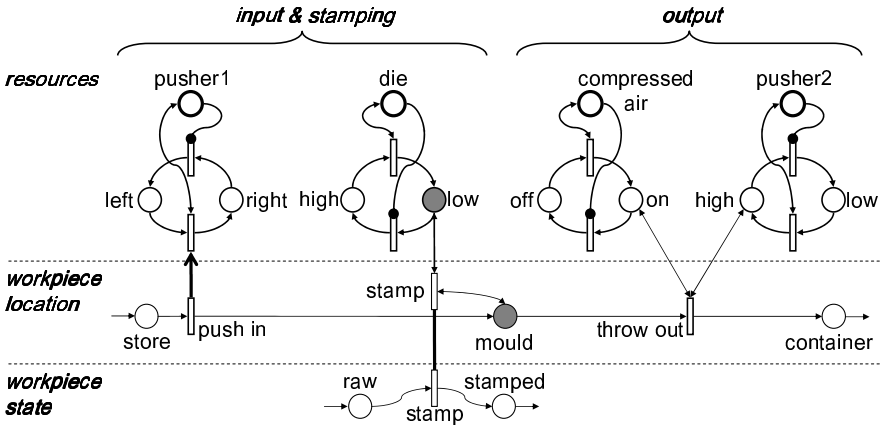


Figure 3: A Petri net model of the plant

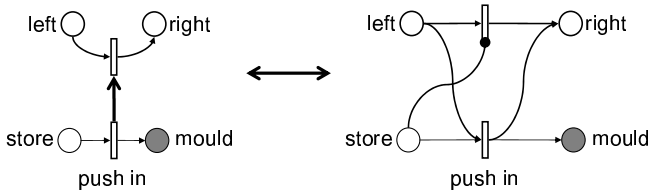


Figure 4: Implied transition: notation and expansion

transitions: The two stamp transitions in the material flow and the workpiece layer are synchronised, that is, they may only fire simultaneously. The transition of the pusher1 component which represents the move from left to right is *implied* by the transition push in. That means, the implied transition has to fire whenever push in fires, but if push in is disabled the implied transition can also fire in isolation. This behaviour is achieved by the pattern shown in Fig. 4.

2.2 Specification of the controller

To specify requirements on the behaviour of the system, a temporal specification language is deployed. As atomic propositions, properties of states (markings) can be expressed by requiring e.g., the presence or absence of tokens in certain places. Using temporal operators safety properties can be expressed, or state properties can be combined to requirements on sequences, which must hold for all executions of the model. Such combinations express lifeness properties (Ober, 1998).

As an example for a safety property for the plant net in Figure 3, to avoid a collision between pusher1 and die we could state that the die must not be in the low position if, simultaneously, the pusher is in the right position.

$$\textbf{always}\ \neg\ (\ \text{die low}\ \wedge\ \text{pusher1 right}\ )$$

As a lifeness property we may require that, whenever there is a workpiece in the store and its state is raw, then sometime later there will be a stamped workpiece in the container, as a formula:

$$\text{store}\ \wedge\ \text{raw}\ \Rightarrow\ \textbf{sometime}\ (\ \text{container}\ \wedge\ \text{stamped}\ )$$

2.3 Control synthesis

The control synthesis problem now consists in generating a controller model, i.e., a second Petri net, which, when composed with the plant net, ensures that the temporal constraints of the specification are fulfilled by all firing sequences of the composed net. This is achieved in two steps. First, a search is performed for firing sequences of the plant net which conform to the specification. Then, a controller net is synthesised which forces the plant net to execute one of these sequences.

A sequence consists of a list of firing steps. Each step represents a number of transitions of the plant which can be concurrently executed, and defines the control outputs necessary to fire these transitions as well as the marking of the sensor places when executing them. The fundamental structure of a control

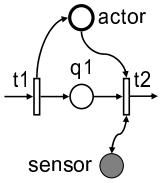


Figure 5: Controller fragment corresponding to a simple step

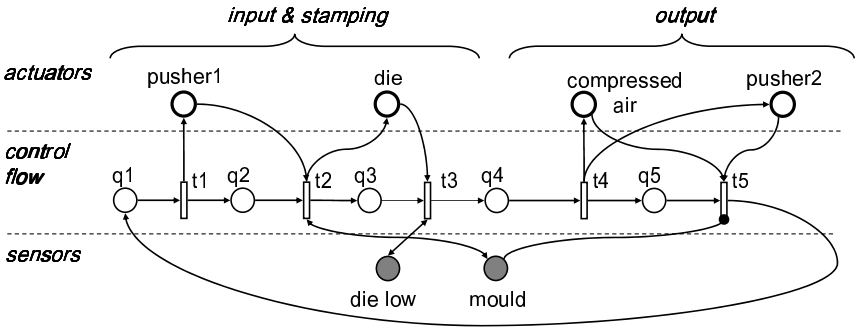


Figure 6: The controller net.

net for a simple step is represented in Fig. 5. Here the transition t1 sets the actuator place enabling one or more transitions in the plant model. As soon as the sensor place indicates the end of the step, transition t2 resets the actuator place.

The controller net of the stamping process is shown in Fig. 6. The net consists of three parts. The central control flow cycle representing the control algorithm is connected to actuator places to specify when controls are set and reset. Connections between sensor places and the transitions of the control cycle model the inputs of the controller.

Summarising, we observe three different ways in which the nets in the control synthesis approach are structured. Plant nets are structured in different layers, like resources, material flow, and workpiece. The resource layer is further structured horizontally into subnets corresponding to different resources. The overall model is composed of the plant and the controller part. Orthogonally, as indicated in the top of Fig. 3 and 6, we may distinguish phases like the input, processing, and output phase.

Next, we shall introduce open nets to support the decomposition of nets at interface places that allow a compositional understanding of their behaviour. In Section 3 below, we will first turn to the decomposition of the (output phase of the) plant net into layers. Then, in Section 4, the separation of plant and controller and the vertical decomposition into different phases shall be modelled.

3 A Compositional Approach to Petri Nets

An *open net* is an ordinary P/T Petri net with a distinguished set of *open places* which represent the interface of the net towards the environment. For example, the open net in Fig. 3 modelling the plant

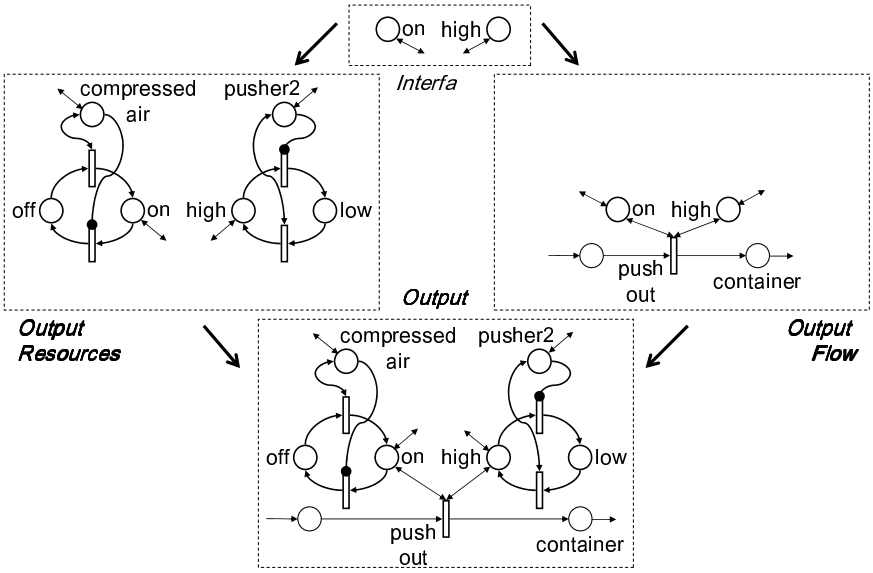


Figure 7: Open nets and their composition.

provides an interface to the controller given by actuator places. From the point of view of the plant net, tokens can freely appear in and disappear from the open places. Concretely, an open place can be either an *input* or an *output* place (or both), meaning that the context can put or remove tokens from that place.

3.1 Open Petri Nets

We assume the standard definition (Reisig, 1985) of a place-transition Petri net N defined as a bipartite multigraph over disjoint sets of places S and transitions T. By $\sigma\ \tau : T \quad S$ we denote the functions assigning to each transition its pre- and post-multiset, where X is the set of finite multisets over a set X. Thus, a net is at the same time a hypergraph with places as vertices and transitions as hyperedges (Meseguer and Montanari, 1990). Sometimes we will find it convinient to use this hypergraph presentation. The powerset over X shall be denoted by $\mathbf{2}^X$.

Given a place $s \quad S$, the pre- and post-set of s are defined by $s \quad t \quad T \ s \quad t$ and $s \quad t \quad T \ s \quad t$.

A net N_0 *is a subnet of* N_1, denoted by $N_0 \quad N_1$ if $T_0 \quad T_1 \ S_0 \quad S_1$ such that source and target functions $\sigma_0 \quad \sigma_{1\ T_0}\ \tau_0 \quad \tau_{1\ T_0}$ are defined by restriction of the corresponding functions of N_1 to the smaller set of transitions. That means, the connections of a given transition in N_0 are exactly the same as in N_1.

Definition 1 (open net) *An* open net *is a pair* $Z \quad (N_Z\ O_Z)$, *where* $N_Z \quad (S_Z\ T_Z\ \sigma_Z\ \tau_Z)$ *is an ordinary P/T Petri net and* $O_Z \quad (O_Z\ O_Z) \quad \mathbf{2}^{S_Z} \quad \mathbf{2}^{S_Z}$ *are the input and output open places of the net.*

Observe that the sets O_Z and O_Z are not necessarily disjoint, hence a place can be both an input and an output open place at the same time.

The open nets for the resources and material flow of the output phase of the stamping process are shown in Fig. 7. Ingoing and outgoing arcs without source or target designate the input and output

Invitation to EASST General Assembly

Dear EASST Members,

following up the EASST General Assembly last year at ETAPS 2002 in Grenoble we will have our next EASST General Assembly during ETAPS 2003 in Warsaw. Enclosed you will find the agenda for this meeting, where especially TOP 4 (Election of new EASST Board) is important. The new EASST Board will have a meeting in Warsaw after the General Assembly to elect the new EASST President, Vice President, Treasurer and Secretary. Moreover, the EASST activities for the future will be discussed during the General Assembly and the Board meeting.

With this announcement we would like to invite you to attend the EASST General Assembly in connection with ETAPS 2003. More information concerning ETAPS you can find on the ETAPS 2003 web site http://www.mimuw.edu.pl/etaps03.

With all best wishes,

Herbert Weber (EASST President) and Hartmut Ehrig (EASST Vice President)

AGENDA EASST GENERAL ASSEMBLY
APRIL 10, 2003, Warsaw

Date: April 10, 2003, 12:30 - 13:30

Place: FASE conference location at ETAPS 03

TOP 1: Report of EASST President

TOP 2: EASST Newsletter Report: J. Padberg

TOP 3: Future EASST Activities Report: T. Margaria

TOP 4: Election of new EASST Board

FMICS 2003

Eighth International Workshop on Formal Methods for Industrial Critical Systems
`http://www.inrialpes.fr/vasy/fmics/workshop-8/`ss

Trondheim, Norway, June 5-7, 2003

Co-located with ERCIM meeting and supported by EASST

INVITED SPEAKERS
Werner Damm
Oldenburg University, Germany
Reiner Hähnle
Chalmers, Sweden
Birger Møller Pedersen
Ericsson and University of Oslo, Norway

LOCAL ORGANISING COMMITTEE
The Norwegian university of science and technology hosts the workshop with the following organising committee:
Finn Arve Aagesen (local org. chair), Frank Lie, Jarle Kotsbak

IMPORTANT DATES
Deadline for submission: March 24, 2003
Accept/Reject notification: May 5, 2003
Final manuscript: May 19, 2003
Workshop: June 5-7, 2003

PROGRAMME COMMITTEE
chairs

Thomas Arts	IT-Univ. in Gothenburg
Wan Fokkink	CWI
Finn Arve Aagesen	Norwegian Univ. of Sc. & Tech.
Gilles Barthe	INRIA Sophia-Antipolis
Armin Biere	ETH Zrich
Jonathan Bowen	South Bank Univ.
Lubos Brim	Masaryk Univ.
Andrew Butterfield	Dublin Univ.
Muffy Calder	Univ. of Glasgow
Dennis Dams	Bell Labs
Leszek Holenderski	Philips
Diego Latella	CNRA/ISTI Pisa
Martin Leucker	Uppsala Univ.
Radu Mateescu	INRIA Rhone-Alpes
Pedro Merino Gmez	Malaga Univ.
Ina Schieferdecker	Fraunhofer FOKUS
Antti Valmari	Tampere Univ. of Technology

SCOPE OF THE WORKSHOP The aim of the FMICS workshops is to provide a forum for researchers who are interested in the development and application of formal methods in industry. In particular, these workshops are intended to bring together scientists who are active in the area of formal methods and interested in exchanging their experiences in the industrial usage of these methods. These workshops also strive to promote research and development for the improvement of formal methods and tools for industrial applications. Topics include, but are not restricted to:

- Tools for the design and development of formal descriptions
- Verification and validation of complex, distributed, real-time systems and embedded systems

- Verification and validation methods that aim at circumventing shortcomings of existing methods in respect to their industrial applicability
- Formal methods based conformance, interoperability and performance testing
- Case studies and project reports on formal methods related projects with industrial participation (e.g. safety critical systems, mobile systems, object-based distributed systems)
- Application of formal methods in standardization and industrial forums

SUBMISSIONS Papers submitted to FMICS 03 must be in English and present original research that is unpublished and not submitted for publication elsewhere. Papers should be between 10 and 16 pages, with a clear abstract and list of keywords. Please use the ENTCS style available from the internet.
Papers should be submitted by e-mail to fmics03@ituniv.se. The format should be either standard PostScript or PDF.

In case of acceptance of a paper at least one author must present the contribution at the workshop, otherwise it will be removed from the list of publications.

The proceedings of the workshop will be published physically by the University of Trondheim and electronically in Elseviers ENTCS electronic notes. High quality papers sent to FMICS are candidate for publication in a special section of the journal for Software Tools for Technology Transfer (STTT).

European Association of Software Science and Technology EASST

Who are we?

EASST is a European non-profit Association that aims at promoting research, development and applications in the area of systematic and rigorous engineering of software and systems.

What are our aims?

Software and Systems Engineering does not receive the public recognition it deserves as one of the most advanced technologies with a great impact on Europe‘s economic and societel prosperity. This is due to a large extent to the low degree of visibility of the community. Especially research is scattered around a rather large number of communities, meetings in different conferences and workshops.

How do you benefit?

When joining us you enter a larger community and you will help to strengthen a new association that is aiming at a better visibility and recognition of your work.
When joining us you will benefit from a cross-fertilisation between a number of subcommittees in joint initiatives, meetings and activities.
When joining us you will have easy access to consolidated information collected from scattered sources.

How to participate?

All information will be made easily accessible by a number of electronic services.

Membership is for free.

Visit our Web-Site: http://www.isst.fhg.de

Statute of EASST

Name

European Association of Software Science and Technology

Location

The Association is located in Berlin/Germany.

Legal Status

The Association is a non-profit organization under German law (»gemeinnütziger eingetragener Verein«).

Purpose and Nature of Activities

The purpose of EASST is to promote the development of science and engineering on software intensive-systems, that play an increasing role in Europe´s way into the information society. It therefore supports education and qualification in software science and engineering, advices decision makers on appropriate measures, and informs the general public on the impact of technology developments.

The Association will

1. organize the exchange of information and spread research results by appropriate means to the community
2. provide help in the coordination of initiatives and projects in the area
3. organize and/or sponsor conferences like ETAPS and other professional meetings
4. coordinate its activities with other professional associations with the goal to give birth to a joint European association in informatics.

Membership

Ordinary membership in the association is open to individuals and legal entities, including other professional associations that support the goals of the EASST.
Associated membership may be obtained by members of other professional societies after proper agreement between them and EASST. Membership applications are requested in written form as determined by the board.

Membership Fee

A membership fee is not collected initially and may be collected later on, only after a decision taken by the membership at large.

Termination of Membership

Membership may be terminated by the member's resignation.
Membership will be terminated if the interest of the member in the membership in EASST vanishes. Indication of lost interest is abstention from decisions taken in EASST in electronic ballots for more than four times consecutively.
Membership will also be terminated if a membership fee due according to decision taken by the membership at large is not paid after its invoicing and after a second request.

Organs and Officers

General Assembly
The membership at large constitutes the general assembly of EASST. The General Assembly elects members of the board of EASST once every two years.
The General Assembly meets at least once a year to receive the annual report of the board including a financial and an activities report. An acceptance vote is expected four weeks after the issue of the report.
The General Assembly votes on the statutes of EASST not later than one year after its constitution, and on further amendments to the statute as well as on the dissolution of the association.

Board

The Board consists of the president, the vice president, the treasurer, the secretary, and four other board members without a particular portfolio.

Voting

Voting takes place in written form as determined by the board. The acceptance/rejection of the statutes, the amendment of the statute and the dissolution of the association require a two thirds majority of the members taking part in the vote.

Termination

In the event of the dissolution of the Association any remaining fund shall be disposed of in a manner determined by the General Assembly so as to support the purposes of EASST.

Application Form

I wish to become a member of EASST.

Please complete the following:

Name, First Name_____

Title_____

Company/University_____

Position_____

Street_____

Postal Code, City_____

Phone_____

Fax_____

E-Mail_____

Date and Signature_____

and return this form as soon as possible to:

EASST c/o
Herbert Weber
Fraunhofer-Institut für Software- und Systemtechnik
Mollstraße 1
D-10178 Berlin

Fax: +49 (0) 30/2 43 06-1 99
E-Mail: herbert.weber@isst.fhg.de