



Doctoral Thesis

Kausale Beschreibung von Systemen

Author(s):

Gisiger, Hans Peter

Publication Date:

1992

Permanent Link:

<https://doi.org/10.3929/ethz-a-000684188> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

DISS. ETH Nr. 9888

Kausale Beschreibung von Systemen

ABHANDLUNG
zur Erlangung des Titels
DOKTOR DER TECHNISCHEN WISSENSCHAFTEN
der
EIDGENÖSSISCHEN TECHNISCHEN HOCHSCHULE ZÜRICH

vorgelegt von
Hans Peter Gisiger
Dipl. El.-Ing. ETH
Dipl. Informatik- Ing. ETH
geboren am 4. März 1957
von Selzach/SO

angenommen auf Antrag von
Prof. Dr. A. Kündig, Referent
Prof. Dr. W. Reisig, Korreferent

Zürich, September 1992

Kurzfassung

In einer umfassenden Systemspezifikation und -Modellierung werden normalerweise die beiden Paradigmen, *Nicht-Determinismus* und *Nebenläufigkeit* eingesetzt. Auf Grund der inhärenten Komplexität verteilter und nebenläufige Systeme sind deren Spezifikationen selten übersichtlich genug, um einfach verstanden zu werden.

Alternativ zur direkten Modellierung nebenläufiger Systeme mit PETRI-Netzen beruht unser Ansatz darauf, Abläufe nebenläufiger Ausführungen zu verwenden. In der PETRI-Netz-Terminologie sind Abläufe nicht-sequentielle Prozesse und unterstützen Spezifikationen mit echter Nebenläufigkeit.

Um auch verschiedene Alternativen in einer Systemspezifikation zu beschreiben, z.B. mit Nicht-Determinismus, werden mehrere, verschiedene Abläufe spezifiziert. Erfüllen diese verschiedenen Abläufe gewisse Konsistenzregeln, bilden sie zusammen eine Prozessfamilie und beschreiben gemeinsam das Verhalten eines ganzen Systems. Durch Faltung der Prozesse einer Prozessfamilie kann das zugehörige Systemnetz einfach generiert werden.

Die verschiedenen Prozesse einer Familie werden in einem sogenannten *Prozessgerüst* organisiert, dessen Entstehungsgeschichte durch ein sog. *Evolutionsgerüst* aufgezeichnet wird.

Der Gewinn dieses Ansatzes drückt sich durch eine bessere Handhabbarkeit nebenläufigen Verhaltens und eine klare Trennung der beiden orthogonalen Konzepte *Nebenläufigkeit* und *Nicht-Determinismus* aus.

Diese Arbeit verfolgt dabei das Ziel, aus einzelnen oder ganzen Familien von Prozessen mit eingeschränkten Strukturen und Beschriftungen sowohl Netzsysteme als auch Code zu einer effizienten und gut eingebetteten Implementation zu generieren.

Abstract

Usually, both paradigms *non-determinism* as well as *concurrency* are used in compound descriptions to specify and model the behaviour of systems. Due to the inherent complexity of distributed and concurrent systems their specifications are rarely transparent enough to be easily understandable.

Our approach to specify systems alternatively to Petri Nets is to use 'runs' of concurrent executions of the target system. In Petri Net terminology runs are non-sequential processes and support specifications with true concurrency.

In order to describe different alternatives in specifications, too, e.g. with non-determinism, we specify several different runs. If such different runs fulfill some consistency rules, they form together a family of processes and constitute the behaviour of a whole system. By folding the processes of a family, the corresponding system net can easily be generated.

All processes of a family will be organized in a so called *process schedule*, and an *evolution schedule* sketches the development history of the processes of the process schedule.

The benefits of this approach is a better maintainability of the concurrent behaviour itself and a clear separation of the two orthogonal concepts, i.e. concurrency and non-determinism.

Our final goal is to be able to transform single processes or complete families of processes with restricted structures and annotations to a net system as well as finally to generate code for an efficient and well-embedded implementation.