WS17/18 (Synchrone Sprachen)

Sinn dieses Seminars ist es, sich mit einem Themengebiet aus dem Bereich der synchronen Sprachen und angrenzenden Themen intensiv und selbständig wissenschaftlich auseinanderzusetzen. Das Thema ist in einem mündlichen Vortrag und einer schriftlichen Ausarbeitung zusammenzufassen. Ein weiterer Sinn dieses Seminars ist es, das Arbeiten in strukturierten zeitlichen Abläufen zu praktizieren, wie es z.B. für Workshops/Tagungen üblich ist. Beide Aspekte sind erfahrungsgemäß eine gute Vorbereitung auf die Anfertigung einer Abschlussarbeit.

Dieses Seminar wird in zwei Varianten angeboten, als Bachelor-Modul und als Master-Modul. Im Vergleich zum Bachelorseminar erwartet das Masterseminar eine größere Einbeziehung von verwandten Arbeiten, und dementsprechend eine umfangreichere Ausarbeitung und Präsentation (siehe unten).

Voraussetzungen

Das Seminar baut auf Inhalten aus der Vorlesung "Synchrone Sprachen" auf. Seminarteilnehmern, welche diese Vorlesung noch nicht gehört haben, wird als Einstieg folgendes Überblickspapier empfohlen, zumindest hiervon die Abschnitte I und II:

Benveniste, A.; Caspi, P.; Edwards, S.A.; Halbwachs, N.; Le Guernic, P.; de Simone, R., "The synchronous languages 12 years later," *Proceedings of the IEEE* , vol.91, no.1, pp.64,83, Jan 2003 (pdf).

Wir empfehlen zudem den Besuch der Blockveranstaltung **Wissenschaftliches Arbeiten für Seminar und Abschlussarbeiten** von Frau Peters.

Dozenten

Reinhard von Hanxleden (rvh@informatik.uni-kiel.de)
Alexander Schulz-Rosengarten (als@informatik.uni-kiel.de)

Themen

Zur Verfügung stehen die folgenden Paper. Sie sind grob in Master- und Bachelor-Themen eingeteilt, anhand von Umfang, Komplexität und benötigtem Vorwissen. Dies soll aber nicht davon abhalten bei überwältigendem Interesse auch als Bachelor ein Master-Thema zu bearbeiten.

Die aktuelle Auswahl ist vorläufig! Die Themen können bereits gewählt werden, aber bis zum Beginn des Wintersemesters wird die Liste ggf. noch erweitert.
Die Paper werden first-come-first-serve vergeben. Ist ein Paper schon an jemanden vergeben vermerken wir das hier.

Viele der Links werden nur aus dem Netz der Uni Kiel heraus funktionieren. Bei Problemen einfach Bescheid sagen.

Bachelor Empfehlung

Yannic Borgfeld: Rémy El Sibaïe and Emmanuel Chailloux. 2016. Synchronous-reactive web programming. In *Proceedings of the 3rd International Workshop on Reactive and Event-Based Languages and Systems* (REBLS 2016). ACM, New York, NY, USA

Julian Pleines: Bourke, T.; Pouzet, M., Zélus: a synchronous language with ODEs, Proceedings of the 16th international conference on Hybrid systems: computation and control, pp. 113-118, 2013.
Dieses Paper ist etwas weniger umfangreich, dafür ist voraussichtlich deutlich mehr Arbeitsaufwand für das Hintergrundverständnis erforderlich, insbesondere für Studierende ohne Vorkenntnisse auf dem Gebiet Synchrone Sprachen.

Dennis Smolka: Nadeem, M.; Biglari-Abhari, M.; Salcic, Z., GALS-JOP: A Java Embedded Processor for GALS Reactive Programs, IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC), 2011.

Nils Sauer: JiaJie Wang; Patha S. Roop; Alain Girault. Energy and timing aware synchronous programming, *2016 International Conference on Embedded Software (EMSOFT)*, Pittsburgh, PA, 2016, pp. 1-10.

Thies Weber: Louis Mandel, Cédric Pasteur, and Marc Pouzet. ReactiveML, Ten Years Later. In *ACM International Conference on Principles and Practice of Declarative Programming (PPDP)*, Siena, Italy, July 2015. Invited paper for PPDP'05 award

Florian Scheurer: Sun, W.-T.; Salcic, Z.; Malik, A., LibGALS: a library for GALS systems design and modeling, Proceedings of the 15th Asia South Pacific Design Automation Conference (ASP-DAC)2010:107-112.

Lars Viertel: Magara, A.; Salvaneschi, G., Ways to react: comparing reactive languages and complex event processing, 1st Workshop on Reactivity, Events and Modularity (REM'13). 2013.

Dennis Pehlke: Gueye, S. M.; De Palma, N.; Rutten, E.; Tchana, A., Coordinating multiple administration loops using discrete control (language: Heptagon/BZR), ACM SIGOPS Operating Systems Review, vol. 47, issue 3, December 2013, pp. 18-25.

Talpin, J.-P; Brandt, J.; Gemünde, M.; Schneider, M.; Shukla, S., Constructive Polychronous Systems, Logical Foundations of Computer Science. vol. 7734, 2013.

Felix von der Heide: Reinhard von Hanxleden and Timothy Bourke and Alain Girault. Real-Time Ticks for Synchronous Programming. In Proc. Foru m on Specification and Design Languages (FDL '17), Verona, Italy, 2017.

Lennart Ideler: Guillaume Baudart and Timothy Bourke and Marc Pouzet. Symbolic Simulation of Dataflow Synchronous Programs with Timers. In Proc. Forum on Specification and Design Languages (FDL '17), Verona, Italy, 2017.

D. Li, Z. Zhai, Z. Pang, V. Vyatkin and C. Liu, Synchronous-reactive Semantic Modelling and Verification for Function Block Networks, in *IEEE Transactions on Industrial Informatics*

Master Empfehlung

Srinivas Pinisetty, Partha S. Roop, Steven Smyth, Stavros Tripakis and Reinhard von Hanxleden. Runtime enforcement of reactive systems using synchronous enforcers. In *CoRR*, vol. abs/1612.05030, 2016.

Lewe Andersen: Yip, E.; Kuo, M.; Roop. P. S., Broman, D., Relaxing the Synchronous Approach for Mixed-Criticality Systems, Proceedings of the 20th IEEE Real-Time and Embedded Technology and Application Symposium (RTAS), Berlin, Germany, April 15-17, 2014.

Niklas Rentz: Mendler, M, Roop, PS & Bodin, B 2016, A Novel WCET semantics of Synchronous Programs. In Formal Modeling and Analysis of Timed Systems: 14th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS 2016). Lecture Notes in Computer Science (LNCS), vol. 9884, Springer International Publishing, pp. 195-210. DOI: 10.1007/978-3-319-44878-7_12

Philip Eumann: Timothy Bourke, Lélio Brun, Pierre-Evariste Dagand, Xavier Leroy, Marc Pouzet, Lionel Rieg. A formally verified compiler for Lustre . In PLDI 2017, Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation Pages 586-60

Sören Domrös: Gesell, M.; Schneider, K., Modular verification of synchronous programs, Application of Concurrency to System Design (ACSD), 2013 13th International Conference on, pp. 70, 79, 8-10 Juli 2013

Lena Grimm: Elisabetta De Maria, Alexandre Muzy, Daniel Gaffé, Annie Ressouche, Franck Grammont. Verification of Temporal Properties of Neuronal Archetypes Using Synchronous Models. [Research Report] RR-8937, UCA, Inria; UCA, I3S; UCA, LEAT; UCA, LJAD. 2016, pp.21.

Bourke T., Colaço JL., Pagano B., Pasteur C., Pouzet M. (2015) A Synchronous-Based Code Generator for Explicit Hybrid Systems Languages. In: Franke B. (eds) Compiler Construction. CC 2015. Lecture Notes in Computer Science, vol 9031. Springer, Berlin, Heidelberg

Guillaume Baudart, Timothy Bourke, and Marc Pouzet. Soundness of the Quasi-Synchronous Abstraction. In *International Conference on Formal Methods in Computer-Aided Design (FMCAD)*, Mountain View, California, USA, October, 3-6 2016

Brandt, J.; Schneider, K.; Passive code in synchronous programs, ACM Transactions on Embedded Computing Systems (TECS), Special Section, vol. 13 issue 2s, Jan2014, article No. 67.

Andreas Stange: Attar, P.; Boussinot, F.; Mandel, L.; Susini, J.-F., Proposal for a Dynamic Synchronous Language, INRIA, Research Proposal, 2011.

Gamatié, A.; Gonnord, L., Static analysis of synchronous programs in signal for efficient design of multi-clocked embedded systems, Conference on Languages, Compilers and Tools for Embedded Systems (ACM SIGPLAN/SIGBED), LCTES 2011:71-80.

Termine

Datum	Meilenstein
Di, 24.10.	Ende der Frist für die Themenauswahl (per Email)
Di, 24.10., 14:30 Uhr	Vorbesprechung/Kick-Off, Latex/Git Kurzeinführung, CAP 4, R. 1115
Di, 14.11., 8:00 Uhr	Abgabe Ausarbeitungsgerüst (Abstract, Einleitung, Gliederung, Stichworte zum Inhalt der Kapitel, Bibliographie)
Mi, 15.11 & Fr, 17.11	Individualtermine, CAP 4, R. 1113
Di, 12.12., 8:00 Uhr	Abgabe der Erstversion der vollständigen Ausarbeitung
Mi, 13.12 & Fr, 15.12	Individualtermine, CAP 4, R. 1113
Di, 09.01., 8:00 Uhr	Abgabe der Review-Version der Ausarbeitung
anschließend	Zuordnung Ausarbeitungen/Reviewer (per Email)
Di, 16.01., 8:00 Uhr	Abgabe der Reviews
tba	evtl. Vortrag zur Gestaltung einer guten Präsentation
Di, 23.01., 8:00 Uhr	Abgabe der Vortragsfolien und Handoutfolien (siehe Hinweise unten)
Mi, 24.01 & Fr, 26.01	Individualtermine, CAP 4, R. 111**3**
Do, 01.02., 8:00 Uhr	Abgabe der Endversionen der Vortragsfolien, Handouts und Ausarbeitungen, Anschließend Druck der Proceedings (inkl. Ausarbeitungen und Handoutfolien)

Mo, 05.02. (+ Di, 06.02.)	Ganztägiges Blockseminar mit Vorträgen

Agenda des Blockseminars

Die Vorstellung des jeweils bearbeiteten Themas wird im Rahmen eines Blockseminars verteilt auf zwei Tage stattfinden. Die Teilnahme an beiden Seminartagen ist Pflicht.

Das erste Blockseminar findet am Montag, den **5. Februar 2018** im Ostseejugenddorf Falckenstein, Falkenhorst 6, 24159 Kiel-Friedrichsort statt, und zwar im Raum "Fördeblick". Bitte wetterfeste Kleidung und passendes Schuhwerk für einen Spaziergang in der Mittagspause mitbringen!

Zeit	Tagesordnungspunkt	
8:20	*Begrüßung*	
8:30	LibGALS: A library for GALS systems design and modeling	Florian Scheurer
9:00	Java Embedded Processors specialized for GALS Programs	Dennis Smolka
9:30	Energie- und zeitabhängige synchrone Programmierung	Nils Sauer
10:00	*Kaffeepause*	
10:30	A Formally Verified Compiler for Lustre	Philip Eumann
11:15	Modular Verification of Synchronous Programs	Sören Domrös
12:00	*Mittagessen*	
14:00	Temporal Properties for Neuronal Archetypes using Synchronous Models	Lena Grimm
14:45	The Dynamic Synchronous Language DSL	Andreas Stange
15:30	*Kaffeepause*	
16:00	WCET Semantics for Synchronous Programs	Niklas Rentz
16:45	Relaxing the Synchronous Approach for Mixed-Criticality Systems	Lewe Andersen
17:30	*Schlusswort*	

Das zweite Blockseminar findet am Dienstag, den **6. Februar 2018** in der Uni in CAP4 R.1115 (RTSYS Labor) statt.

Zeit	Tagesordnungspunkt	
10:10	*Begrüßung*	
10:15	Eine Einordnung der Sprache Zélus	Julian Pleines
10:45	Symbolic Simulation of Dataflow Synchronous Programs with Timers	Lennart Ideler
11:15	Synchrone Programme für Echtzeitsysteme mittels dynamischer Tickfunktionen	Felix von der Heide
11:45	Synchronous-Reactive Web Programming	Yannic Borgfeld
12:15	*Mittagessen*	
13:00	ReactiveML, 10 Years Later - Ein Überblick	Thies Weber
13:30	Reactive Languages and Complex Event Processing	Lars Viertel
14:00	Koordination mehrerer Autonomer Manager	Dennis Pehlke
14:30	*Schlusswort*	

Ausarbeitung, Vortrag, Review

Das Seminar beinhaltet die Erstellung einer Ausarbeitung, eines Vortrags, und zweier Reviews.

Die **Ausarbeitung** soll eine Übersicht über das behandelte Themengebiet darstellen. Sie sollte so verfasst sein, dass sie von einen fortgeschrittenen Bachelor-Informatikstudenten gut verstanden werden kann. Die Ausarbeitung soll 6 (Master) bzw. 4 (Bachelor) Seiten umfassen, nicht mehr und nicht weniger, und den ACM LaTeX-Style verwenden. Für mögliche Vorlagen zu den Ausarbeitungen siehe die Proceedings der früheren Seminare (Achtung, viele dieser Seminare sind gemischte Bachelor-/Masterveranstaltungen gewesen, bitte orientieren Sie sich an den Masterausarbeitungen (Bachlorausarbeitungen 4seitig, Master 6seitig)). Auch empfehlenswert ist ein Blick in die Hinweise für die Anfertigung einer Abschlussarbeit.

Der **Vortrag** soll 40 Minuten (Master) bzw. 25 Minuten (Bachelor) lang sein. Das Vortragsprogramm wird etwas zusätzliche Zeit für Fragen (5 min) einplanen. Zu dem Vortrag sollen Folien erstellt werden. Die Vortragsfolien sollten Seitennummern enthalten. Sollte das Thema auch eine konkrete Implementierung behandeln, ist eine entsprechende kurze Tool-Demo im Rahmen des Vortrages sinnvoll. Die Arbeitsgruppe bietet jedem/r Vortragenden an, eine Videoaufnahme des Vortrags zu erstellen und dem/r Vortragenden anschließend zur Verfügung zu stellen.

Ein **Review** einer Ausarbeitung besteht aus folgenden zwei Komponenten:

1. Generellen Anmerkungen (was gefällt Ihnen/gefällt Ihnen nicht, zu Inhalt, Gliederung und Lesbarkeit) sowie generelle Verbesserungsvorschläge etc. Mindestens eine halbe Seite, abgegeben als PDF-Datei.
2. Detaillierteren Korrekturen als elektronisch annotierte PDF-Version der Review-Version der Ausarbeitung. Es empfiehlt sich, hier nur mit dem Adobe Reader zu arbeiten, da Annotationen verschiedener PDF-Viewer oft inkompatibel sind.

Ein eingescannter, handschriftlich annotierter Ausdruck der Ausarbeitung (generelle Anmerkungen sind auch hier erforderlich!) ist notfalls auch ok, wenn gut lesbar, sollte aber vermieden werden. Die Zuordnung Paper/Reviewer geschieht kurzfristig nach dem Abgabetermin für die Review-Versionen der Ausarbeitungen, basierend auf den dann abgegebenen Ausarbeitungen.

Beispiele zur Ausarbeitung und zu Vortragsfolien finden sich in den Proceedings früherer Seminare.

Jede(r) Seminarteilnehmer(in) erhält die Proceedings des laufenden Seminars.

Namenskonventionen

Auch wenn das Einchecken von generierten Binärdateien generell eher vermieden werden sollte, sind für dieses Seminar auch die folgenden pdfs einzuchecken, um unnötige Compilierungsschwierigkeiten bei Dozenten und Reviewern zu vermeiden. Grafiken sollten in einem Unterordner (z.B. "images") abgelegt werden. Grafiken sollten weiterhin möglichst skalierbare Verktorgrafiken sein, die als PDF eingebunden werden können. Nicht einzuchecken sind temporäre Dateien (.aux etc.).

Die Namen für die Dateien, die im Git abzulegen sind, sollen wie folgt (gleichartig) aufgebaut sein. **Bitte halten Sie sich von Anfang an an diese Namenskonventionen.** Das vermeidet unnötige Sucherei, bewahrt uns vor späteren Schwierigkeiten mit automatischen Skripten und macht umständliches Umbenennen überflüssig.

- Ausarbeitung: *<login>/sem17ws-<login>.[tex/pdf]*
- Vortragsfolien: *<login>/sem17ws-<login>-talk.[tex/pdf]*
- Handoutfolien - ohne Animationen, für Ausdrucke und die Proceedings: *<login>/sem17ws-<login>-handout.[tex/pdf]*
- Review (generelle Anmerkungen): <login review-Empfänger>/sem17ws-<login review-Empfänger>-reviewnotes-<login Reviewer>.[pdf]
- Review (annotiertes PDF): <login review-Empfänger>/sem17ws-<login review-Empfänger>-review-<login Reviewer>.[pdf] (Beispiel also: Alexander (als) reviewt die Ausarbeitung von Steven (ssm) und checkt das Review-PDF mit dem Namen sem17ws-ssm-review-als.pdf im Ordner ssm des Seminarrepositorys ein.

Anmerkung: Die Handoutfolien unterscheiden sich von den Vortragsfolien dadurch, dass die Handoutfolien keine Animationen für die Präsentation am Beamer enthalten. Beim Arbeiten mit der latex-beamer Klasse können Handoutfolien durch das Hinzufügen eines optionalen Argumentes bei der Deklaration der Dokumentenklasse generiert werden ("\documentclass[trans]{beamer}").

Benotung

Das Seminar ist benotet. Die Endnote basiert auf den einzelnen Meilensteinen (Versionen der Ausarbeitung, Reviews, Folien, Vortrag). Es werden jeweils die Qualität sowie die Rechtzeitigkeit (siehe Terminplanung) bewertet. Das Nicht-Einhalten von Terminen kann zum Nicht-Bestehen des Seminars führen.

Technisches

- Reichlich Dokumentation zum *Git* Source Code Management System findet man unter http://www.git-scm.com/.
- Für den Zugriff auf das Repository müssen wir Sie in unserem Bitbucket dafür freischalten. Das sollten wir anhand der Teilnehmerliste prima tun können. Falls Sie keinen Zugriff auf das Repository bekommen, schreiben Sie uns eine E-Mail.
- Git-Repository auschecken: git clone ssh://git@git.rtsys.informatik.uni-kiel.de:7999/sem/17ws-synch.git
    - Um die Erstellung der Proceedings zu erleichtern, richten Sie sich bitte nach den oben beschriebenen Namenskonventionen.
- ACM Style in deutscher oder englischer Version. Im Git Repository befinden sich im Unterverzeichnis *template/* eine Reihe von Dateien, welche Sie als Vorlage verwenden sollen (siehe *README.txt*).
- Wir benutzen pdflatex (erstellt PDF Dateien) und nicht direkt latex (erstellt DVI Dateien)
    - sind im Prinzip gleich zu benutzen
    - Hauptunterschied ist die Einbindung von Grafiken. In pdflatex siehe z.B. http://latex.mschroeder.net/#grafiken (Es sollte immer eine komplette figure Umgebung mit caption, label und Referenz im Text benutzt werden!)
    - Von der Kommandozeile aus kann ein pdf mit "rubber -d sem17ws-<login>" erstellt werden (rubber ruft automatisch pdflatex und bibtex auf).
- Bibliographie: Die Bibliographielemente werden in eine eigene *.bib Datei ausgelagert. Manuell wird dann einmal *pdflatex dokument.tex* aufgerufen. Dies erzeugt eine dokument.aux Datei. Darauf wird *bibtex dokument.aux* aufgerufen und dann nochmal zweimal *pdflatex dokument.tex*. Erst dann sind die Bibliographieelemente richtig im pdf-file.

Weiterführende Hinweise / Links

Generell ist es empfehlenswert, aus dem Uni-Netz heraus nach verwandten Publikationen zu suchen, da man hier Zugriff auf einige Online-Blbliotheken bekommt. Zur Suche empfehlen sich folgende Suchmaschinen und Seiten:

- Google Scholar: http://scholar.google.de/
- CiteSeer: http://citeseer.ist.psu.edu/
- IEEE-Xplore: http://ieeexplore.ieee.org/Xplore/dynhome.jsp
    - Kostenloser Download nur aus Rechnern im Uninetz möglich
- ACM Digital Library: http://portal.acm.org/dl.cfm
- Universitätsbibliothek Digitale Medien: http://www.uni-kiel.de/ub/emedien/index.html