VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kansliets noteringar Kod	Dnr
2013-45635-105143-33	2013-5218

2013
Project Grant Junior Researchers

Area of science
Natural and Engineering Sciences

Announced grants
Research grants NT April 11, 2013

Total amount for which applied (kSEK)

2014	2015	2016	2017	2018
1093	1126	945	971	

APPLICANT

Name(Last name, First name)
Weyns, Daniel

Date of birth
580930-1533

Gender
Male

Email address
danny.weyns@lnu.se

Academic title
Associate professor

Position
Docent

Phone
+32474208251

Doctoral degree awarded (yyyy-mm-dd)
2006-10-11

WORKING ADDRESS

University/corresponding, Department, Section/Unit, Address, etc.
Linnéuniversitetet
Institutionen för datavetenskap, fysik och matematik, Växjö
Datavetenskap
Hus B - 3009
35195 Växjö, Sweden

ADMINISTRATING ORGANISATION

Administrating Organisation
LINNÉUNIVERSITETET

DESCRIPTIVE DATA

Project title, Swedish (max 200 char)
Garantier i distribuerade själv-anpassande mjukvarusystem

Project title, English (max 200 char)
Assurances in Decentralized Self-Adaptive Software Systems

Abstract (max 1500 char)
Engineering the upcoming generation of software systems and guaranteeing their required qualities (performance, robustness, etc.) is complex due to uncertainties resulting from incomplete knowledge at design time, such as new user needs and faults that are difficult to predict. These challenges have motivated the use of self-adaptation, which is typically realized with a feedback loop that enables a system to adapt itself to internal changes and dynamics in the environment to achieve particular quality objectives. Despite substantial achievements in the field, a key challenge in is to provide assurances for decentralized self-adaptive systems i.e., systems that are controlled by multiple feedback loops that act locally. To tackle this challenge, we propose an approach that combines formal modelling and verification with online learning. To design feedback loops, we employ principles for control theory, which provide a mathematical basis for analysing key properties, incl. stability and transient behaviour. Online learning enables the system to acquire knowledge about design time uncertainties during operation. We validate the research in case studies on smart homes and decentralized supply chains. If successful, the project will contribute to a fundamental understanding of engineering self-adaptation in complex software systems, add to significant improvements of qualities of next generation software systems for industry and society, and provide a basis for future research.

Kod
2013-45635-105143-33

Name of Applicant
Weyns, Daniel

Date of birth
580930-1533

Abstract language
English

Keywords
self-adaptation, quality properties, formal methods, control theory,

Research areas
*Nat-Tek generellt

Review panel
NT-2

Classification codes (SCB) in order of priority
10205

Aspects

Application is also submitted to

similar to:

identical to:

ANIMAL STUDIES

Animal studies
No animal experiments

OTHER CO-WORKER

Name(Last name, First name)
Maggio, Martina

University/corresponding, Department, Section/Unit, Addressetc.
Lunds universitet
Department of Automatic Control

Date of birth
841128

Gender
Female

Academic title
PhD

Doctoral degree awarded (yyyy-mm-dd)
2011-12-31

Name(Last name, First name)
,

University/corresponding, Department, Section/Unit, Addressetc.

Date of birth

Gender

Academic title

Doctoral degree awarded (yyyy-mm-dd)

Name(Last name, First name)
,

University/corresponding, Department, Section/Unit, Addressetc.

Date of birth

Gender

Academic title

Doctoral degree awarded (yyyy-mm-dd)

Name(Last name, First name)
,

University/corresponding, Department, Section/Unit, Addressetc.

Date of birth

Gender

Academic title

Doctoral degree awarded (yyyy-mm-dd)

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2013-45635-105143-33

Name of Applicant
Weyns, Daniel

Date of birth
580930-1533

ENCLOSED APPENDICES

A, B, C, N, S

APPLIED FUNDING: THIS APPLICATION

Funding period (planned start and end date)
2014-01-01 -- 2017-12-31

Staff/ salaries (kSEK)

Main applicant	% of full time in the project	2014	2015	2016	2017	2018
Danny Weyns	30	265	273	282	290	
Other staff						
Martina Magio	20	130	134			
PhD student	80	319	329	339	349	
	Total, salaries (kSEK):	714	736	621	639	

Other projectrelated costs (kSek)	2014	2015	2016	2017	2018
Indirekt costs	292	301	254	261	
Costs for rooms	57	59	50	51	
Travel costs	30	30	20	20	
Total, other costs (kSEK):	379	390	324	332	

Total amount for which applied (kSEK)

2014	2015	2016	2017	2018
1093	1126	945	971	

ALL FUNDING

Other VR-projects (granted and applied) by the applicant and co-workers, if applic. (kSEK)

Funded 2013	Funded 2014	Applied 2014
		3600

Project title	Applicant
Power and temperature control for large-scale computing infrastructures	Martina Maggio

Funds received by the applicant from other funding sources, incl ALF-grant (kSEK)

Funding source	Total	Proj.period	Applied 2014
EU Marie Curie	900	2012-2015	

Project title	Applicant
A Foundation for Engineering Decentralized Self-Adaptive Systems	Daniel Weyns

Kod
2013-45635-105143-33

Name of Applicant
Weyns, Daniel

Date of birth
580930-1533

POPULAR SCIENCE DESCRIPTION

Popularscience heading and description (max 4500 char)

Mjukvarusystem integreras alltmer med varandra, t.ex. affärssystem och sensor-nätverk i smarta hus. Konstruktion av sådana system där specifika kvalitetskrav så som prestanda och tillförlitlighet garanteras är en utmaning, bl. a. beroende på osäker och ofullständig information när designbeslut fattas. Exempel på sådan information är krav, vilka delsystem och andra resurser vars tillgänglighet varierar samt fel som är svåra att förutse. Dessa utmaningar motiverar behovet av själv-anpassande system. Ett sådant system består av ett delsystem som m h a återkoppling kontrolleras och anpassas av ett kontrollsystem. Detta för att uppnå och garantera specifika mål. Trots avsevärda framsteg inom området återstår ett stort antal utmaningar. En nyckelutmaning är garantier för kvalitetskrav. I ett sådant system är mjukvaran distribuerad på ett antal noder och kontrolleras därför av flera kontrollsystem som i stor utsträckning agerar lokalt.

Vi tar oss an den här utmaningen och föreslår en ny och unik ansats som kombinerar formella modeller och formell verifikation med dynamisk maskin-inlärning. Ett formellt ramverk skapar förutsättningar för automatisk verifikation av egenskaper hos själv-anpassade system redan under designfasen. För design och konstruktion av återkopplingsmekanismer används klassisk reglerteori. Detta ger en matematisk grund som är nödvändig för verifiera nyckel egenskaper is systemet så som stabilitet och beteende vid transienter. Dynamisk maskin inlärning skapar förutsättningar för själv-anpassande system att skaffa sig kunskap avseende osäker eller ofullständig design information under drift. Projektet delas in i fyra huvudsakliga uppgifter. Uppgift I studerar formell modellering av system och deras omgivningar. Modellerna specificerar kvalitetsegenskaper i kvantitativa eller stokastiska termer. Modellerna översätts till tidsdiskreta dynamiska system för att göra det möjligt att konstruera ett kontrollsystem. I uppgift II studeras konstruktion och analys lokala kontrollmekanismer och systemövergripande kontroll arkitekturer. Vi kommer att studera konfigurerbara och adaptive kontrollstrategier och kombinera dessa med maskin-inlärning. Uppgift III studerar kontrollmekanismer och mer specifikt hur de kan transformeras till mjukvara och integreras i mjukvarusystemets design för att skapa förutsättningar för den önskade verifieringen av kvalitetsegenskaper. Avslutningsvis, uppgift IVs huvudsakliga ansvar är att arbeta fram konkreta implementationer av kontrollsystemen.

Vi kommer att använda två projekt för att styra och validera forskningsresultat. Ett projekt mot ?smarta hem för äldre? och ett mot ?decentraliserade leverantörskedjor?. Projektet med smarta hem avser studera själv-konfiguration och anpassning baserad på förändringar i systemets omgivning. Själv-konfiguration gör det möjligt för ett system att upptäcka och inkludera nya tjänster baserat på exempelvis individers preferenser och tillstånd. Exempelvis kan ett system aktivera tjänster som informerar omsorgspersonal automatiskt när en vårdtagare vakar och går på toaletten under natten. Kunskap om ett systems omgivning kan användas för att styra tjänster och information som finns tillgängliga i ett system, t.ex. ett system för omsorgspersonal som konfigureras specifikt för den vårdtagare som skall besökas. Strategier för dynamisk maskin-inlärning måste arbetas fram för att ge stöd till denna typ av omgivningsstyrd konfiguration. I fallet med en decentraliserad leverantörskedja kommer vi att studera specifika anpassningar för optimering, robusthet och öppenhet. De varierande behoven i en leverantörskedja ställer krav på dynamisk optimering av dess prestanda, exempelvis genom att förändra policys för lagerhållning. Förändringar av partners kräver att systemet kan hantera om en partner försvinner på ett robust sätt samtidigt som det är öppet för nya partners i leverantörskedjan. Vi avser även här ta fram lämpliga inlärningsstrategier för att dynamisk skapa nya policys och därmed hantera denna typ av osäkerhet.

Projektet kommer att genomföras under fyra år av en grupp med en senior forskare, en postdok och en doktorand. Ett framgångsrikt projekt bidrar till teoribildning kring hur man konstruerar själv-anpassande decentraliserade komplexa system enligt ingenjörsprinciper. Därmed öka kvalitén på nästa generations mjukvara och utgöra en grund för framtida forskning.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
þÿ2013-4 63 -10 143-
Name of applicant
Weyns, Daniel
Date of birth
þÿ 80930-1 33

Title of research programme
Assurances in Decentralized Self-Adaptive Software Systems

Appendix A

Research programme

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Assurances in Decentralized Self-Adaptive Software Systems

1. Purpose and Aims

The upcoming generation of software systems will increasingly consist of loosely coupled interacting subsystems. Examples are systems for business collaborations and networked smart homes. Engineering these systems and guaranteeing the required qualities (performance, robustness, etc.) during system operation is complex due to uncertainties resulting from incomplete knowledge at design time. Among the uncertainties are new user needs, subsystems that come and go at will, dynamically changing availability of resources, and faults that are difficult to predict. The challenges of the next generation software systems have motivated the development of self-adaptive software systems. Self-adaptation endows a system with the capability to adapt itself to internal changes and dynamics in the environment in order to achieve particular quality objectives. Central to the realization of self-adaptation is a feedback loop that monitors and adapts a managed system when needed to realize some goals. For a decentralized self-adaptive system, the managed system is typically a distributed system that is controlled by different feedback loops that act locally, in a relative independent manner. Despite substantial achievements in the field, important challenges remain open for future research. One key challenge is to provide assurances for the required quality properties in decentralized self-adaptive software systems. Providing such assurances is particularly difficult due to a high degree of uncertainty and decentralized decision making of adaptations.

To tackle this challenge, we propose an approach that combines formal modelling and verification with online learning. We take an architecture-centric perspective on self-adaptation, which provides the right level of abstraction to manage complexity and the required generality of solutions. To design feedback loops, we employ principles for control-based approaches, which provide a mathematical basis for analysing key properties of self-adaptation in decentralized settings, incl. stability and transient behaviour. Online learning provides the means to enable the self-adaptive system to acquire knowledge about design time uncertainties during system operation. E.g., the system learns over time the reliability of a set of services, which enables better service selection. The overall goal of this project is:

To study and develop a formally founded approach to assure the required quality properties of decentralized self-adaptive software systems.

The research goal poses the following concrete research questions:

RQ1: What are appropriate formal models of managed systems to design controllers for different quality requirements and how can these models be casted to discrete time dynamic systems, which provide the basis for controller design?

RQ2: What are appropriate control schemas (local) and decentralized architectures (system-wide) to realize self-adaptation for different quality requirements? What are appropriate online learning mechanisms to deal with the inherent uncertainties of decentralized self-adaptive systems at design time?

RQ3: How to guarantee the required qualities during design and realization of decentralized self-adaptive systems with online learning?

To steer and validate the research results we use case studies on "smart homes for elderly care" and "decentralized supply chains." We focus on three important types of adaptation requirements: performance (responsiveness under a certain workload), robustness (the ability to cope with errors during execution), and openness (the ability to deal with possibly new parts that enter/leave the system at will).

The proposed research is based on the assumption that basic support for consistent adaptations of the managed system is available (support to add/remove components/services, buffer messages, etc.), for which solutions exist that we can rely on in this project. Furthermore, we assume that the managed system is a cooperative system in which entities have common goals, and that the dynamics in the environment are orders of magnitude slower than communication and execution of adaptations. These assumptions put restrictions on the target application domain, scoping them to systems in our area of expertise. However, these assumptions hold for a large class of systems such as traffic and mobile applications. Example domains out of scope are real-time and competitive systems (i.e., systems with entities that pursue their own goals). These systems require dedicated approaches (e.g., real-time operating systems) or pose particular challenges (e.g., establishing trust).

2. Survey of the Field

The literature review is divided in sections that focus on specific aspects of self-adaptation. We conclude with explaining how this proposal aims to go beyond the state of the art.

Reference models. Central to self-adaptation is the use of an architectural model of the managed system at runtime. Pioneering work that introduced this idea is [1]. Rainbow [2] employs constraints defined over an architectural model of the managed system to realize self-adaptation. Particularly influential has been IBM's MAPE model [3] that describes the different components of an autonomic control loop: monitor, analyse, plan and execute. [4] proposes a three-layered reference model that maps to three fundamental activities in self-adaptive systems: component control at the bottom, change management in the middle, and goal management at the top. [5] presents FORMS (FOrmal Reference Model for Self-adaptation), which unifies different perspectives on architecture-based self-adaptation.

Middleware-based self-adaptation. [6] presents a component based reflective middleware that can be inspected and adapted by application code via a meta-interface to deal with changes in the environment. In MADAM [7], an adaptation middleware uses an architecture model that specifies dependencies between an application and its context to derive proper application variants and adapt the application when needed. The MUSIC [8] middleware uses a QoS-aware model that relates quality properties to system configurations to deploy a configuration based on the best utility. MACODO [9] provides abstractions for modelling adaptive service collaborations. The approach is validated in the domain of supply chains.

Service-oriented approaches. PLASTIC [10] considers both service-level specification adaptations (quality properties) and context-aware adaptations (resource characteristics), targeting mobile applications. MOSES [11] offers distributed monitoring components, a brokering service, and an optimization engine to supports runtime adaptation of composite services with different service levels. [12] studies performance profiling of virtualized, multi-tier Web applications, which supports model-driven engineering of an autonomic controller.

Formal approaches. [13] verifies system invariants and safety (e.g, hazards) of self-adaptive systems using graph models and rules, exploiting the local character of structural properties. [14] uses Petri nets and linear temporal logic to verify invariants, liveness, tolerance, and adaptation integrity. Conformance between the models and executions of the implementation are tested. [15] employs Markov models and probabilistic computation tree logic to verify response time and failure handling. [16] uses timed automata to verify a decentralized self-adaptive application. Requirements are expressed in timed computation tree logic. To deal with the unpredictable operating conditions of self-adaptive software systems, a recent article [17] argues for combined use of offline and runtime verification.

Handling uncertainty. [18] proposes fuzzy goals to reason about runtime adaptations, and [19] introduces the RELAX language that incorporates constructs to express uncertainty in requirements for self-adaptive systems. Loki [20] automatically discovers and mitigates requirements violations in self-adaptive systems. [21] employs techniques from possibility theory to assess uncertainties internal to self-adaptive systems to make adaptation decisions.

Decentralized self-adaptation. [22] expresses structural constraints over an architectural specification that are shared by component managers to automatically configure the system. [23] introduces a gossip protocol to improve the scalability of this approach. K-Components [24] reifies a system's architecture as a configuration graph of components and connectors that can be rewritten by a configuration manager to adapt the system when needed. [25] describes key attributes of decentralized self-adaptive systems derived from a number of case studies. [26] extends MAPE loops with support for inter-loop and intra-loop coordination, and [27] documents several recurring patterns of interacting MAPE loops.

Control-based approaches. Classic controllers (P, I, PID, etc.) have been applied extensively to computing systems. E.g., [28] describes the design of feedback loops for high-performance servers. Servers are modeled as difference equations and different types of controller models (e.g. PI and PID) are applied to deal with performance requirements (e.g., server response time, convergence). [29] employs multi-input multi-output (MIMO) techniques to deal with CPU and memory utilizations of a Web server. System models are derived from experiments, and the controller design is based on a linear quadratic regulator that defines control gains based on a cost function. [30] models a service-based system as a Markov chain and reliability requirements as reachability properties. The system model is then casted to a dynamic system to design and analyse a feedback loop system. [31] studies CPU allocation to applications hosted on virtualized servers based on two nested control loops. The outer loop improves response time by controlling CPU utilization in the inner loop. [32] applies limited look ahead control to optimize resource provisioning in data centres, while guaranteeing the quality of service. A two-layer controller selects the first action of the optimal path of actions to guide the system to a state within a prediction horizon. More as a decade ago, [33] advocated the use of learning techniques to deal with partial knowledge in control design.

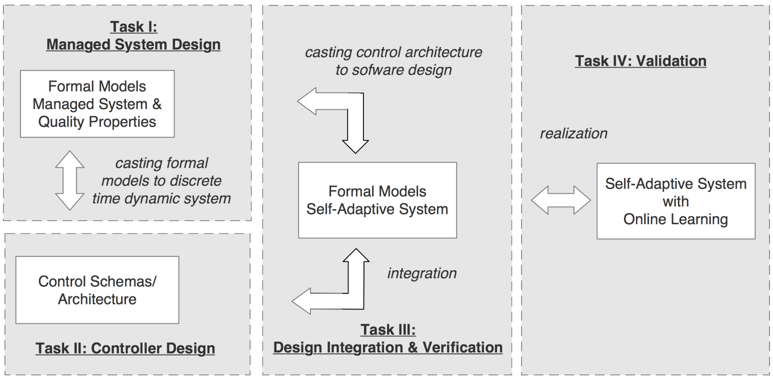
Control in robotics. Behaviour-based control is a classic approach for controlling robots. E.g., [34] presents an architecture that consists of three levels: functional level, control level, and planner level. Most schemes rely on stable control at a lower level while providing coordination at a higher level. Our work on behaviour-based coordination of automated guided vehicles [35] follows this approach. [36] models a multi-robotic system as an interconnected control system that allows feedback communication so that the robots can coordinated their behaviour. Stability analysis is based on vector Liapunov functions. [37] optimizes the positioning of robots in a partially known environment. Adaptive control architectures learn a parameterized environment model by propagating sensor data of robots.

Decentralized controllers. Decentralized control of computing systems is not very well studied. We discuss two representative examples. DEUCON [38] allocates local controllers to computing units that only coordinate with neighbours. Regular stability analysis is based on the location of poles of the composite system's transfer function. [39] addresses workload consolidation in large-scale server clusters. A local controller is associated to each server that tunes the number of operational cores on its server to satisfy service level agreements. System analysis is based on grouping controllers.

Challenges. State of the art recognizes the crucial role of feedback control loops in realizing self-adaptation. However, feedback loops often remain implicit in the design and implementation of the system. Control-based approaches emphasize synthesis and analysis of control schemas, but most studies focus on adding feedback loops to control resources (CPU cycles, storage, bandwidth) rather than controlling software entities. Furthermore, current approaches mainly support centralized or hierarchical control of adaptation. There is an agreement that decentralized controllers for software systems deserve more attention. A number of approaches have been proposed that employ formal methods to provide guarantees about properties of self-adaptation. However, important challenges on assurances need to be tackled (www.dagstuhl.de/en/program/calendar/semhp/?semnr=13511). Finally, handling uncertainties at runtime is not well studied. This project aims to go beyond the state of the art by: (i) study in depth decentralized self-adaptive systems, which are of growing importance, (ii) reconciling formal modeling and verification of architectures with applying principles from control theory to engineer self-adaptive systems, (iii) employing online learning techniques to deal with uncertainties resulting from incomplete knowledge at design time.

3. Project Description

Overview of the Project. The project will be organized in four tasks that incrementally and iteratively improve earlier project results:



Task I studies formal modelling of the managed system with its environment, dealing with research question RQ1. The models express the quality properties of interest in a quantitative/ stochastic manner, and are casted to discrete time dynamic systems to enable controller design. Task II studies the design and analysis of control schemas (local) and decentralized control architectures (system-wide), dealing with RQ2. This includes the study of learning techniques to handle design time uncertainties. Task III casts the controller to a software design, integrates this with the design of the managed system, and verifies the required quality properties, dealing with RQ3. Finally, Task IV maps the design to concrete software realizations and validates the solutions, including benchmarking of online learning.

Description of the Tasks. We give a description of the main research activities in each of the tasks and explain how these activities connect with one another.

Task I: Managed System Design

The goal of the first task is to study principled models of managed systems and their environments that are required to realize controllers for different self-adaptation requirements.

Principled models of managed systems. The application of control theory to software systems requires a model of the managed system (plant) that has to be controlled. A system model essentially relates current and past control inputs (to adapt the system) to current and past outputs (what we want to control) under disturbance inputs (uncontrollable factors). In addition, state variables of the system can be used to model the input-output relationship. For decentralized systems, both local subsystems and integrated systems have to be modelled. Existing systems can be modelled using system identification or they can be constructed from first principles [40]. As we do not assume that the managed system is already operational, we envision a different approach, where we model the system behaviour with an appropriate formalism and then cast this model to a discrete time dynamic system, which is an appropriate formalism to model software systems [40]. To deal with robustness, we plan to study Markov models for managed systems, where robustness properties can be formulated as probabilities to reach failure states. A Markov model can be casted to difference equations that express the system robustness in terms of input, output, and state, and failure probabilities map to control variables. A similar approach can be applied to deal with performance using probabilistic automata, where performance properties are expressed as probabilistic reachability properties that map to control variables in a difference equation model of the system. A candidate approach we envision to deal with openness is to model managed systems as automata, where openness is formulated as a quantitative property of the model, e.g., the degree of collaboration or the utility of the system. The automaton can be casted to difference equations with actions to add/remove subsystems map to control variables in this model.

Principled models of environments. Any software system operates in an environment that impacts the system via disturbances. When the environment is dynamic, which is the case in this research project, its dynamics need to be modelled. In this research, we consider discrete environments, which dynamics can be modelled as discrete time dynamic systems. We envision reusable models that capture essential aspects of environments from which controller design would benefit, while other relevant aspects will be modelled as disturbances, reducing the order of the model, which is important for neat controller design. Open environments are subject to dynamics that cannot be fully specified at design time. To handle these uncertainties we plan to study the use of stochastic models that capture monitored aspects of the environment in the form of probability distributions. An inspiring approach to support openness is proposed in [41] where the notion of variability is used to support dynamic discovery of quality-aware services. In line with the approach for managed systems described above, we will start from models of the environment specified in a suitable formalism that we cast to an appropriate format to design control schemas (the latter is studied in Task II).

Principled models for quality properties. Controlling a software system may require explicit models of the quality properties of interest. These quality property models are updated with data from the managed system and the environment, providing input for controllers to compare the current quality conditions of the system with the control objectives and steer the managed system to the control objectives. Measured data may require pre-processing before it can be used to update the quality models, e.g., smoothing out stochastics with filters, mapping sequence of events to probabilities using a learning approach, etc. In a decentralized setting, quality models inherently capture local knowledge of the overall quality properties of interest. For performance requirements, we can exploit established approaches (e.g., queuing models). However, models for the other considered quality properties (robustness, and in particular openness) are less well studied, and require additional research.

<u>Task II: Controller Design</u>

The goal of the second task is to study principled controller schemas for different adaptation requirements in decentralized settings.

Controller schemas for decentralized control. Decentralized control breaks down a control problem in sub-problems, which can be solved relatively independently. Design of local controllers starts from the discrete time dynamic system models of managed systems and environment and quality models (Task I). We will start our study of controller design with adaptive control [43] and reconfigurable control [44]. Adaptive control can deal with uncertainty in the model parameters of the managed system and the controller. Reconfigurable control supports online change of the controller to deal with invasive changes in the controlled system. Coordination of local controllers is central to decentralized control. In a strict decentralized setting, controllers do not coordinate directly [42], but, typically there will be indirect interactions. E.g., a local controller may affect the quality of service requests, triggering other controllers to adapt. A key aspect in control design of complex distributed systems is the inherent uncertainty. To deal with design time uncertainties (see Task I) we will combine adaptive/reconfigurable control with learning techniques. Learning allows the various models of the control schema to evolve while the system operates. We will start with studying reinforcement learning to handle uncertainties [45,46]. Other candidate approaches we plan to explore are Bayesian estimators [47] to learn probabilities in Markov chains for reliability, and Kalman filters [48] for updating performance models.

Multiple control objectives. Practical systems typically require support for multiple adaptation objectives. For example, a critical service may fail requiring immediate action while an adaptation to improve quality of service for clients is on-going. In architecture-based self-adaptation, some studies use utility functions to balance multiple concerns, e.g. [41,49], or pre-emption to switch adaption concerns [50]. On the other hand, control for multiple objectives is very well studied in control-based adaptation. In this project, we will study MIMO systems that allow controlling multiple outputs [29,51] of local controllers, and nested and layered architectures to design controllers with multiple control objectives.

Guaranteeing required properties in decentralized control systems. One of the powerful aspects of using a control-based approach is that control schemas can be formally analysed, e.g. for stability, steady-state error (accuracy), and transient behaviour (settling time, overshoot). Stability is a key property for decentralized systems, where system behaviour emerges from interacting subsystems. An important type of stability is bounded-input bounded-output stability, which can be determined via the location of the poles of a control system's transfer function. Decentralized controllers require system-wide stability analysis. Providing system-wide guarantees with arbitrary data exchange between controllers is an open research question in control theory. However, a number of control schemas have been described that allow restricted data exchange. Candidate approaches we plan to study in this project are stability analysis based on grouping of controllers (see survey) and identification of Lyapunov functions [52] (convergence of a systems' behaviour near equilibriums). Central in applying control theory to software systems is to understand the relationship between the quality properties that are subject of adaptation (throughput, fault handling, etc.) and classic control properties (stability, accuracy, etc.). Understanding this relationship and identifying principled approaches to guarantee end-to-end quality properties are key challenges.

<u>Task III: Design Integration and Verification</u>

The goal of this task is to integrate and verify the models of managed system and controller, and refine the design to support system realization.

Casting controllers to software designs. (Task 4.2) The controllers identified in Task II have to be casted to concrete software designs. In the simplest case, an existing implementation of the controller algorithm can be configured or the algorithm can directly be implemented. However, as explained in Task II, complex distributed systems typically require advanced controllers, which poses significant engineering challenges. Central to controller design is the decision logic that realizes the control law, which produces actions to adapt the managed system (possibly as workflows of adaptation steps). Decision making may require runtime models that maintain representations of the managed system, the environment, and quality properties. Depending on the control schema, specific subsystems have to be designed, such as a model estimator for an adaptive controller, or a model repository and model selector for a reconfigurable controller. To handle uncertainties, additional learning modules have to be designed and implemented for various models of the controller (Task II).

Design managed system. The model of the managed system is refined to support system implementation (legacy parts can be integrated in the design). Central to the design of the managed system are sensor and actuator design. Sensor design includes: providing measures to update the runtime models (e.g., response time of the system, bandwidth of the network), processing of data (e.g., smoothing out stochastics), timing (sensing frequency, synchronization of measurements, etc.), and efficiency (overhead). Actuator design includes: mapping control variables to software handlers (from parameter change to architectural reorganization), and timing of actuation (synchronization, etc.). Furthermore, environment sensors have to be designed based on the environment model, with similar design issues.

Guaranteeing the adaptation requirements during design. This activity includes two parts: 1) guaranteeing the adaptation requirements for the integrated design of the control system, and 2) preserving the guarantees during system realization. The integrated design includes the models of the managed system, its environment, and quality properties (result of Task I), and a model of the controller (result of Task II). Guaranteeing the adaptation requirements boils down to verifying the required quality properties that can be formulated as expressions over the design models. While we can start from regular verification techniques in this part, the study requires specific attention for the verification of local controllers with learning modules, interactions between the managed system and controller, and the interplay between local and system-wide properties. The second part requires disciplined design practice, for which we can rely on established approaches for correct refinement of designs, e.g., [53]. Architeture refinement may reveal the need for a revision of the original design models. Once the design is sufficiently well refined, the system can be implemented. To guarantee conformance of the implementation with the models, we plan to use model-based testing [54,55,56]. Model-based testing automatically generates test cases from a concise model of the system under test and shows that the implementation of the system behaves compliant with this model.

<u>Task III: Validation</u>

We plan to validate the research results with scenarios of two case studies in different domains, providing a basis for the validity and generality of the obtained research results.

Smart Homes for Elderly Care. The first case study is situated in the domain of ICT for elderly care. We recently performed a pilot project with a local municipality in Sweden, in which we studied requirements for the use of smart home technology and innovative services to improve the night care services provided by welfare helpers to elderly people living in their own houses. That pilot project provides the requirements the first case study.

Problem Scenarios. In the pilot study, we identified three primary classes of problems with night care services: (1) unnecessary visits of welfare helpers that could be avoided (e.g., an elderly sleeps quietly), (2) anticipatory visits that could anticipate a lot of overhead, if detected in time (e.g., a saturating diaper), and (3) critical visits that could avoid severe problems (e.g., an elderly fell). Today, ambulant welfare helpers do not have access to the required information needed to deal with these problems. To ensure sustainability, the solution should be open to integrate new services and technology. Furthermore, the solution should support personalization to the context of use and robustness to degrading sensors.

Envisioned Approach. Our objective is to study a decentralized self-adaptive software system that consists of two types of subsystems: smart home systems and mobile care assistants, plus a supporting department server. At each home, a smart home system collects and synthesizes data from a (wireless) sensor network and sends useful information to the mobile care assistants. Each welfare helper uses such a system to supports her decisions about visits and interact with the elderly or other persons when needed. The main role of the department server is to provide a repository where new software can be downloaded by the subsystems.

Self-Adaptation Challenges. We consider three types of self-adaptation requirements. First, we aim to support self-configuration of the subsystems, that is, dynamic discovery of (new) services and sensors, and self-configuration based on personal needs and environment conditions. E.g., the system may activate a new service that enables an elderly to alarm a welfare helper via voice when he/she enters the bad room at night. Second, we aim to support self-healing of the smart home system. Self-healing will exploit redundancy of the sensor infrastructure when the quality of particular sensors drop or some of them fail. Third, we aim to support context-aware adaptation. For example, a mobile care assistant may offer a service that provides particular information regarding an elderly once the welfare helper approaches the home of the elderly. Learning approaches have to be identified to effectively handle quality drop of sensors, as well as supporting context-aware adaptation of services.

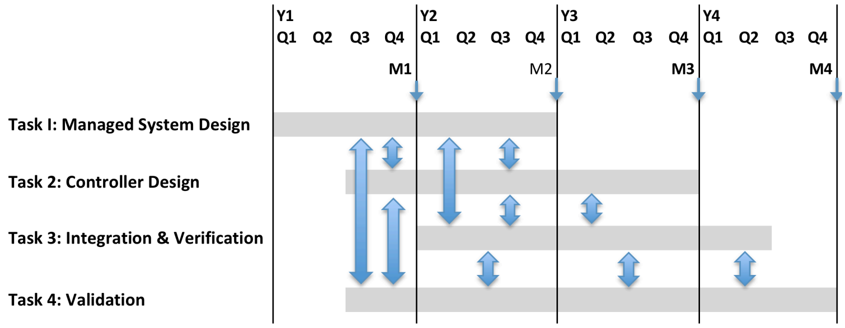
Decentralized supply chains. The second case study is situated in the domain of ICT for business support. Our focus is on supply chains that require collaboration among multiple entities, probably from multiple companies. In previous research [9], we have studied supply chains that are managed by a trusted third-party. This excludes collaborations without such a party. In this case study, we apply the research to supply chains without a central coordinator.

Problem scenarios. A supply chain consists of a network of transporters, warehouses, retailers, etc. which collaborate to create a product flow from supplier to customer. To deal with the dynamic and unpredictable market, a supply chain needs to be easily adapted to current needs. We focus on two particular problems: demand changes and partner dynamics. The demand of a supply chain may dynamically change based on external factors that are often difficult to predict. In an open environment, supply chain partners may dynamically change due to unavailability or withdrawal of partners, or new partners that become available.

Envisioned Approach. To realize the scenarios and handle the identified problems, we will develop a decentralized self-adaptive software system. The managed system will consist of a network of subsystems, one for each supply chain partner, which will automate the flow of information in the chain. We assume that these subsystems can use local information systems (considered as black boxes) that provide access to local data, e.g., stock of a warehouse or a traffic schedule. We will use a Web service- based approach to realize the system.

Self-Adaptation Challenges. Change of the demand requires optimizing the performance of the supply chain, e.g., by changing the policies of partners regarding inventory or replenishment. Dynamic change of supply chain partners requires both support for robustness to unavailability of partners and support for openness to new partnership. Handling such changes will affect the flows in the supply chain. Local controllers will be added to the subsystems that will deal with demand changes and partner dynamics. Learning approaches need to be identified to learn new policies for handling demand uncertainty and deal with partner availability.

Project Planning. The next figure shows the tasks with their interactions over 4 years.



The work plan defines four milestones (M1-M4) that support assessment of the project progress. M1 provides an initial version of the design of managed systems and controllers for case study scenarios with single adaptation requirements. M2 provides a mature integrated design of a decentralized self-adaptive system for scenarios with single adaptation requirements. M3 provides a mature integrated design of a decentralized self-adaptive system for case study scenarios with multiple adaptation requirements. M4 completes the case studies. Validation will give evidence about the realization of research goals.

4. Significance

The key role of self-adaptation for managing complex software systems and guaranteeing their qualities is beyond dispute. Despite advances in the field, an important challenge that needs to be tackled is to provide assurances for the required qualities. This challenge is particularly difficult for decentralized self-adaptive software systems due to a high degree of uncertainty and decentralization of decision making of adaptations. To tackle this challenge, this project follows an innovative approach that reconciles an architecture-centric perspective on self-adaptation with applying principles for control theory to design control loops. To deal with the inherent uncertainties, the approach combines formal modelling and analysis during system design with learning after deployment. Validation is planned in two domains with different adaptation requirements. If successful, the project will contribute to a fundamental understanding of engineering self-adaptation in complex software systems, add to significant improvements of qualities of next generation software systems creating economical impact for industry and societal impact for people, and provide a basis for future research.

5. Preliminary Results

Central aspects that are essential to the success of this project proposal are: modeling abstractions, formal design and verification, and control-based self-adaptation. During the last years, the team has obtained initial research results for each of these aspects upon which the project can build. [25] presents key attributes of decentralized self-adaptive systems and [9] introduces modeling abstractions that provide a basis to develop service-based self-adaptive systems. [60] surveys the use of formal methods in self-adaptive systems. FORMS [5] offers a formally specified reference model that servers as a guiding framework for modeling self-adaptive systems. [16,57] present case studies on formal verification of robustness and openness properties in decentralized systems. [27] documents a set of architectural patterns for decentralized self-adaptive systems. [26] zooms in on the interactions between feedback loops, distinguishing intra-loop interactions from inter-loop interactions. [30] models a service-based system as a Markov chain and reliability requirements as reachability properties. The system model is then casted to a dynamic system to design and analyze a feedback loop system. [58] extends this work for dynamical binding of components, enabling automatic selecting of the most suitable configuration. [59] studies concurrent task scheduling with control objectives formulated as a cost function and a set of input/state constraints.

Bibliography

[1] P. Oreizy, N. Medvidovic, R. Taylor, Architecture-based Runtime Software Evolution. International Conference on Software Engineering, 1998
[2] D. Garlan, S.W. Cheng, A.C. Huang, et al., Rainbow: Architecture-based Self Adaptation with Reusable Infrastructure, IEEE Computer, 37(10), 2004
[3] J. Kephart, D. Chess. The vision of autonomic computing, IEEE Computer, 36(1), 2003
[4] J. Kramer and J. Magee. Self-managed systems: an architectural challenge, FOSE 2007
[5] D. Weyns, S. Malek, J. Andersson, FORMS: Unifying Reference Model for Formal Specification of Distributed Self-Adaptive Systems, ACM TAAS 7(1), 2012
[6] G. Coulson, G. Blair, P. Grace, et al., A generic component model for building systems software, ACM Transactions on Computer Systems, 26(1), 2008
[7] K. Geihs, P. Barone, F. Eliassen, et al., A comprehensive solution for application-level adaptation, Journal on Software—Practice & Experience, 39(4), 2009
[8] R. Rouvoy et al., MUSIC: Middleware Support for Self-Adaptation in Ubiquitous and Service-Oriented Environments. Lecture Notes In Computer Science, Vol. 5525, 2009
[9] R. Haesevoets, D. Weyns, T. Holvoet, Architecture-Centric Support for Adaptive Service Collaborations, ACM Transactions on Software Engineering and Methodology, accepted
[10] M. Autili, P. Di Benedetto, P.Inverardi, Context-Aware Adaptive Services: The PLASTIC Approach, Fundamental Approaches to Software Engineering, 2009
[11] Cardellini et al., Adaptive Management of Composite Services under Percentile-Based Service Level Agreements. Lecture Notes in Computer Science vol. 6470, 2010
[12] G. Toffetti, A. Gambi, M. Pezzè and C. Pautasso, Engineering Autonomic Controllers for Virtualized Web Applications, Web Engineering 2010
[13] B. Becker et al., Symbolic invariant verification for systems with dynamic structural adaptation, International Conference on Software Engineering, 2006.
[14] J. Zhang and B. Cheng, Model-based development of dynamically adaptive software, International Conference on Software Engineering 2006
[15] R. Calinescu, L. Grunske, M. Kwiatkowska, et al., Dynamic QoS management and optimization in service systems, IEEE Transactions on Software Engineering, 37(3), 2011.
[16] U. Iftikhar et al., A Case Study on Formal Verification of Self-Adaptive Behaviours in a Decentralized System, Foundations of Coordination Languages and Self Adaptation 2012
[17] R. Calinescu, C. Ghezzi, M. Kwiatkowska, R. Mirandola: Self-adaptive software needs quantitative verification at runtime. Communications of the ACM 55(9), 2012
[18] L. Baresi et al., Fuzzy goals for requirements-driven adaptation, IEEE RE, 2010
[19] J. Whittle et al., RELAX: Incorporating uncertainty into the specification of self-adaptive systems, Requirements Engineering Conference, RE 2009
[20] A. J. Ramirez, A. C. Jensen, B. H. Cheng, and D. B. Knoester, Automatically exploring how uncertainty impacts behavior of dynamically adaptive systems, ASE 2011
[21] N. Esfahani et al., Taming Uncertainty in Self-Adaptive Software, European Software Engineering Conference & Symposium on Foundations of Software Engineering, 2011
[22] I. Georgiadis, J. Magee, J. Kramer. Self-Organising Software Architectures for Distributed Systems. Workshop on Self-Healing Systems 2002
[23] D. Sykes, J. Magee, and J. Kramer, FlashMob: Distributed Adaptive Self-Assembly, Software Engineering for Adaptive and Self-Managing Systems, 2011
[24] J. Dowling, V. Cahill. The k-component architecture meta-model for self-adaptive software, Metalevel Architectures and Separation of Crosscutting Concerns, Springer, 2001
[25] D. Weyns et al., On Decentralized Self-Adaptation: Lessons from the Trenches & Challenges for the Future, Software Eng. for Adaptive and Self-Managing Systems, 2010
[26] P. Vromant et al., On Interacting Control Loops in Self-adaptive Systems, SEAMS, 2011
[27] D. Weyns, B. Schmerl, V. Grassi, et al., On Patterns for Decentralized Control in Self-Adaptive Systems, Lecture Notes in Computer Science vol. 7475, 2012
[28] T. Abdelzaher et al., Feedback performance control in software services, IEEE Control Systems 23(3), 2003

[29] Y. Diao et al., Using MIMO feedback control to enforce policies for interrelated metrics with application to the Apache Web server, Network Operations and Management, 2002
[30] A. Filieri, C. Ghezzi, A. Leva, M. Maggio: Self-adaptive software meets control theory: A preliminary approach supporting reliability requirements, ASE 2011
[31] X. Zhu, Z. Wang, and S. Singhal, Utility-Driven Workload management using Nested Control Design, American Control Conference, 2006
[32] D. Kusic et al. Power and Performance Management of Virtualized Computing Environments via Lookahead Control, Journal on Cluster Computing, 12(1), 2009
[33] P. Antsaklis, Intelligent Learning Control, IEEE Control Systems 15(3), 1995
[34] F. Noreils, Toward a robot architecture integrating cooperation between mobile robots: Application to indoor environment, International Journal of Robotics Research, 12(1), 1993
[35] D. Weyns, et al., Decentralized Control of Automatic Guided Vehicles: Applying Multi-Agent Systems in Practice, ACM SIGPLAN OOPSLA, Development Track 2008
[36] J. Feddema, et al., Decentralized control of cooperative robotic vehicles: theory and application, IEEE Transactions on Robotics and Automation, 18(5), 2002
[37] M. Schwager, D. Rus and J. J. Slotine, Decentralized, Adaptive Coverage Control for Networked Robots, International Journal of Robotics Research, March, 28(3), 2009
[38] X. Wang, et al., DEUCON: Decentralized End-to-End Utilization Control for Distributed Real-Time Systems, IEEE Transactions on Parallel and Distributed Systems, 18(7), 2007
[39] R. Wang and N. Kandasamy, On the design of decentralized control architectures for workload consolidation in large-scale server clusters, Autonomic computing, ICAC 2012
[40] J. Hellerstein et al., Feedback Control of Computing Systems, IEEE Press, 2004
[41] K. Geihs et al., Modeling of Component-Based Self-Adapting Context-Aware Applications for Mobile Devices, Lecture Notes in Computer Science vol. 227, 2007
[42] L. Bakule, Decentralized Control: An Overview, Annual Reviews in Control 32, 2008
[43] K. Astrom, B. Wittenmark. Adaptive Control Addison-Wesley, 1994
[44] Y. Zhanga, J. Jiangb, Bibliographical review on reconfigurable fault-tolerant control systems, Annual Reviews in Control, 32(2), 2008
[45] G. Tesauro, Reinforcement Learning in Autonomic Computing, Internet Computing 11(1), 2007
[46] R. Hafner, M. Riedmiller, Reinforcement learning in feedback control - Challenges and benchmarks from technical process control, Machine Learning, 84:137–169, 2011
[47] A. Filieri et al., Run-time efficient probabilistic model checking, ICSE 2011
[48] T. Zheng, M. Woodside, and M. Litoiu. Performance model estimation and tracking using optimal filters, IEEE Transactions on Software Engineering, 34(3), 2008
[49] S. Cheng, D. Garlan, and B. Schmerl. Architecture-based self-adaptation in the presence of multiple objectives, Self-adaptation and Self-managing Systems 2006
[50] R. Raheja, S. Cheng, D. Garlan, B. Schmerl, Improving architecture-based self-adaptation using pre-emption, Self-Organizing Architectures, LNCS vol. 6090, 2010.
[51] S. Skogestad et al., Multivariable Feedback Control: Analysis and Design, Wiley, 2007
[52] J. Daafouza, J. Bernussoub, Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties, Systems & Control Letters 43, 2001
[53] M. Moriconi, X. Qian, and R. Riemenschneider, Correct Architecture Refinement. IEEE Transactions on Software Engineering, 21(4), 1995
[54] R. Hierons et al., Formal specifications to support testing, Comp. Surveys, 41(2), 2009
[55] J. Tretmans, Testing concurrent systems: A formal approach, Concurrency Theory, 1999
[56] Uppaal TRON, http://people.cs.aau.dk/~marius/tron/index.html
[57] D. Gil de la Iglesia and D. Weyns, Guaranteeing Robustness in a Mobile Learning Application Using Formally Verified MAPE Loops, SEAMS 2013
[58] A. Filieri, C. Ghezzi, A. Leva, M. Maggio, Reliability-Driven Dynamic Binding via Feedback Control, Software Engineering for Self-Managing and Adaptive Systems, 2012
[59] M. Maggio, A. V. Papadopoulos, and A. Leva, On the Use of Feedback Control in the Design of Computing System Components, Asian Journal of Control 15(1), 2013
[60] D. Weyns, U. Iftikhar, D. Gil de la Iglesia, and T. Ahmad, A Survey on Formal Methods in Self-Adaptive Systems, Formal Methods for Self-Adaptive Systems, FMSAS 2012

Kod
2013-45635-105143-33
Name of applicant
Weyns, Daniel
Date of birth
580930-1533

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Title of research programme
Assurances in Decentralized Self-Adaptive Software Systems

Appendix B

Curriculum vitae

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Short CV – Danny Weyns – April 2013

Higher education degree

June 2011	Master in Applied Informatics, Katholieke Universiteit Leuven Thesis: "Serialization of Distributed Execution State in Java"
September 2009	Certificate Software Architecture Professional, Carnegie Mellon University, SEI, USA
February 2010	Certificate ATAM Evaluator, Carnegie Mellon University, SEI. USA

Doctoral degree

October 2006	PhD in Computer Science, Katholieke Universiteit Leuven "An Architectural Approach for Software Engineering of Situated Multi-Agent Systems." Committee T. Holvoet, Y. Berbers, H. Blockeel, W. Joosen, V. Parunak, F. Zambonelli

Postdoctoral positions

1/2007-10/2007	Post-doctoral fellow funded by Katholieke Universiteit Leuven, Belgium "Web service composition with multi-agent systems" DistriNet Labs, Katholieke Universiteit Leuven
10/2007-9/2010	Research fellow of the Research Foundation Flanders (FWO) "Study of an architecture description language for decentralized software systems" DistriNet Labs, Katholieke Universiteit Leuven
10/2010-3/2011	Research fellow of the Research Foundation Flanders (FWO) "A framework for decentralized self-adaptive systems" DistriNet Labs, Katholieke Universiteit Leuven
2008	Visiting researcher Carnegie Mellon University, Software Engineering Institute, Pittsburgh (4/2008-5/2008) funded by Research Foundation Flanders (FWO)
2009-2010	Visiting researcher Valoria Lab of the Université de Bretagne-Sud, France (5/2009-12/2009 and 6/2010-7/2010) funded by Research Foundation Flanders (FWO)

Docent level

12/2011	Docent in Computer Science, Software Engineering Linnaeus University Sweden "A Foundation for Engineering Decentralized Self-Adaptive Software Systems"

Present position

From 3/2011	Associate professor (docent, currently 45% research) School of Computer Science, Physics and Mathematics Linnaeus University, Växjö Campus, Sweden Website: http://homepage.lnu.se/staff/daweaa/index.htm

Supervision

Current Ph.D students

From 2/2012	Usman Iftakhir, Validating qualities in self-adaptive software systems
From 5/2012	Didac Gil de la Iglesia, Linnaeus University (Co-supervisor)
From 1/2013	Kostiantyn Kucher, Pre-doctoral student

Previous Ph.D students

2006-2/2012	Robrecht Haesevoets, Katholieke Universiteit Leuven
2010-2/2012	Bartosz Michalik, Double degree Katholieke Universiteit Leuven
2005-10/2009	Nelis Boucké, Katholieke Universiteit Leuven (Counselor).

Ph.D committees

From 2012	Tobias Ohlsson, Linnaeus University (examinator) Daniel Toll, Linnaeus University (examinator)
From 2010	Stijn Vandael, Katholieke Universiteit Leuven, Coordination patterns for charging plug-in hybrid vehicles in a Smart Grid (supervision committee)
From 2010	Xiong Qin, Université de Bretagne-Sud, France, Software environment for architecting service-oriented systems based on Pi-ADL. (supervision committee)

PhD juries

2/2010	Maruf Pasha, France, Université de Bretagne-Sud, France
9/2010	Nadeem Akhtar, Université de Bretagne-Sud, France
7/2011	Maarten Bynens, Katholieke Universiteit Leuven, Belgium
9/2011	Elke Steegmans, Katholieke Universiteit Leuven, Belgium
1/2012	Koen Buyens, Katholieke Universiteit Leuven, Belgium
3/2013	Koen Yskout, Katholieke Universiteit Leuven, Belgium
5/2013	Tobias Gutzmann, Linnaeus University
6/2013	Ilir Jusufi, Linnaeus University

Master students

2013	Tobias Hakansson, Linnaeus University
2013	Quan Nguyen, Linnaeus University
2003-2012	Supervisor of 18 degree projects

Projects and grants

2/2012-1/2015	A Foundation for Engineering Decentralized Self-Adaptive Software Systems (100K EUR), Marie Curie Career Integration Grant
7/2012-6/2013	CareSmart: Improving Welfare Services for Elderly People Using Smart Homes, Funded by Nybro's Welfare Department (http://www.nybro.se/) and Linnaeus' Information Engineering Center (http://lnu.se/IEC).
4/2009-3/2011	Software Product Lines for Logistic Systems (E'SPLS) in collaboration with Egemin (Research budget 250K EUR) Funded by the Institute for Promotion of Innovation through Science and Technology in Flanders.

Publications

4/2013	h-index 25, 2600 citations (Google Scholar 4/2013)
2002-2013	Author/co-author of 20 international reviewed journal articles, 28 book chapters, and +80 international conference and workshop papers
2010	Author Architecture-Based Design of Multi-Agent Systems, Springer

Awards and credits

2012	Marie Curie Career Integration Grant awarded
2007	Finalist Cor Baayen Award for a most promising young researcher in computer science and applied mathematics by European Research Consortium for Informatics and Mathematics (ERCIM)
2009-2013	4 Dagstuhl Seminars, incl. 3 on Engineering Self-Adaptive Systems
2005-2013	6 invited talks, 13 invited lectures, and 4 invited tutorials

Curriculum Vitae

1 Higher education degree

- March 2007 - December 2008: Master of Science in Computer Engineering (Laurea Magistrale in Ingegneria Informatica), Politecnico di Milano, summa cum laude.
- September 2003 - March 2007: Bachelor Degree in Computer Engineering (Laurea in Ingegneria Informatica), Politecnico di Milano.

2 Doctoral degree

- January 2009 - December 2011: Ph.D. in Information and Communication Technology, Politecnico di Milano, Dipartimento di Elettronica e Informazione. Thesis: "Control based design of computing systems". Advisor: Prof. Alberto Leva.

3 Postdoctoral position - Present position

- January 2012 - December 2013: Postdoctoral Associate at Lunds Tekniska Högskola, under the supervision of Prof. Karl-Erik Årzén. The time for research is 100%.

4 Previous positions

- February 2010 - March 2011: Visiting Ph.D. Student at Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory. Supervisor of Visiting Research Period: Prof. Anant Agarwal.

5 Awards

- Progetto Rocca doctoral fellowship for visiting the Computer Science and Artificial Intelligence Laboratory at MIT both during Spring 2010 and Fall 2011.
- Ph.D. Scholarship funded by the Italian Government.

Appendix C

List of publications

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Publications – Danny Weyns – April 2013

International reviewed journals

IJ.20	R. Haesevoets, D. Weyns, T. Holvoet, Architecture-Centric Support for Dynamic Service Collaborations, ACM Transactions on Software Engineering and Methodology (TOSEM), accepted, 2013
IJ.19	D. Weyns, S. Malek, J. Andersson, FORMS: Unifying Reference Model for Formal Specification of Distributed Self-Adaptive Systems, ACM Transactions on Autonomous and Adaptive Systems (TAAS), 7(1), 2012
IJ.18	R. Claes, T. Holvoet, D. Weyns, A Decentralized Approach for Anticipatory Vehicle Routing using Delegate Multi-Agent Systems, IEEE Transactions on Intelligent Transportation Systems, 12 (2), pp. 364-373, 2011
IJ.17	N. Boucke, D. Weyns, T. Holvoet, Composition of Architectural Models: Empirical Analysis and Language Support, Journal of Systems and Software (JSS), 83(11), pp. 2108-2127, 2010
IJ.16	D. Weyns, M. Georgeff, Self-Adaption using Multi-Agent Systems, IEEE Software, 27(1), pp. 86-91, 2010
IJ.15	D. Weyns, N. Boucke, and T. Holvoet, A Field-Based Versus a Protocol-Based Approach for Adaptive Task Assignment, Journal on Autonomous Agents and Multi-Agent Systems, 17(2), pp. 288-319, 2008
IJ.14	D. Weyns, A. Omicini, and J. Odell, Environment as a first-class abstraction in multiagent systems, International Journal on Autonomous Agents and Multi-Agent Systems 14 (1), pp. 5-30, February, 2007
IJ.13	D. Weyns, M. Schumacher, A. Ricci, M. Viroli, and T. Holvoet, Environments in multiagent systems, The Knowledge Engineering Review 20 (2), pp. 127-141, June, 2005
IJ.12	D. Weyns, and T. Holvoet, A formal model for situated multi-agent systems, Fundamenta Informaticae 63 (2-3), pp. 125-158, November, 2004
IJ.11	D. Weyns, E. Steegmans, and T. Holvoet, Towards active perception in situated multi-agent systems, Applied Artificial Intelligence 18 (9-10), pp. 867-883, October, 2004
IJ.10	D. Weyns, A. Helleboogh, R. Haesevoets, T. Holvoet, W. Joosen, The MACODO Middleware for Context-Driven Dynamic Agent Organizations, ACM Transaction on Adaptive and Autonomous Systems, TAAS, 5(1), 2010
IJ.09	D. Weyns, A. Helleboogh, R. Haesevoets, The MACODO Organization Model for Context-driven Dynamic Agent Organizations, ACM Transaction on Adaptive and Autonmous Systems, TAAS, 5(4), 2010
IJ.08	D. Weyns, H.V.D. Parunak, O. Shehory, The Future of Software Engineering and Multiagent Systems, International Journal on Agent Oriented Software Engineering, 3(4), pp. 369-377, 2009
IJ.07	D. Weyns, A. Helleboogh, T. Holvoet, How to Get Multiagent Systems Accepted in Industry? International Journal on Agent Oriented Software Engineering, 3(4), pp. 383-390, 2009
IJ.06	D. Weyns and T. Holvoet, Mediating Agents' Activities in Situated Multiagent Systems, CPE Special Issue on Coordination in Pervasive Environments, Ubiquitous Computing and Communications Journal, June 2008
IJ.05	D. Weyns and T. Holvoet, Architectural Design of a Situated Multiagent System for Controlling Automatic Guided Vehicles, Special Issue on Multiagent Systems and Software Architecture, International Journal on Agent Oriented Software Engineering, 2(1), pp. 90-128, 2008
IJ.04	D. Weyns, A. Helleboogh, T. Holvoet, M. Schumacher, The Environment in Multiagent System: A Middleware Perspective, Special Issue on Engineering Environments for Multiagent Systems, International Journal on Multiagent and Grid Systems, 5(1), pp. 93-108, 2009

IJ.03 K. Schelfthout, D. Weyns, and T. Holvoet, Middleware for protocol-based coordination in mobile applications, IEEE Distributed Systems Online 7 (8), pp. 1-18, August, 2006

IJ.02 D. Weyns, and T. Holvoet, On the role of environments in multiagent systems, Informatica 29 (4), pp. 409-421, 2005.

IJ.01 D. Weyns, E. Truyen, and P. Verbaeten, Serialization of Distributed Threads in Java, Scalable Computing: Practice and Experience 6 (1), pp. 81-98, 2005.

Book chapters

BC.28 D. Weyns, B. Schmerl, V. Grassi, S. Malek, R. Mirandola, C. Prehofer, J. Wuttke, J. Andersson, H. Giese, and K. Goschka, On Patterns for Decentralized Control in Self-Adaptive Systems, Software Engineering for Self-Adaptive Systems II, Lecture Notes in Computer Science vol. 7475, Springer, 2012

BC.27 de Lemos et al., Software engineering for self-adaptive software systems: A second roadmap, Software Engineering for Self-Adaptive Systems II, Lecture Notes in Computer Science vol. 7475, Springer, 2012

BC.26 J. Juziuk, D. Weyns, and T. Holvoet, Design Patterns for Multi-Agent Systems: A Systematic Literature Review, (O. Shehory, A. Sturm eds.) Engineering Multi-agent Systems, Springer, 2013

BC.25 R. Haesevoets, D. Weyns, M. H. C. Torres, A. Helleboogh, T. Holvoet, and W. Joosen, A Middleware Model in Alloy for Supply Chain-Wide Agent Interactions, Lecture Notes in Computer Science, Springer

BC.24 J. Andersson, R. de Lemos, S. Malek, and D. Weyns. Modeling Dimensions of Self-Adaptive Software Systems, (B. H. C. Cheng, R. de Lemos, H. Giese, P. Inverardi, and J. Magee, eds.) Lecture Notes in Computer Science, vol. 5525, 2009.

BC.23 B. Cheng, R. de Lemos, H. Giese, P. Inverardi, J. Magee, J. Andersson, B. Becker, N. Bencomo, Y. Brun, B. Cukic, G. Di Marzo Serugendo, S. Dustdar, A. Finkelstein, C. Gacek, K. Geihs, V. Grassi, G. Karsai, H. Kienle, J. Kramer, M. Litoiu, S. Malek, R. Mirandola, H. Muller, S. Park, M. Shaw, M. Tichy, M. Tivoli, D. Weyns, J. Whittle, Software Engineering for Self-Adaptive Systems: A Research Roadmap, Lecture Notes in Computer Science, vol. 5525, 2009.

BC.22 N. Boucke, D. Weyns, R. Hilliard, T. Holvoet, and A. Helleboogh, Characterizing Relations between Architectural Views, 2nd European Conference on Software Architecture, ECSA 2008, (Morrison, R. Balasubramaniam, D. Falkner, K., eds.) Lecture Notes in Computer Science, vol. 5292, 2008

BC.21 A. Helleboogh, D. Weyns, T. Holvoet, On the Role of Software Architecture for Simulating Multi-Agent Systems, (A. Uhrmacher and D. Weyns eds.), Chapter 6 in: Multi-Agent Systems: Simulation and Applications, Taylor & Francis, 2009

BC.20 R. Haesevoets, B. Van Eylen, D. Weyns, A. Helleboogh, T. Holvoet, W. Joosen, Managing agent interactions with context-driven dynamic organizations, Revised Selected and Invited Papers Engineering Environment-Mediated Multi-Agent Systems, (Weyns, D. and Brueckner, S. and Demazeau, Y., eds.), Lecture Notes in Computer Science, vol. 5049, 2008

BC.19 H. V. D. Parunak, S. Brueckner, D. Weyns, T. Holvoet, P. Verstraete, P. Valckenaers, E Pluribus Unum: Polyagent and Delegate MAS Architectures, Revised and Invited Papers Multi-Agent-Based Simulation VIII (L. Antunes, M. Paolucci, E. Norling eds.), Lecture Notes in Computer Science, vol. 5003, 2008

BC.18 D. Weyns, and T. Holvoet, Architecture-centric software development of situated multiagent systems, Engineering Societies in the Agents World VII, Revised Selected and Invited Papers (Dikenelli, O. and Gleizes, M.P. and Ricci, A., eds.), vol 4408, Lecture Notes in Computer Science, pp. 1-24, 2007

BC.17 D. Weyns, and T. Holvoet, A framework for situated multiagent systems, Software Engineering for Large Scale Systems V, Revised Selected and Invited

Papers (Choren, R., Garcia, A., Giese, H., Leung, H., Lucena, C., and Romanovsky, A., eds.), Lecture Notes in Computer Science, pp. 1-28, 2007

BC.16 D. Weyns, and T. Holvoet, From reactive robotics to situated multiagent systems: A historical perspective on the role of environment in multiagent systems, Engineering Societies in the Agents World VI, Revised Selected and Invited Papers (Dikenelli, O. and Gleizes, M.P. and Ricci, A., eds.), vol 3963, Lecture Notes in Computer Science, pp. 63-88, 2006

BC.15 D. Weyns, and T. Holvoet, A reference architecture for situated multiagent systems, Proceedings of the Third International Workshop on Environments for Multiagent Systems (Weyns, D. and Paruank, V. and Michel, F., eds.), vol 4389, Lecture Notes in Computer Science, pp. 120-170, 2006

BC.14 D. Weyns, G. Vizzari, and T. Holvoet, Environments for situated multiagent systems: Beyond infrastructure, Environments for Multi-Agent Systems II (Weyns, D. and Parunak, V. and Michel, F., eds.), vol 3830, Lecture Notes in Computer Science, pp. 1-17, 2006

BC.13 D. Weyns, K. Schelfthout, and T. Holvoet, Exploiting a virtual environment in a real-world application, Environments for Multiagent Systems II (Weyns, D. and Parunak, V. and Michel, F., eds.), vol 3830, Lecture Notes in Computer Science, pp. 218-234, 2006

BC.12 N. Boucké, D. Weyns, K. Schelfthout, and T. Holvoet, Applying the ATAM to an architecture for decentralized control of a transportation system, Revised Papers of Second International Conference on Quality of Software Architectures (Hofmeister, C. and Crnkovic, I. and Reussner, R., eds.), vol 4214, Lecture Notes in Computer Science, pp. 180-198, 2006

BC.11 D. Weyns, A. Helleboogh, and T. Holvoet, The Packet-World: A testbed for investigating situated multiagent systems, Software Agent-Based Applications, Platforms, and Development Kits, (Unland, R. and Klush, M. and Calisti, M., eds.), Whitestein Series in Software Agent Technologies, Birkhauser Verlag, Basel, Boston, Berlin, September, pp.383-408, 2005

BC.10 D. Weyns, K. Schelfthout, T. Holvoet, and O. Glorieux, Towards adaptive role selection for behavior-based agents, Adaptive Agents and Multi-Agent Systems III: Adaptation and Multi-Agent Learning (Kudenko, D. and Kazakov, D. and Alonso, E., eds.), vol 3394, Lecture Notes in Computer Science, pp. 295-314, 2005

BC.09 D. Weyns, H. Parunak, F. Michel, T. Holvoet, and J. Ferber, Environments for multi-agent systems, state-of-the-art and research challenges, Environments for multi-agent systems (Weyns, D. and Parunak, H.V.D. and Michel, F., eds.), vol 3374, Lecture Notes in Computer Science, pp. 1-48, 2005

BC.08 D. Weyns, E. Steegmans, and T. Holvoet, Integrating free-flow architectures with role models based on statecharts, Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications (Choren, R. and Garcia, A. and Lucena, C. et al., eds.), vol 3390, Lecture Notes in Computer Science, pp. 104-120, 2005

BC.07 D. Weyns, K. Schelfthout, T. Holvoet, T. Lefever, and J. Wielemans, Architecture-centric development of an AGV transportation system, Multi-Agent Systems and Applications IV (Pechoucek, M. and Petta, P. and Varga, L.Z., eds.), vol 3690, Lecture Notes in Computer Science, pp. 640-645, 2005

BC.06 E. Steegmans, D. Weyns, T. Holvoet, and Y. Berbers, A design process for adaptive behavior of situated agents, Agent-Oriented Software Engineering V (Odell, J. and Giorgini, P. and Müller, J.P., eds.), vol 3382, Lecture Notes in Computer Science, pp. 109-125, 2005

BC.05 A. Helleboogh, T. Holvoet, D. Weyns, and Y. Berbers, Extending time management support for multi-agent systems, Multi-Agent and Multi-Agent-Based Simulation: Joint Workshop MABS 2004, New York, NY, USA, July 19, 2004, Revised Selected Papers (Davidsson, P. and Logan, B. and Takadama, K., eds.), vol 3415, Lecture Notes in Computer Science, pp. 37-48, 2005

BC.04 A. Helleboogh, T. Holvoet, D. Weyns, and Y. Berbers, Towards time management adaptability in multi-agent systems, Adaptive Agents and Multi-Agent Systems III: Adaptation and Multi-Agent Learning (Kudenko, D. and Kazakov, D. and Alonso, E., eds.), vol 3394, Lecture Notes in Computer Science, pp. 88-105, 2005

BC.03 D. Weyns, and T. Holvoet, Regional synchronization for simultaneous actions in situated multi-agent systems, Multi-Agent Systems and Applications III (Marik, V. and Müller, J. and Pechoucek, M., eds.), vol LNAI 2691, Lecture Notes in Computer Science, pp. 497-511, 2003

BC.02 D. Weyns, E. Truyen, and P. Verbaeten, Serialization of distributed execution-state in Java, Objects, Components, Architectures, Services, and Applications for a Networked World (Aksit, M. and Mezini, M. and Unland, R., eds.), vol 2591, Lecture Notes in Computer Science, pp. 41-61, 2003

BC.01 D. Weyns, and T. Holvoet, Model for simultaneous actions in situated multi-agent systems, Multiagent System Technologies (Schillo, M. and Klusch, M. and Muller, J. and Tianfield, H., eds.), vol 2831, Lecture Notes in Computer Science, pp. 105-119, 2003

Other scientific journals

OJ.11 Matthias Galster, Danny Weyns, Paris Avgeriou, and Martin Becker, Variability in software architecture: views and beyond, ACM SIGSOFT Software Engineering Notes 37, :1-9, 6 2013

OJ.10 Matthias Galster, Paris Avgeriou, Danny Weyns, and Tomi Mannisto (2011) Variability in Software Architecture: Current Practice and Challenges. ACM SIGSOFT Software Engineering Notes, 36(5):30–32, 2011

OJ.09 Matthias Galster, Paris Avgeriou, Danny Weyns, Martin Becker: Second international workshop on variability in software architecture. WICSA/ECSA Companion Volume, 2012

OJ.08 D. Weyns, Guest editors’ introduction, special issue on multiagent systems and software architecture, International Journal on Agent Oriented Software Engineering, 2(1), pp. 1-2, 2008

OJ.07 V. Parunak, and D. Weyns, Guest editors’ introduction, special issue on environments for multi-agent systems, International Journal on Autonomous Agents and Multi-Agent Systems 14 (1), pp. 1-4, Feb. 2007

OJ.06 D. Weyns, An Architecture-Centric Approach for Software Engineering with Situated Multiagent Systems, Abstracts of Recent PhDs, The Knowledge Engineering Review, 22(4), 2007

OJ.05 D. Weyns, K. Schelfthout, T. Holvoet, Architectural design of a distributed application with autonomic quality requirements, ACM SIGSOFT Software Engineering Notes, 30(4), 2005

OJ.04 A. Garcia, R. Choren, C. Lucena, A. Romanovski, H. Giese, D. Weyns, T. Holvoet, and P. Giorgini, Software engineering for large-scale multi-agent systems - SELMAS 2005: Workshop report, ACM SIGSOFT Software Engineering Notes 30 (4), pp. 1-8, July, 2005

OJ.03 D. Weyns, and T. Holvoet, On environments in multi-agent systems, AgentLink Newsletter 16, pp. 18-19, December, 2004

OJ.02 R. Choren, A. Garcia, C. Lucena, M. Griss, D. Kung, N. Minsky, A. Romanovski, J. Castro, R. de Lemos, and D. Weyns, Software engineering for large-scale multi-agent systems - SELMAS 2004: Workshop report, ACM SIGSOFT Software Engineering Notes 29 (5), pp. 1-10, September, 2004

OJ.01 D. Weyns, MATES @ Net.ObjectDays, KI - Zeitschrift Künstliche Intelligenz 04 (1), pp. 74, February, 2004

Authored book

D. Weyns, Architecture-Based Design of Multi-Agent Systems, Springer, 230 p. ISBN: 978-3-642-01063-7, May 2010

Edited volumes

EB.14	D. Weyns, S. Malek, J. Andersson, and B. Schmerl, Special Issue on State-of-the-Art and Research Challenges in Self-Adaptive Systems in the Journal on Systems and Software, vol. 85(12) JSS, 2012
EB.13	D. Weyns and J. Mueller, Special Issue on Challenges in Agent-Oriented Software Engineering in The Knowledge Engineering Review (in press)
EB.12	M.P. Gleizes, D. Weyns, Agent-Oriented Software Engineering XI, Lecture Notes in Computer Science, Springer-Verlag, 2010
EB.11	F. Dechesne, H. Hattori, A. ter Mors, J. Such, D. Weyns, F. Dignum, Advanced Agent Technology - AAMAS 2011 Workshops, Taiwan, Revised Selected Papers. Lecture Notes in Computer Science 7068, Springer 2012
EB.10	D. Weyns, S. Malek, R. de Lemos, J. Andersson, Self-Organizing Architectures, vol. 6090, Lecture Notes in Computer Science, Springer-Verlag, 2010
EB.09	A. Uhrmacher and D. Weyns, Multi-Agent Systems: Simulation and Applications, Taylor and Francis, ISBN 9-781-42007023-1, 2009
EB.08	D. Weyns, H.V.D. Parunak, O. Shehory, Special Issue on Future of Software Engineering and Multiagent Systems and Software Architecture, International Journal on Agent Oriented Software Engineering, vol. 3, number 4, ISSN 1746-1375, 2009
EB.07	D. Weyns, Special Issue on Multiagent Systems and Software Architecture, International Journal on Agent Oriented Software Engineering, vol. 2, number 1, ISSN 1746-1375, 2008
EB.06	D. Weyns, S. Brueckner, and Yves Demazeau, Engineering environment-mediated multiagent systems, vol. 5049, Lecture Notes in Computer Science, Springer-Verlag, 2008
EB.05	D. Weyns, H. Parunak, and F. Michel, Environments for multi-agent systems III, vol. 4389, Lecture Notes in Computer Science, Springer-Verlag, ISBN 3-540-24575-8, 2007
EB.04	H. Parunak, and D. Weyns, Special issue on environments for multi-agent systems, International Journal on Autonomous Agents and Multi-Agent Systems, vol. 14, number 1, ISSN 1387-2532, 2007
EB.03	D. Weyns, and T. Holvoet, Proceedings Multiagent Systems and Software architecture track Net.ObjectDays 2006, ISBN 90-5682-737-5, 2006
EB.02	D. Weyns, H. Parunak, and F. Michel, Environments for multi-agent systems II, vol. 3830, Lecture Notes in Computer Science, Springer-Verlag, ISBN 3-540-32614-6, 2006
EB.01	D. Weyns, H. Parunak, and F. Michel, Environments for multi-agent systems, vol. 3374, Lecture Notes in Computer Science, Springer-Verlag, ISBN 3-540-24575-8, 2005

International conferences and workshops, proceedings

IC.84	D. Weyns and T. Ahmad, Claims and Evidence for Architecture-Based Self-Adaptation: A Systematic Literature Review, European Conference on Software Architecture, ECSA 2013
IC.83	D. Weyns, U. Iftikhar and J. Söderlund, Do External Feedback Loops Improve the Design of Self-Adaptive Systems? A Controlled Experiment, Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2013
IC.82	D. Gil de la Iglesia and D. Weyns, Guaranteeing Robustness in a Mobile Learning Application Using Formally Verified MAPE-K Loops, Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2013

IC.81	D. Gil de la Iglesia and D. Weyns, Software Qualities in MAS using Self-Adaptation (short paper), International Joint Conference on Autonomous Agents and Multi-Agent Systems, AAMAS 2013
IC.80	U. Iftikhar and D. Weyns, A Case Study on Formal Verification of Self-Adaptive Behaviors in a Decentralized System, Foundations of Coordination Languages and Self Adaptation, FOCLASA 2012
IC.79	M. Usman Iftikhar and D. Weyns, Model Checking of Self-Adaptive Behaviors in a Multi-Agent System for Traffic Monitoring, European Workshop on Multi-Agent Systems, EUMAS 2012
IC.78	D. Gil de la Iglesia and D. Weyns, Enhancing Software Qualities in Multi-Agent Systems using Self-Adaptation, European Workshop on Multi-Agent Systems, EUMAS 2012
IC.77	D. Weyns, U. Iftikhar, D. Gil de la Iglesia, and T. Ahmad, A Survey on Formal Methods in Self-Adaptive Systems, Formal Methods for Self-Adaptive Systems, FMSAS 2012
IC.76	D. Weyns, Towards an Integrated Approach for Validating Qualities of Self-Adaptive Systems, ACM Workshop on Dynamic Analysis, WODA 2012
IC.75	N. Abbas, J. Andersson, D. Weyns, Modeling Variability in Product Lines Using Domain Quality Attribute Scenarios, International Workshop on Variability in Software Architecture, VARSA 2012
IC.74	D. Weyns, U. Iftikhar, S. Malek, J. Andersson, Claims and Supporting Evidence for Self-Adaptive Systems: A Literature Study, Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2012
IC.73	D. Weyns, J. Andersson, CAKE: Codifying Architecture Knowledge Effectively, Intl. Software Technology Exchange Workshop, STEW 2011
IC.72	B. Michalek, D. Weyns, A. Helleboogh, N. Boucke, Reconstructing Architectural Models to Support SPL Products Updates: A Controlled Experiment, 5th International Symposium on Empirical Software Engineering and Measurement, ESEM 2011
IC.71	N. Abbas, J. Andersson, D. Weyns, Knowledge Evolution in Autonomic Software Product Lines, 4th International Workshop on Dynamic Software Product lines, DSPL 2011
IC.70	B. Michalik, Danny Weyns, Towards a Solution for Change Impact Analysis of Software Product Line Products, 1st International Workshop on Variability in Software Architecture, VARSA 2011
IC.69	D. Weyns, B. Michalek, A. Helleboogh, N. Boucke, An Architectural Approach for Online Updates of Software Product Lines, 9th Working IEEE/IFIP Conference on Software Architecture, WICSA 2011
IC.68	P. Vromant, D. Weyns, S. Malek, and J. Andersson, On Interacting Control Loops in Self-Adaptive Systems, 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS 2011
IC.67	D. Weyns and B. Michalek, Codifying Architecture Knowledge to Support Online Evolution of Software Product Lines, 6th International Workshop on SHAring and Reusing architectural Knowledge, SHARK 2011
IC.66	B. Michalek, D. Weyns, and W. Van Betsbrugge, On the Problems with Evolving Egemin's Software Product Line, 2nd International Workshop on Product LinE Approaches in Software Engineering, PLEASE 2011
IC.65	D. Tofan, M. Galster, P. Avgeriou, and D. Weyns, Software Engineering Researchers' Attitudes on Case Studies and Experiments: an Exploratory Survey, 15th International Conference on Evaluation and Assessment in Software Engineering, EASE'11
IC.64	D. Weyns, B. Michalik, A. Helleboogh, and N. Boucke, On-demand Generation of Views to Support Online Evolution of Software Product Lines, SEI Architecture Technology User Network Conference, SATURN 2011

IC.63	T. Holvoet, D. Weyns, P. Valckenaers, Delegate MAS Patterns for Large-Scale Distributed Coordination and Control Applications, 15th European Conference on Pattern Languages of Programs, EuroPlop 2010, ACM
IC.62	B. Michalik and D. Weyns, Architecture Query Language Framework, 14th International Software Product Line Conference, SPLC Doctoral Symposium, Jeju Island, South Korea, 13-17 September 2010
IC.61	D. Weyns, S. Malek, J. Andersson, FORMS: a FOrmal Reference Model for Self-adaptation, 7th International Conference on Autonomic Computing and Communications, ICAC, 2010.
IC.60	D. Weyns, S. Malek, J. Andersson, On Decentralized Self-Adaptation: Lessons from the Trenches and Challenges for the Future, Software Engineering for Adaptive and Self-Managing Systems, SEAMS, 2010
IC.59	R. Haesevoets, D. Weyns, M. Henrique Cruz Torres, A. Helleboogh, T. Holvoet, W. Joosen, A Middleware Model in Alloy for Supply Chain-Wide Agent Interactions, 11th International Workshop on Agent-Oriented Software Engineering, AOSE, 2010.
IC.58	D. Weyns, A Pattern Language for Multi-Agent Systems, Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture, WICSA/ECSA 2009
IC.57	T. Holvoet, D. Weyns, P. Valckenaers, Patterns for Delegate Multiagent Systems, 3th IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2009
IC.56	J. Andersson, R. de Lemos, S. Malek, D. Weyns, Reflecting on Self-Adaptive Software Systems, International ICSE Workshop on Software Engineering for Adaptive and Self-managing Systems, SEAMS 2009
IC.55	A. Helleboogh, D. Weyns, K. Schmid, T. Holvoet, K. Schelfthout, W. Van Betsbrugge, Adding Variants on-the-fly: Modeling Meta-Variability in Dynamic Software Product Lines, SPLC Workshop on Dynamic Software Product Lines, DSPL 2009
IC.54	W. Labeeuw, K. Driessens, D. Weyns, T. Holvoet, and G. Deconinck, Prediction of Congested Traffic on the Critical Density Point Using Machine Learning and Decentralised Collaborating Cameras, 14th Portuguese Conference on Artificial Intelligence, EPIA 2009
IC.53	R. Haesevoets, D. Weyns, A. Helleboogh, T. Holvoet, and W. Joosen, Formal Model for Self-Adaptive and Self-Healing Organizations, International ICSE Workshop on Software Engineering for Adaptive and Self-managing Systems, SEAMS 2009
IC.52	D. Weyns, A. Helleboogh, T. Holvoet, Transferring Research into Practice: Experiences in the EMC2 Project, OOPSLA '08 Workshop: Escaped from the Lab: Crossing the Gap from Invention to Practice, 2008
IC.51	D. Weyns, A. Helleboogh, T. Holvoet, K. Schelfthout, and W. Wim Van Betsbrugge, Towards a Software Product Line for Automated Transportation Systems, Proceedings of the 2nd International SPLC Workshop on Dynamic Software Product Lines, DSPL 2008
IC.50	R. Haesevoets, D. Weyns, L. Christiaens, F. Meulders, T. Holvoet, W. Joosen, Hierarchical Organizations for Decentralized Traffic Monitoring, Environment-Mediated Coordination in self-Adaptive and Self-Organizing Systems, ECOSOA 2008
IC.49	D. Weyns, T. Holvoet, K. Schelfthout, J. Wielemans, Decentralized Control of Automatic Guided Vehicles: Applying Multi-Agent Systems in Practice, Development Track OOPSLA, Companion ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, Nashville, USA, 2008
IC.47	A. Helleboogh, D. Weyns, T. Holvoet, and N. Boucke, On ADLs and tool support for documenting view-based architectural descriptions, Fourth SEI Software Architecture Technlogy User Network Workshop, SATURN 2008

IC.46	A. Ricci, E. Platon, F. Ishikawa, D. Weyns, Special track on Agent-oriented Programming, Systems, Languages, and Applications (APSLA): editorial message. SAC 2008
IC.45	D. Weyns, R. Haesevoets, B. Van Eylen, T. Holvoet, and W. Joosen, Endogenous versus Exogenous Self-Management, Proceedings of the International ICSE Workshop on Software Engineering for Adaptive and Self-managing Systems, SEAMS 2008
IC.44	D. Weyns, T. Holvoet, and A. Helleboogh, Anticipatory Vehicle Routing using Delegate Multi-Agent Systems, IEEE International Conference on Intelligent Transportation Systems, ITSC, 2007
IC.43	R. Haesevoets, B. Van Eylen, D. Weyns, A. Helleboogh, T. Holvoet, W. Joosen, Coordinated Monitoring of Traffic Jams with Context-Driven Dynamic Organizations, International Conference on Engineering Environment-Mediated Multiagent Systems, EEMMAS 2007
IC.42	D. Weyns, N. Boucke, and T. Holvoet, DynCNET: A Protocol for Flexible Task Assignment in Situated Multiagent Systems, Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems, SASO 2007
IC.41	D. Weyns, and T. Holvoet, An Architectural Strategy for Self-Adapting Systems, Proceedings of the International ICSE Workshop on Software engineering for Adaptive and Self-managing Systems, SEAMS 2007
IC.40	V. Parunak, S. Brueckner, D. Weyns, T. Holvoet, P. Verstraete, P. Valckenaers, E Pluribus Unum: Polyagent and Delegate MAS Architectures, Proceedings of the 8th International Workshop on Multi-Agent-Based Simulation, MABS 2007
IC.39	D. Weyns, and T. Holvoet, Formal Specification of Laws for Mediating Agents' Activities in Situated Multiagent Systems, Proceedings of the 16th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Interdisciplinary Aspects of Coordination Applied to Pervasive Environments: Models and Applications, CoMA 2007
IC.38	N. Boucké, D. Weyns, and T. Holvoet, View Composition as a First-Class Concept in Architectural Descriptions, Proceedings of the International AOSD Workshop on Aspects in Architecture Descriptions, AARCH 2007
IC.37	D. Weyns, N. Boucké, and T. Holvoet, Gradient field-based task assignment in an AGV transportation system, Proceedings of Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (Stone, P. and Weiss, G., eds.), pp. 842-849, 2006
IC.36	D. Weyns, T. Holvoet, and K. Schelfthout, Multiagent systems as software architecture: another perspective on software engineering with multiagent systems, Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (Stone, P. and Weiss, G., eds.), pp. 1314-1317, 2006
IC.35	D. Weyns, and T. Holvoet, A reference architecture for situated multiagent systems, Proceedings of the Third International Workshop on Environments for Multiagent Systems (Weyns, D. and Paruank, V. and Michel, F., eds.), pp. 1-40, 2006
IC.34	D. Weyns, and T. Holvoet, On the connection between multiagent systems and software architecture, Proceedings of the 7th International Workshop on Engineering Societies in the Agents World (O'Hare, G. and Ricci, A. and O'Grady, G. Dikenelli, O., eds.), pp. 80-85, 2006
IC.33	D. Weyns, and T. Holvoet, Architectural design of an automatic guided vehicle transportation system with a multiagent system approach, Proceedings of the Second Software Architecture Technology User Network (SATURN) Workshop (Nord, D., ed.), CMU/SEI-2006-TR-010, ESC-TR-2006-010, pp. 27-29, 2006
IC.32	D. Weyns, and T. Holvoet, Multiagent systems and software architecture, Multiagent Systems and Software Architecture, Proceedings of the Special Track at Net.ObjectDays 2006 (Weyns, D. and Holvoet, T., eds.), pp. 7-30, 2006

IC.31	D. Weyns, N. Boucké, T. Holvoet, and K. Schelfthout, DynCNET: a protocol for flexible task assignment applied in an AGV transportation system, Proceedings of the Fourth European Workshop on Multi-Agent Systems (Omicini, A. and Dunin-Keplicz, B. and Padget, J., eds.), pp. 359-370, 2006
IC.30	N. Boucké, D. Weyns, K. Schelfthout, and T. Holvoet, Applying the ATAM to an architecture for decentralized control of a transportation system, Proceedings of the 2nd International Conference on the quality of software architectures (QoSA) (Hofmeister, C. and Crnkovic, I. and Reussner, R. and Becker, S., eds.), pp. 1-18, 2006
IC.29	N. Boucké, D. Weyns, and T. Holvoet, Experiences with Theme/UML for architectural design of a multiagent system, Multiagent Systems and Software Architecture, Proceedings of the Special Track at Net.ObjectDays 2006 (Weyns, D. and Holvoet, T., eds.), pp. 87-110, 2006
IC.28	D. Weyns, G. Vizzari, and T. Holvoet, Environments for multiagent systems: Beyond infrastructure, Environments for Multiagent Systems (Weyns, D. and Parunak, V. and Michel, F., eds.), pp. 101-117, 2005
IC.27	D. Weyns, K. Schelfthout, and T. Holvoet, Architectural design of a distributed application with autonomic quality requirements, Proceedings Design and Evolution of Autonomic application Software (Garlan, D. and Litoui, M. and Müller, H. and Mylopoulos, J. and Smith, D. and Wong, K., eds.), pp. 52-59, 2005
IC.26	D. Weyns, K. Schelfthout, and T. Holvoet, Exploiting a virtual environment in a real-world application, Environments for Multiagent Systems (Weyns, D. and Parunak, V. and Michel, F., eds.), pp. 21-36, 2005
IC.25	D. Weyns, N. Boucké, T. Holvoet, and W. Schols, Gradient field-based task assignment in an AGV transportation system, Proceedings of Third European Workshop on Multiagent Systems (Gleizes, M.P. and Kaminka, G. and Nowé, A. and Ossowski, S. and Tuyls, K. and Verbeeck, K., eds.), pp. 447-459, 2005
IC.24	D. Weyns, and T. Holvoet, From reactive robotics to situated multiagent systems: a historical perspective on the role of the environment in multiagent systems, Sixth International Workshop on Engineering Societies in the Agents World (Dikenelli, O. and Gleizes, M.P. and Ricci, A., eds.), pp. 31-56, 2005
IC.23	D. Weyns, K. Schelfthout, T. Holvoet, and T. Lefever, Decentralized control of E'GV transportation systems, Autonomous Agents and Multiagent Systems, Industry Track (Pechoucek, M. and Steiner, D. and Thompson, S., eds.), pp. 67-74, 2005
IC.22	K. Schelfthout, D. Weyns, and T. Holvoet, Middleware for protocol-based coordination in dynamic networks, Proceedings of the 3rd international workshop on Middleware for pervasive and ad-hoc computing (Terzis, S. and Donsez, D., eds.), pp. 1-8, 2005
IC.21	T. Holvoet, D. Weyns, K. Schelfthout, and T. Lefever, Decentralized control of autonomous guided vehicles scalable warehouse systems, The IEE Seminar on Autonomous Agents, ISSN 0537-9989, pp. 11-18, 2005
IC.20	D. Weyns, and T. Holvoet, Situated multi-agent systems for developing self-managing distributed applications, Proceedings of the Doctoral Mentoring Symposium, AAMAS 2004 (Kaminka, G., ed.), pp. 25-26, 2004
IC.19	D. Weyns, K. Schelfthout, T. Holvoet, and O. Glorieux, A role based model for adaptive agents, Proceedings of the AISB 2004, Fourth Symposium on Adaptive Agents and Multi-Agent Systems, pp. 75-86, 2004
IC.18	D. Weyns, E. Steegmans, and T. Holvoet, Combining adaptive behavior and role modeling with state charts, Proceedings of the Third International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (Choren, R. and Garcia, A. and Lucena, C. and Griss, M. and Kung, D. and Minsky, N. and Romanovsky, A., eds.), pp. 81-90, 2004
IC.17	D. Weyns, E. Steegmans, and T. Holvoet, Protocol based communication for situated multi-agent systems, Proceedings of The Third International Joint

	Conference on Autonomous Agents and Multi Agent Systems (Jennings, N. and Sierra, C. and Sonenberg, L. and Tambe, M., eds.), pp. 118-126, 2004
IC.16	D. Weyns, and T. Holvoet, A colored Petri net for regional synchronization in situated multiagent systems, Proceedings of the First International Workshop on Coordination and Petri Nets (Ciancarini, P. and Bocchi, L., eds.), pp. 65-86, 2004
IC.15	D. Weyns, E. Steegmans, and T. Holvoet, Protocol-based communication for situated agents, Proceedings of the Sixteenth Belgium-Netherlands Confernce on Artificial Intelligence (Verbrugge, R. and Taatgen, N. and Schomaker, L., eds.), pp. 303-304, 2004
IC.14	D. Weyns, E. Steegmans, and T. Holvoet, Towards commitments for situated agents, IEEE SMC'2004 Conference Proceedings (Thissen, W. and Wieringa, P. and Pantic, M. and Ludema, M., eds.), pp. 5479-5485, 2004
IC.13	D. Weyns, A. Helleboogh, E. Steegmans, T. De Wolf, K. Mertens, N. Boucké, and T. Holvoet, Agents are not part of the problem, agents can solve the problem, Proceedings of the OOPSLA Workshop on Agent-Oriented Methodologies, 2004 (Gonzales-Perez, C., ed.), pp. 101-112, 2004
IC.12	E. Steegmans, D. Weyns, T. Holvoet, and Y. Berbers, Designing roles for situated agents, The fifth international workshop on agent-oriented software engineering (Odell, J. and Giorgini, P. and Muller, J.P., eds.), pp. 17-32, 2004
IC.11	A. Helleboogh, T. Holvoet, and D. Weyns, Time management adaptability in multi-agent systems, Proceedings of the AISB 2004 Fourth Symposium on Adaptive Agents and Multi-Agent Systems (Kudenko, D. and Alonso, E. and Kazakov, D., eds.), pp. 20-30, 2004
IC.10	A. Helleboogh, T. Holvoet, and D. Weyns, Time management support for simulating multi-agent systems, Joint workshop on multi-agent and multi-agent-based simulation (Davidsson, P. and Gasser, L. and Logan, B. and Takadama, K., eds.), pp. 31-40, 2004
IC.09	N. Boucké, D. Weyns, T. Holvoet, and K. Mertens, Decentralized allocation of tasks with delayed commencement, EUMAS'04 Proceedings (Chiara, G. and Ciorgini, P. and van der Hoek, W., eds.), pp. 57-68, 2004
IC.08	D. Weyns, and T. Holvoet, Synchronous versus asynchronous collaboration in situated multi-agent systems, Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (Rosenschein, J. and Sandholm, T. and Wooldridge, M. and Yokoo, M., eds.), pp. 1156-1158, 2003
IC.07	D. Weyns, E. Steegmans, and T. Holvoet, A model for active perception in situated multi-agent systems, Proceedings of the First European Workshop on Multi-Agent Systems (d'Iverno, M. and Sierra, C. and Zambonelli, F., eds.), pp. 1-15, 2003
IC.06	D. Weyns, and T. Holvoet, Model for situated multi-agent systems with regional synchronization, Concurrent Engineering, Enhanced Interoperable Systems (Jardim-Goncalvas, R. and Cha, J. and Steiger-Garcao, A., eds.), pp. 177-188, 2003
IC.05	D. Weyns, E. Truyen, and P. Verbaeten, Serialization of a Distributed Execution-state in Java, Hauptkonferenz Net.ObjectDays 2002 - 3th International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (Aksit, M. and Mezini, M., eds.), pp. 55-72, 2002
IC.04	D. Weyns, E. Truyen, and P. Verbaeten, Distributed Threads in Java, Informatica (Grigoras, D., ed.), pp. 94-109, 2002
IC.03	D. Weyns, and T. Holvoet, Look, talk and do: a synchronization scheme for situated multi-agent systems, Proceedings of UK Workshop on Multiagent Systems, EUMAS'02 (McBurney, P. and Wooldridge, M., eds.), pp. 1-8, 2002
IC.02	D. Weyns, and T. Holvoet, A colored Petri-net for a multi-agent application, Proceedings of Modeling Object, Components and Agents, MOCA'02 (Moldt, D., ed.), vol 561, DAIMI PB, pp. 121-141, 2002

IC.01 K. Schelfthout, T. Coninx, A. Helleboogh, T. Holvoet, E. Steegmans, and D. Weyns, Agent Implementation Patterns, Proceedings of the OOPSLA 2002 Workshop on Agent-Oriented Methodologies (Debenham, J. and Henderson-Sellers, B. and Jennings, N. and Odell, J., eds.), pp. 119-130, 2002

Demos

D.02 D. Weyns, Kurt Schelfthout, Jan Wielemans, and T. Lefever, EMC^2 AAMAS demo: Decentralized Control of E'GV Transportation System, 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems, Utrecht, July 25-29, 2005 http://emc2.egemin.com/

D.01 D. Weyns, and T. Holvoet, The Packet-World as a case to study Sociality in Multi-Agent Systems, Autonomous Agents and Multi-Agent Systems, AAMAS 2002, Bolgona, Italy, July 15-19, 2002, Università di Bologna and Università di Modena e Reggio Emilia

Publications – Martina Maggio – April 2013

Publication list

Peer-reviewed articles

* [J12] Martina Maggio, Alessandro Vittorio Papadopoulos, and Alberto Leva. “On the use of feedback control in the design of computing system components”. In: *Asian Journal of Control* 15.1 (2013).

[J11] Martina Maggio, Federico Terraneo, and Alberto Leva. “Task scheduling: a control-theoretical viewpoint for a general and flexible solution”. In: *ACM Transactions on Embedded Computing Systems* — accepted for publication. (2013).

[J10] Martina Maggio et al. “Power optimization in embedded systems via feedback control of resource allocation”. In: *Transaction on Control System Technology* 21.1 (2013).

[J9] Martina Maggio, Marco Bonvini, and Alberto Leva. “The PID+p controller structure and its contextual autotuning”. In: *Journal of Process Control* 22.7 (2012).

* [J8] Martina Maggio et al. “Comparison of Decision Making Strategies for Self Optimization in Autonomic Computing Systems”. In: *ACM Transactions on Autonomous and Adaptive Systems* 7.4 (2012).

[J7] Alessandro Vittorio Papadopoulos, Martina Maggio, Sara Negro, and Alberto Leva. “General control-theoretical framework for online resource allocation in computing systems”. In: *IET Control Theory & Applications* 6.11 (2012).

[J6] Alberto Leva and Martina Maggio. “A systematic way to extend ideal PID tuning rules to the real structure”. In: *Journal of Process Control* 21.1 (2011).

[J5] Martina Maggio and Alberto Leva. “Extending Ideal PID Tuning Rules to the ISA Real Structure: Two Procedures and a Benchmark Campaign”. In: *Industrial & Engineering Chemistry Research* 50.16 (2011).

[J4] Alberto Leva, Filippo Donida, and Martina Maggio. “Object-oriented modelling of starch mashing for simulation-based control studies”. In: *Mathematical and Computer Modelling of Dynamical Systems* 16.3 (2010).

[J3] Alberto Leva and Martina Maggio. “Feedback process scheduling with simple discrete-time control structures”. In: *IET Control Theory and Applications* 4.11 (2010).

[J2] Martina Maggio and Alberto Leva. “A new perspective proposal for preemptive feedback scheduling”. In: *International Journal of Innovative Computing, Information and Control* 6.10 (2010).

[J1] Alberto Leva and Martina Maggio. “The PI+p Controller Structure and its Tuning”. In: *Journal of Process Control* 19.9 (2009).

Peer-reviewed conference contributions

[C24] Georgios Chasparis, Martina Maggio, Karl-Erik Årzén, and Enrico Bini. "Distributed Management of CPU Resources for Time-Sensitive Applications". In: *ACC, American Control Conference*. Washingon, District of Columbia, USA, 2013.

[C23] Martina Maggio, Enrico Bini, Georgios Chasparis, and Karl-Erik Årzén. "A Game-Theoretic Resource Manager for RT Applications". In: *ECRTS, European Conference on Real-Time Systems*. Paris, France, 2013.

* [C22] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. "Autotuning control structures for reliability-driven dynamic binding". In: *CDC, Conference on Decision and Control*. Maui, Hawaii, USA, 2012.

[C21] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. "Discrete-time dynamic modeling for software and services composition as an extension of the Markov chain approach". In: *MSC, Multi-Conference on Systems and Control*. Dubrovnik, Croatia, 2012.

* [C20] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. "Reliability-driven dynamic binding via feedback control". In: *SEAMS, Symposium on Software Engineering for Adaptive and Self-Managing Systems*. Zurich, Switzerland, 2012.

[C19] Henry Hoffmann et al. "Self-aware computing in the Angstrom processor". In: *DAC, Design Automation Conference*. San Francisco, California, USA, 2012.

[C18] Alberto Leva, Marco Bonvini, and Martina Maggio. "Object-oriented modelling of industrial PID controllers". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy, 2012.

[C17] Alberto Leva, Martina Maggio, and Federico Terraneo. "Performance analysis of operating systems schedulers realised as discrete-time controllers". In: *MSC, Multi-Conference on Systems and Control*. Dubrovnik, Croatia, 2012.

[C16] Martina Maggio, Federico Terraneo, Alessandro Vittorio Papadopoulos, and Alberto Leva. "A PI-based control structure as an operating system scheduler". In: *IFAC Conference on Advances in PID Control*. Brescia, Italy, 2012.

[C15] Alessandro Vittorio Papadopoulos, Martina Maggio, Francesco Casella, and Johan Åkesson. "Function Inlining in Modelica Models". In: *MATHMOD, Conference on Mathematical Modelling*. Vienna, Austria, 2012.

[C14] Alessandro Vittorio Papadopoulos, Martina Maggio, and Alberto Leva. "Control and Design of Computing Systems: What to Model and How". In: *MATHMOD, Conference on Mathematical Modelling*. Vienna, Austria, 2012.

* [C13] Antonio Filieri, Carlo Ghezzi, Alberto Leva, and Martina Maggio. "Self-Adaptive Software Meets Control Theory: A Preliminary Approach Supporting Reliability Requirements". In: *ASE, Conference on Autonomated Software Engineering*. Lawrence, Kansas, USA, 2011.

[C12] Martina Maggio and Alberto Leva. "Teaching to write control code". In: *IFAC World Congress*. Milan, Italy, 2011.

[C11] Martina Maggio and Alberto Leva. "The PID+p controller structure and its contextual model-based tuning". In: *IFAC World Congress*. Milan, Italy, 2011.

[C10] Martina Maggio et al. "Decision Making in Autonomic Computing Systems: Comparison of Different Approaches and Techniques". In: *ICAC, Conference on Autonomic Computing*. Karlsrhue, Germany, 2011.

[C9] Alessandro Vittorio Papadopoulos, Martina Maggio, Sara Negro, and Alberto Leva. "Enhancing feedback process scheduling via a predictive control approach". In: *IFAC World Congress*. Milan, Italy, 2011.

[C8] Alberto Leva and Martina Maggio. "On the use of models with delay in PI(D) autotuning". In: *CDC, Conference on Decision and Control*. Atlanta, Georgia, USA, 2010.

[C7] Martina Maggio and Alberto Leva. "Toward a deeper use of feedback control in the design of critical computing system components". In: *CDC, Conference on Decision and Control*. Atlanta, Georgia, USA, 2010.

[C6] Martina Maggio et al. "Controlling software applications via resource allocation within the Heartbeats framework". In: *CDC, Conference on Decision and Control*. Atlanta, Georgia, USA, 2010.

[C5] Filippo Sironi et al. "Self-Aware Adaptation in FPGA-based Systems". In: *FLP, Conference on Field Programmable Logic and Applications*. Milan, Italy, 2010.

[C4] Martina Maggio and Alberto Leva. "Object-oriented simulation of preemptive feedback process schedulers". In: *Modelica Conference*. Como, Italy, 2009.

[C3] Martina Maggio et al. "Parallel Simulation of Equation-based Object-Oriented Models with Quantized State Systems on a GPU". In: *Modelica Conference*. Como, Italy, 2009.

[C2] Martina Maggio, Alberto Leva, and Luigi Piroddi. "Closed- versus open-loop active vibration control in the presence of finite precision arithmetic". In: *MSC, Multi-conference on Systems and Control*. San Antonio, Texas, USA, 2008.

[C1] Martina Maggio, Alberto Leva, and Luigi Piroddi. "Finite-precision implementation issues in narrowband active control". In: *CDC, Conference on Decision and Control*. New Orleans, Louisiana, USA, 2007.

Books and book chapters

[B3] Alberto Leva, Martina Maggio, Alessandro Vittorio Papadopoulos, and Federico Terraneo. *Control-based operating system design*. IET, 2013. ISBN: 978-1-84919-609-3.

[B2] Alberto Leva and Martina Maggio. "Model-Based PI(D) Autotuning". In: *PID Control in the Third Millennium* (2012), 45–73, Chapter 2.

[B1] Alberto Leva and Martina Maggio. *Esercizi di Fondamenti di Automatica (Exercises for the course "Foundations of automatic control")*. 2010. ISBN: 978-8-87488-354-7.

Open-access computer programs

- SEPF, Software Emulation Floating Point precision, developed for [C2, C1] `https://github.com/martinamaggio/sefp`.

- APRE, Analyzing the Parameters Relationship to Effectors
  `https://github.com/martinamaggio/apre`.
- GTRM, game theoretic resource manager, developed within [C24]
  `https://github.com/martinamaggio/gtrm`.
- Jobsignaler library, developed for [C24]
  `https://github.com/martinamaggio/jobsignal`.

Kod

þÿ2013-4 63 -10 143-

Name of applicant

Weyns, Daniel

Date of birth

þÿ 80930-1 33

Title of research programme

Appendix N

Budget and research resources

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se

Budget and Research Resources

Justification budget

The following table shows the details of the requested budget.

Kostnader	YEAR 1	YEAR 2	YEAR 3	YEAR 4	YEAR 5	Totalt
Lönekostnader, *Costs for salaries*	714,744	736,186	620,386	638,998	0	2,710,315
Indirekta kostnader, *Indirect costs*	292,330	301,100	253,738	261,350	0	1,108,519
Lokaler, *Premises*	57,180	58,895	49,631	51,120	0	216,825
Resor, *Travel*	30,000	30,000	20,000	20,000	0	100,000
						0
Totalt behov av medel	**1,094,254**	**1,126,182**	**943,755**	**971,468**	**0**	**4,135,659**

The salary costs are calculated for the PI, one PhD student, and one Post Doc. The PI has a full-time contract with Lnu as an associate professor (docent). The budget is based on 30% of his working time over the period of 4 years. In accordance with the normal policy at the Department of Computer Science at Linnaeus University the PhD student will devote 80% of his/her time to the research work within this project and 20% to teaching (the latter is not financed within this project). Lnu will also fund the student's 5th year for finalizing the dissertation. Salary costs are 20% for the Post Doc for year 1 and 2 (Martina Maggio).

The PI will supervise the team and contribute with his expertise in architecture-based self-adaptation and formal modeling and verification (focus on Tasks I and III). The Post Doc will bring in additional expertise in the domain of control theory (focus on Task II). For validation in Task IV, supporting Master degree projects will be launched to support the team.

The personnel costs and overhead are based on the scales currently applicable in Sweden including all social security costs with a small margin for cost increases. Budget for travel is requested to support visits between the team members and visits to conferences and to peer researchers.

Research resources

The following table shows the currently available research resources of the PI, as well as the resources the PI and Post Doc have applied for that are pending.

Type of grant	Applied or granted	Funding source	Grant holder /Project leader	Grant period	Total amount in thousands
Marie Curie CIG	Granted	EU	Danny Weyns	2012-2015	100 kEURO
STREP ICT-2013.1.6	Applied	EU	Alcatel-Lucent Bell NV	2013-2014	1962 kEURO
Improve quality of life for the elderly	Applied	Kamprad Foundation	Danny Weyns	2013-2014	7940 kSEK
Research Grant for Junior Researchers	Applied	VR	Martina Maggio	2014-2017	3600 kSEK
This proposal	Applied	VR	Danny Weyns	2014-2017	4136 kSEK

The Marie Curie CIG resources are used for funding of an on-going PhD student supervised by the PI. The STREP proposal aims for funding of a PhD student of the PI's research team. The Kamprad Foundation proposal aims to fund a Post Doc. The research project described in this proposal connects with these proposals, but complements them by focusing on the combined use of formal modelling/verification and the application of principles from control theory to design feedback loops. The requested funding for the research grant proposal of Martina Maggio relates to this proposal, but focuses on power and temperature control for large-scale computing infrastructures.

VETENSKAPSRÅDET
THE SWEDISH RESEARCH COUNCIL

Kod
2013-45635-105143-33

Dnr
2013 -

Name of applicant
Weyns, Daniel

Date of birth
580930-1533

Reg date
2013-04-11 11:21:06

Project title
Assurances in Decentralized Self-Adaptive Software Systems

NT

NT-2

Research grants NT April 11, 2013
Project Grant Junior Researchers
*Nat-Tek generellt
LINNÉUNIVERSITETET

Appendix S

Signatures

2014	2015	2016	2017	2018
1093	1126	945	971	

2014-01-01 -- 2017-12-31

Applicant

Date

Head of department at host University

Clarifi cation of signature

Telephone

A signature on the application is required not only from the applicant but also from the authorised representative of the administrating organisation/institution or equivalent (normally the head of the department or establishment where the research is to be conducted). The signature confirms that the department can accommodate the proposed research, position or equipment; that the costing in the application is approved for the department's part, that any proposed experimentation on human or animal subjects has been reported, and that the applicant has reported any secondary occupations and commercial ties (s)he may have, and nothing inconsistent with good research practice has thereby emerged. The applicant must have discussed these conditions with the representative of the administrating organisation/institution or equivalent before the latter approves and signs the application.

NOTE the exceptions!
Please note that a different procedure applies for the following types of grants:

- Infrastructure- Operation Grants
- Infrastructure- Grants for Expensive Equipment
- Infrastructure- Grants for Large Databases
- Framework Grant SIMSAM

For these types of grants the vice-chancellor, instead of the head of department, must sign Appendix S. Thereby (s)he confirms the commitments of the administrative organization. The details are specified in the Call for Proposals, published at www.vr.se.

Vetenskapsrådets noteringar
Kod

2013-45635-105143-33

Vetenskapsrådet, Box 1035, SE-101 38 Stockholm, tel. +46 (0)8 546 44 000, vetenskapsradet@vr.se