Ubi-AAL: A Self-Adaptive Ubiquitous Computing System Problem Exemplar in Ambient Assisted Living

Javier Cámara
ITIS Software
Universidad de Málaga
Málaga, Spain
jcamara@uma.es

Ilias Gerostathopoulos
Vrije Universiteit Amsterdam
Amsterdam, Netherlands
i.g.gerostathopoulos@vu.nl

Livia Lestingi
Politecnico di Milano
Milan, Italy
livia.lestingi@polimi.it

Michele Loreti
Università di Camerino
Camerino, Italy
michele.loreti@unicam.it

Hiroyuki Nakagawa
Okayama University
Okayama, Japan
h-nakagawa@okayama-u.ac.jp

Marjan Sirjani
Mälardalen University
Västerås, Sweden
marjan.sirjani@mdu.se

Kenji Tei
Institute of Science Tokyo
Tokyo, Japan
tei@comp.isct.ac.jp

Abstract

Emerging and future applications of ubiquitous computing systems pose unique self-adaptation challenges. To support the research needed to address these challenges, we provide Ubi-AAL, a self-adaptive ubiquitous computing problem exemplar in the domain of assisted living. Ubi-AAL captures key requirements, sources of uncertainty, and other distinguishing characteristics of such applications, such as the tight involvement of human participants in various roles associated with different run-time activities, as well as the interplay of technical goals and human-centric considerations.

CCS Concepts

• **Human-centered computing** → **Ubiquitous computing**; • **Computer systems organization** → **Self-organizing autonomic computing**; • **Applied computing** → *Health care information systems*;

Keywords

Ubiquitous Computing, Uncertainty, Ambient-Assisted Living, Run-time Validation, Autonomy, Self-Adaptation

ACM Reference Format:
Javier Cámara, Ilias Gerostathopoulos, Livia Lestingi, Michele Loreti, Hiroyuki Nakagawa, Marjan Sirjani, and Kenji Tei. 2026. Ubi-AAL: A Self-Adaptive Ubiquitous Computing System Problem Exemplar in Ambient Assisted Living. In *21st International Conference on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '26), April 13–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3788550.3794942

1 Introduction

Self-adaptive ubiquitous computing systems are increasingly deployed in critical domains that require run-time adaptation to changes in dynamic environments affected by multiple sources of uncertainty [Tsigkanos et al. 2025; Weyns et al. 2023]. One such domain is assistive care [Lee and Riek 2018; Napoli et al. 2023], where there is a tight interaction between software, physical elements, and human participants. Unlike other types of self-adaptive systems, self-adaptive ubiquitous systems must continuously balance an array of concerns that include competing technical and human-centric requirements (e.g., ethical, legal).

A fundamental challenge in such systems stems from the close involvement of human participants in multiple roles. Participants are not only passive recipients of assistance, but can also actively participate as monitors, analysts, and decision-makers. They contribute to run-time validation processes through continual feedback (Human-on-the-Loop), and may be required to take actions automated systems cannot perform (Human-in-the-Loop) [Cámara et al. 2017; Cámara et al. 2015]. This deep integration of human and automated system capabilities complicates decision-making, as the system must accommodate varying levels of human engagement and trust while ensuring that adaptation strategies remain effective.

Furthermore, the operation of these systems is governed by a delicate balance between *technical system goals* and broader *socio-technical constraints*. While system designers typically prioritize goals such as safety, availability, cost efficiency, and timeliness, real-world deployments must also adhere to ethical principles, legal regulations, and other social norms—referred to as social, legal, ethical, empathetic, and cultural (SLEEC) requirements [Townsend et al. 2022; Troquard et al. 2024; Yaman et al. 2025]. The interplay between these objectives is potentially conflicting, as optimizing one dimension may come at the expense of another. For example, ensuring strict adherence to safety properties (e.g., avoiding collisions of an assistive mobile robot that is sharing the physical space with a human user) might compromise user privacy because a camera might provide better obstacle detection, compared to, e.g., a lidar-based sensor, unable to capture private images of the user. Such trade-offs must be carefully managed at run time.

Additional complexity comes from the potential misalignment between *local* and *global goals*. In assistive care, system-wide optimization strategies (e.g., minimizing operational costs, maximizing coverage of users receiving service) may conflict with preferences or constraints of individual users (e.g., maximizing quality of the received service). Unlike other fully automated systems where all

SEAMS '26, Rio de Janeiro, Brazil

ACM ISBN 979-8-4007-2445-9/2026/04
https://doi.org/10.1145/3788550.3794942

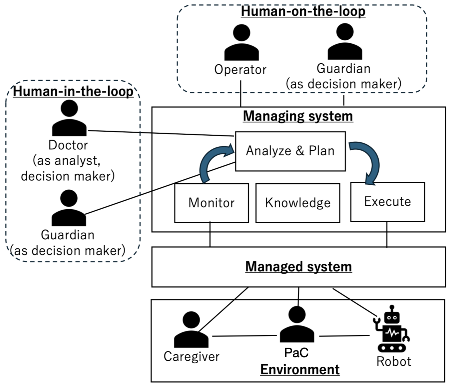

Figure 1: Ubi-AAL's personas mapped to MAPE-K activities.

components operate in unison, ubiquitous self-adaptive systems must accommodate the autonomy of human participants. Users may choose to disregard system recommendations or prioritize personal preferences over the system's prescribed actions. These behavior variations introduce unpredictability and make it challenging to enforce system-wide adaptation strategies effectively.

Several exemplars provide generic frameworks for developing self-adaptive CPS (e.g., [Bures et al. 2013; Krijt et al. 2017]), simulators (e.g., [Maia et al. 2019; Moreno et al. 2019]), and exemplar problems (e.g., [Askarpour et al. 2021; Bennaceur et al. 2016]). Nevertheless, researchers face a scarcity of model problems [uni 2020] to evaluate self-adaptive systems that combine challenges due to: (i) the involvement of humans in multiple roles associated with various activities in adaptation loops, (ii) tradeoffs between technical goals and human-centric requirement compliance, and (iii) potential conflicts and misalignment among local and global goals. This paper addresses this gap by introducing Ubi-AAL[1], a model problem that highlights the critical interdependencies between human actors, technical constraints, and ethical considerations. By defining a problem that incorporates these aspects, we aim to provide an artifact to push forward research in this area.

Analogously to the RoboMAX [Askarpour et al. 2021] and "Feed Me, Feed Me" [Bennaceur et al. 2016] problem exemplars, Ubi-AAL provides high-level requirements, scenarios, contextual information, and adaptation concerns associated with the system considered. Employing this approach, our contribution complements existing SEAMS exemplars, while serving a different purpose.

We provide an overview of Ubi-AAL in Section 2, describe the roles and aims of humans involved in the system in Section 3, define scenarios and requirements in Section 4, which are formalized in Section 5, and finally, discuss key research questions that can arise from this exemplar in Section 6.

2 Ubi-AAL Overview

The Ubi-AAL exemplar focuses on the assistance of individuals with mild cognitive and/or motor impairment. Specifically, care is provided by means of a service robot deployed in the premises and several human subjects that constitute a *circle of support* [Bennaceur et al. 2023] and intervene when needed. Members of the support network have different responsibilities and roles: - family members, - caregivers (e.g., healthcare professionals, social workers), - community members (e.g., neighbours and volunteers). The circle of support must be *resilient*: the *Person at the Centre (PaC)*'s support needs must be met even if the circumstances of a circle member change (e.g., a neighbor moves).

We report an illustrative example. Stella (the PaC) is 78 years old, she is retired and lives alone. She spends a lot of time at home and takes a short walk or visits the market daily. She has two sons and one daughter living in different cities: at least one of them visits her once a month. Stella also has a neighbor, a younger woman who is also retired. Stella has no serious health issues, yet needs to take one pill per day to regulate her blood pressure. At the same time, she feels lonely and is afraid of having an accident at home when she is cooking, cleaning, or showering. Her children decide to set up an ambient-assisted environment in her apartment because they are worried about Stella's mental and physical health. They want to ensure Stella's privacy and use a system conforming to privacy, security, and legal accountability regulations.

Support circles are traditionally managed manually, thus requiring significant effort to orchestrate different institutions distributed geographically and with different contact methods in case of an emergency. The advent of software-intensive systems with state-of-the-art sensor technology can significantly automate these processes, which can be engineered into an ubiquitous computing system. The Internet of Things (IoT) allows instrumenting a PaC's house with an ecosystem of sensors that monitor and analyze their daily activities [Azimi et al. 2017] and connect members of the support network. Recent advances in elderly-machine interaction [Liu 2023]—with the support of AI components—provide customized home assistance accounting for the PaC's mood and physical state.

Nevertheless, the deep connection with behavioral and affective aspects opens up new challenges. The PaC and the support network have different roles and, therefore, different *desiderata*, ranging from functional (e.g., detecting a fall) to SLEEC considerations (e.g., maintaining a certain level of privacy) that must be reconciled to preserve the PaC's physical *and* emotional well-being. We advocate that further research is necessary to blend Human-Robot Interaction (HRI), Artificial Intelligence (AI), IoT, and self-adaptation techniques to achieve this goal.

3 Human roles and aims in Ubi-AAL

The following describes roles and aims of the humans involved in the circle of support (i.e., the *personas* and illustrative user stories). One of the most successful paradigms to date in self-adaptation is MAPE-K [kep 2003], which includes activities to monitor and analyze a software system and its environment, and if the situation demands it, plan and execute adaptations on a managed (sub)system. MAPE-K relies on a knowledge base that can include models of a system's environment, goals, and architecture. Beyond being passive service consumers in a managed system, such as the PaC, humans play various roles that can be mapped to MAPE-K loop activities [Cámara et al. 2017]. Table 1 lists the roles, the main aims,

[1] Link to artifact: https://anonymous.4open.science/r/ubi-aal-F8B2/

Table 1: Roles for human involvement and related user stories in Ubi-AAL.

Role	Type	Description	User stories
PaC	Passive (service consumer)	Person with mild cognitive and/or motor impairment.	- I want to be able to obtain information on a topic (e.g., cooking recipe, weather today) from a robot, via talking to it. - I want to receive assistance to daily chores (when I need it) such as moving a heavy item or cleaning the apartment. - I want to be in charge of my living environment (e.g., not be disrupted by a robot or a sensor). - I want to feel safe at home (e.g., if an accident/health incident happens, my guardians are notified).
Guardian	Monitor/ Decision maker	Relative or acquaintance empowered to make decisions that affect the PaC.	- I want to have real-time visibility on the important metrics of Ubi-AAL (e.g., dashboard, images/video feed). - I want to be notified when there is an ongoing situation that needs to be managed (not necessarily health-related). - I want to be able to approve relevant run time decisions made either by the system and/or doctor. - I want to be able to configure the system/set preferences (e.g., for notifications and other settings).
Doctor	Analyzer/ Decision maker	Medical professional enabled to analyze PaC conditions and make decisions that affect the PaC, jointly with guardians.	- I want to be notified (promptly in case of emergencies) of PaC's health-related situations that need to be managed. - I want to access PaC's health-related information to analyze ongoing situations. - I want to communicate with guardian(s) to decide upon potential course of action.
Caregiver	Actor	Staff specialized in providing care services.	- I want to be notified when the PaC requires assistance. - I want to be involved only when necessary (e.g., not for menial tasks the robot can take care of).
Operator	Actor	Care provider company representative in charge of providing technical support.	- I want to be able to monitor the system and ensure its continuous availability.

and the requirements that the roles demand of the system, while Figure 1 shows the mapping between roles and MAPE-K phases.

Monitors. Monitoring can rely on humans to act as sophisticated sensors and provide information that would be difficult to monitor automatically. For example, *guardians* (cf. Table 1), who are well acquainted with the PaC, can indicate whether there is an ongoing anomaly based on context information that is not captured by the model included in the knowledge base.

Analysts and Decision Makers. Analysis and planning can incorporate into the decision-making process input (e.g. recommendations, validation) from application domain experts who can have additional insight on the best way of managing situations that require system adaptations. In Ubi-AAL, *doctors* can act as analysts by providing insights required for decision making, or even as decision makers by choosing how to manage health-related situations. Humans can be involved in decision making by validating decisions either generated by the system (e.g., the doctor validates a decision made by an automated decision-making component) or by a human decision maker (e.g., the doctor can seek approval from the guardian on a given course of action).

Actors. Execution can employ humans as system-level effectors to execute adaptations when changes to the system cannot be automated, or as a fallback mechanism. In Ubi-AAL, an *operator* can provide technical support, troubleshooting problems that cannot be automatically resolved by the system in the IT infrastructure .

4 From Scenarios to Requirements

From the personas and the user stories, we distill the following basic scenarios setting a baseline for Ubi-AAL's operations, i.e., assuming standard operating conditions:

- BS1. **Obtaining Information**: When the PaC wants to obtain general information (weather condition, etc.), Ubi-AAL offers a speech interface to the robot, which provides the requested information (in different modalities, speech, text, etc.).
- BS2. **Robot cleaning**: Upon the PaC's request, the robot cleans (wipes) the apartment.
- BS3. **Robot help**: Upon the PaC's request, the robot approaches them and helps them move heavy items.
- BS4. **Remote assistance**: When initiated by the PaC, the caregiver, or the system, Ubi-AAL activates a live communication (voice or text) session between them to provide remote assistance.
- BS5. **Threat notification**: In case of a life-threatening situation, Ubi-AAL notifies caregivers, guardians, and/or doctors, depending on the type and level of threat.
- BS6. **PaC Monitoring**: Whenever needed, Ubi-AAL allows caregivers, guardians, and doctors to have an accurate view (and if needed, a detailed raw data view) of the PaC's condition and living environment.
- BS7. **Communication**: Whenever needed, Ubi-AAL allows caregivers, guardians, and doctors to timely and effectively communicate with each other and with the Ubi-AAL.

Ubi-AAL must be equipped with mechanisms to manage runtime uncertainties to avoid brittleness in support of the above scenarios and related *domain concerns*. Therefore, it needs to be designed as a self-adaptive system. Its architecture must thus be enhanced with a **managing system** responsible for handling *adaptation concerns*. We foresee the following adaptation scenarios:

- AS1. **Sensor failure**: When a sensor fails or malfunctions, Ubi-AAL applies countermeasures (e.g., restart, isolate from the rest of the system) and notifies the relevant role
- AS2. **Change in vitals**: When the observed data patterns related to the PaC's vital parameters change (e.g., blood pressure time-series changes), Ubi-AAL starts a diagnosing process to determine the root cause of the change (actual change in blood pressure, miscalibrated sensor or model) and whether any action is needed (e.g., re-training of ML models, notifying others in case of a safety-related detected pattern shift).
- AS3. **Change in environment**: When the observed data related to the living environment change (e.g., blackout, sudden increase in temperature), Ubi-AAL adapts the way it is operating (e.g., sensing the living environment via regular cameras instead of infrared instead).
- AS4. **Delay in action**: If an action is not executed as expected (e.g., it is delayed) by Ubi-AAL, a guardian, a caregiver, or a doctor, the other human(s) in the loop undertake alternative actions to ensure robustness and continuity.
- AS5. **Dangerous robot action**: If the robot performs a potentially dangerous action (e.g., maneuver too close to the PaC), Ubi-AAL immediately stops it, registers the incident, and replans.
- AS6. **Conflicting actions**: If Ubi-AAL detects conflicting actions (e.g., the caregiver overruling the doctor's action/advice), it logs the conflict and attempts to resolve it.

		S1	L1	L2	Et1	Et2	Et3	Et4	Em1	C1
BS1	Obtaining Information			X	X	X				X
BS2	Robot cleaning	X		X						
BS3	Robot help	X		X						
BS4	Remote assistance			X	X	X				
BS5	Threat notification			X					X	
BS6	PaC Monitoring			X		X	X		X	
BS7	Communication			X						
AS1	Sensor failure			X						
AS2	Change in vitals			X				X		
AS3	Change in environment			X				X		
AS4	Delay in action		X	X						
AS5	Dangerous robot action	X		X						
AS6	Conflicting actions			X						

Table 2: Mapping between scenarios and SLEEC concerns.

Scenarios serve as a foundation to elicit requirements, both functional and non-functional (including SLEEC concerns).

Functional Requirements. Functional requirements structure the system's core capabilities. For example, scenario BS2 envisaging the robot cleaning the apartment implies that the robot *must* possess navigation and cleaning capabilities. Being essential, functional requirements are extensively covered in the literature [Iborra et al. 2009; Korchut et al. 2017; Spitalewsky et al. 2013]. Hence, the discussion focuses on *socio-technical* considerations, while functional requirements are covered in the artifact.

SLEEC Requirements. SLEEC requirements govern the interplay between the system's technical goals and the social, ethical, and legal concerns of the environment. We envisage that reconciling the two fronts (technical and socio-technical) represents the most significant hurdle in the development of a system like Ubi-AAL.

We list the concerns that are related to the functional and non-functional scenarios of the previous section. A scenario can have several concerns and a concern can be present in several scenarios. The ID of each concern hints its type: **S**ocial, **L**egal, **Et**hical, **Em**pathetic, **C**ultural.

- **S1** The system does not perform potentially dangerous actions to the PaC unless explicitly asked for.
- **L1** There is a predefined chain of accountability when humans in the loop (i.e. guardian, caregiver, and doctor) fail to execute their assigned actions on time.
- **L2** The system should always provide evidence for its actions.
- **Et1** The speech interface is not activated unless explicitly asked for by the PaC, Guardian, Doctor, or Caregiver.
- **Et2** The listening interface is not activated unless explicitly asked for by the PaC.
- **Et3** The video interface is not activated unless there is an emergency or the PaC authorized it.
- **Et4** All the data of the PaC are handled according to relevant privacy regulations (e.g. GDPR).
- **Em1** The system does not disturb the PaC (e.g. initiating communication) unless there is an urgency.
- **C1** When providing information to the PaC, the robot should not use a language or honorifics that are not appropriate.

5 Formalizing Ubi-AAL Requirements

The Ubi-AAL artifact contains a machine-readable formalization of functional and non-functional requirements of the system grouped by scenarios, as per Section 4. The specification of each scenario also comes with the associated set of SLEEC concerns. Through the LEGOS-SLEEC tool, it is possible to uncover redundancies and conflicts between requirements as well as event sequences that comply with the functional rules but trigger SLEEC concerns [Kolyakov et al. 2025]. The artifact provides a tool-supported exercisable benchmark that exposes the intrinsic complexities and conflicts of a socio-critical ubiquitous system. Its goal is to spur research toward engineering solutions for the highlighted challenges.

Listing 1: LEGOS-SLEEC specification of scenario BS4.

```
def_start
    event CommunicationRequest
    measure submitter: scale(system_s, operator_s, pac_s,
         caregiver_s, guardian_s, doctor_s)
    measure recipient: scale(system_r, operator_r, pac_r,
         caregiver_r, guardian_r, doctor_r)
    measure urgency: scale(no_urgency, routine, urgent, emergency)
    event OpenCommunication
    measure providedInterface: scale(none, sound, text, speech)
    measure quietHours: boolean
    measure validConsent: boolean
    event LogIncident
def_end
rule_start
    r1 when CommunicationRequest then OpenCommunication within 1
         minutes unless (({submitter} = {recipient}) or ({
         providedInterface} = none))
rule_end
concern_start
   Et1 when CommunicationRequest and (({providedInterface} =
        speech) and (not {validConsent})) then OpenCommunication
   Et2 when CommunicationRequest and (({submitter} = pac_s) and
        (({providedInterface} = sound) and (not {validConsent})))
        then OpenCommunication
   L1 when OpenCommunication then not LogIncident within 1 minutes
concern_end
purpose_start
    p1 when CommunicationRequest then OpenCommunication
purpose_end
```

The following reports an excerpt formalization of Ubi-AAL's requirements, including the SLEEC concerns, through the LEGOS-SLEEC tool [Kolyakov et al. 2025]. Through a custom Domain-specific Language (DSL), the tool allows users to code requirements as rules of the form "**when** {condition} **then** {event} **unless** {defeater condition} **then** {event}" [Townsend et al. 2022]. The tool translates the rules to first-order logic constraints subject to SAT solving to identify redundancies, conflicts, and situations potentially triggering SLEEC concerns.

We select scenario BS4 as an illustrative example (see Listing 1). The events are as follows: - a `submitter` (either the system, the PaC, or one of the actors) requests to activate the communication with a `recipient` specifying an `urgency` level; - the system opens the communication and provides an `interface` (either by sound, text, or speech); - the system logs an event for auditing purposes. The specification envisages one functional requirement (rule `r1`): the system must address the communication request promptly unless the system fails to provide an interface or the submitter has mistakenly selected themselves as a recipient. The three concerns that are formalized are raised when the scenario is analyzed with the LEGOS-SLEEC tool. Note that, contrary to their specification at the end of Section 4 as "requirements to be satisfied", concerns in LEGOS-SLEEC are formalized as "negative" cases: rules that should never be triggered. At this point, the analyst can either (i) adapt/enhance the rules so that the concerns are no longer raised, or (ii) accept them as a result of the requirements analysis. For the latter, we expect non-computer scientists to be involved, covering the legal, ethical, and psychological aspects [Kolyakov et al. 2025].

While the scenarios are already informative when analyzed individually, they yield deeper insights when combined in a sequence. A possible combination is: the PaC's vital parameters change, triggering a caregiver notification (AS2), the caregiver then requests the activation of a communication interface (BS4). We have extended the LEGOS-SLEEC tool with an automated procedure for interleaving scenarios, assuming the individual specifications and the necessary inter-scenario "'bridge" rules are provided (see artifact). In the above case, the combination yields a situational conflict: while BS4 requires that communication always starts on request, the combined scenario contains a rule for avoiding communication in quite hours. In this case, the analyst has to resolve the conflict by adapting the rules of the individual scenarios.

6 Research Directions

In this paper, we have provided a number of scenarios formalized in SLEEC for an ambient assisted living (AAL) system, an example of self-adaptive ubiquitous systems. Our artifact provides a starting point for analysis of both SLEEC concerns and potential conflicts. We foresee a number of directions of future research with it:

(1) Requirements engineering: Addressing the currently modeled SLEEC concerns in the different scenarios with the involvement of ethics and legal experts to help navigate the trade-offs.
(2) Software architecture: Use the requirements as a pivot point for system design and implementation. The result would be a technical exemplar that can be used for comparing different self-adaptation approaches.
(3) Runtime monitoring: Create runtime monitors for the requirements that will allow us to have an up-to-date view over their satisfaction at runtime. If some requirements are not satisfied, this signals that the system needs to be evolved.
(4) Probabilistic model-checking: Use the requirements at runtime to support runtime monitoring and verification, and runtime decision making based on SLEEC concerns.

Acknowledgments

This work is a result of Shonan Meeting No. 190, and the authors would like to express their sincere gratitude to the meeting organizers Christos Tsigkanos, Carlo Ghezzi, and Zhenjiang Hu.

References

2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.

2020. What is a Model Problem? https://www.hpi.uni-potsdam.de/giese/public/selfadapt/what-is-a-model-problem/.

Mehrnoosh Askarpour, Christos Tsigkanos, Claudio Menghi, Radu Calinescu, Patrizio Pelliccione, Sergio García, Ricardo Caldas, Tim J von Oertzen, Manuel Wimmer, Luca Berardinelli, Matteo Rossi, Marcello M. Bersani, and Gabriel S. Rodrigues. 2021. RoboMAX: Robotic Mission Adaptation eXemplars. In *Intl. Symp. on Software Engineering for Adaptive and Self-Managing Systems*. 245–251.

Iman Azimi, Amir M. Rahmani, Pasi Liljeberg, and Hannu Tenhunen. 2017. Internet of things for remote elderly monitoring: a study from user-centered perspective. *J. Ambient Intell. Humaniz. Comput.* 8, 2 (2017), 273–289.

Amel Bennaceur, Ciaran McCormick, Jesús García Galán, Charith Perera, Andrew Smith, Andrea Zisman, and Bashar Nuseibeh. 2016. Feed me, feed me: an exemplar for engineering adaptive software. In *Intl. Symp. on Software Engineering for Adaptive and Self-managing Systems*. 89–95.

Amel Bennaceur, Avelie Stuart, Blaine A. Price, Arosha K. Bandara, Mark Levine, Linda Clare, Jessica Cohen, Ciaran McCormick, Vikram Mehta, Mohamed Bennasar, Daniel Gooch, Carlos Gavidia-Calderon, Anastasia Kordoni, and Bashar Nuseibeh. 2023. Socio-Technical Resilience for Community Healthcare. In *International Symposium on Trustworthy Autonomous Systems*. ACM, 26:1–26:6.

Tomas Bures, Ilias Gerostathopoulos, Petr Hnetynka, Jaroslav Keznikl, Michal Kit, and Frantisek Plasil. 2013. DEECO: an ensemble-based component system. In *Intl. ACM Sigsoft Symposium on Component-Based Software Engineering (CBSE '13)*. Association for Computing Machinery, New York, NY, USA, 81–90.

Javier Cámara, David Garlan, Gabriel A Moreno, and Bradley Schmerl. 2017. Evaluating trade-offs of human involvement in self-adaptive systems. In *Managing Trade-Offs in Adaptable Software Architectures*. Elsevier, 155–180.

Javier Cámara, Gabriel A. Moreno, and David Garlan. 2015. Reasoning about Human Participation in Self-Adaptive Systems. In *Intl. Symp. on Software Engineering for Adaptive and Self-Managing Systems*. IEEE Computer Society, 146–156.

Andres Iborra, Diego Alonso Caceres, Francisco J Ortiz, Juan Pastor Franco, Pedro Sanchez Palma, and Barbara Alvarez. 2009. Design of service robots. *IEEE Robotics & Automation Magazine* 16, 1 (2009), 24–33.

Kevin Kolyakov, Lina Marsso, Nick Feng, Junwei Quan, and Marsha Chechik. 2025. LEGOS-SLEEC: Tool for Formalizing and Analyzing Normative Requirements. In *Intl. Conf. on Software Engineering*. IEEE, 33–36.

Agnieszka Korchut, Sebastian Szklener, Carla Abdelnour, Natalia Tantinya, Joan Hernández-Farigola, Joan Carles Ribes, Urszula Skrobas, Katarzyna Grabowska-Aleksandrowicz, Dorota Szczęśniak-Stańczyk, and Konrad Rejdak. 2017. Challenges for service robots—requirements of elderly adults with cognitive impairments. *Frontiers in neurology* 8 (2017), 228.

Filip Krijt, Zbynek Jiracek, Tomas Bures, Petr Hnetynka, and Ilias Gerostathopoulos. 2017. Intelligent Ensembles - A Declarative Group Description Language and Java Framework. In *Intl. Symp. on Software Engineering for Adaptive and Self-Managing Systems*. 116–122.

Hee Rin Lee and Laurel D. Riek. 2018. Reframing Assistive Robots to Promote Successful Aging. *J. Hum.-Robot Interact.* 7 (2018), 1–23.

Jingli Liu. 2023. Design of Rural Home Smart Pension Monitoring System Based on Artificial Intelligence. In *Intl. Conf. on Management Science and Software Engineering*. Atlantis Press, 828–835.

Paulo Henrique Maia, Lucas Vieira, Matheus Chagas, Yijun Yu, Andrea Zisman, and Bashar Nuseibeh. 2019. Dragonfly: a Tool for Simulating Self-Adaptive Drone Behaviours. In *Intl. Symp. on Software Engineering for Adaptive and Self-Managing Systems*. 107–113.

Gabriel Moreno, Cody Kinneer, Ashutosh Pandey, and David Garlan. 2019. DARTSim: An Exemplar for Evaluation and Comparison of Self-Adaptation Approaches for Smart Cyber-Physical Systems. In *Intl. Symp. on Software Engineering for Adaptive and Self-Managing Systems*. 181–187.

Claudia Di Napoli, Giovanni Ercolano, and Silvia Rossi. 2023. Personalized home-care support for the elderly: a field experience with a social robot at home. *User Model. User Adapt. Interact.* 33, 2 (2023), 405–440.

Katharina Spitalewsky, Justine Rochon, Matthias Ganzinger, and Petra Knaup. 2013. Potential and requirements of IT for ambient assisted living technologies. *Methods of information in medicine* 52, 03 (2013), 231–238.

Beverley A. Townsend, Colin Paterson, T. T. Arvind, Gabriel Nemirovsky, Radu Calinescu, Ana Cavalcanti, Ibrahim Habli, and Alan Thomas. 2022. From Pluralistic Normative Principles to Autonomous-Agent Rules. *Minds Mach.* 32, 4 (2022), 683–715.

Nicolas Troquard, Martina De Sanctis, Paola Inverardi, Patrizio Pelliccione, and Gian Luca Scoccia. 2024. Social, Legal, Ethical, Empathetic, and Cultural Rules: Compilation and Reasoning. In *Proc. of AAAI'24 / IAAI'24 / EAAI'24'*. AAAI Press, 22385–22392.

Christos Tsigkanos, Carlo Ghezzi, and Zhenjiang Hu. 2025. Engineering Dependable Ubiquitous Systems. NII Shonan Meeting No.190. https://shonan.nii.ac.jp/seminars/190/.

Danny Weyns, Radu Calinescu, Raffaela Mirandola, Kenji Tei, Maribel Acosta, Amel Bennaceur, Nicolas Boltz, Tomás Bures, Javier Cámara, Ada Diaconescu, Gregor Engels, Simos Gerasimou, Ilias Gerostathopoulos, Sinem Getir Yaman, Vincenzo Grassi, Sebastian Hahner, Emmanuel Letier, Marin Litoiu, Lina Marsso, Angelika Musil, Juergen Musil, Genaína Nunes Rodrigues, Diego Perez-Palacin, Federico Quin, Patrizia Scandurra, Antonio Vallecillo, and Andrea Zisman. 2023. Towards a Research Agenda for Understanding and Managing Uncertainty in Self-Adaptive Systems. *ACM SIGSOFT Softw. Eng. Notes* 48, 4 (2023), 20–36.

Sinem Getir Yaman, Pedro Ribeiro, Ana Cavalcanti, Radu Calinescu, Colin Paterson, and Beverley A. Townsend. 2025. Specification, validation and verification of social, legal, ethical, empathetic and cultural requirements for autonomous agents. *J. Syst. Softw.* 220 (2025), 112229.