Fostering agent cooperation in AmI: a context-aware mechanism for dealing with multiple intentions

Arthur Casals, Assia Belbachir, Amal El Fallah-Seghrouchni, Anarosa Alves Franco Brandão

Fostering agent cooperation in AmI: a context-aware mechanism for dealing with multiple intentions

Arthur Casals and Assia Belbachir and Amal El-Fallah Seghrouchni and Anarosa Alves Franco Brandão

Abstract Ambient Intelligent (AmI) environments dynamically provide contextual information to intelligent agents that interact with them. In such environments, could these agents cooperate to improve their goal achievement, considering multiple intentions from several agents? With multiple agents, cooperation will depend on each agent's own intentions. Agents adapt to dynamic changes in the environment using context-aware planning mechanisms such as the Contextual Planning System (CPS), which proposes an optimal plan for a single agent based on the current context. In this paper we present the Collective CPS (CCPS), an opportunistic cooperative planning mechanism for multiple agents in AmI environments. CCPS allows agents to partially delegate their own plans or to collaborate with other agents' plans during their execution, while retaining individual planning capabilities. A working scenario is shown for a realistic AmI environment, such as a smart Campus.

1 Introduction

Ambient Intelligence (AmI) is a reference to electronic environments in which electronic devices or systems can perceive and respond to the presence of people, while also being able to communicate with each other [1]. In such environments, cooperation and coordination among different systems may exist in different forms, and

Arthur Casals
Escola Politécnica - Universidade de São Paulo, SP, Brazil, e-mail: arthur.casals@usp.br

Assia Belbachir
IPSA, Paris, France, e-mail: assia.belbachir@ipsa.fr

Amal El-Fallah Seghrouchni
Sorbonne Université, LIP6- CNRS UMR 7606, Paris, France e-mail: amal.elfallah@lip6.fr

Anarosa Alves Franco Brandão
Escola Politécnica - Universidade de São Paulo, SP, Brazil, e-mail: anarosa.brandao@usp.br

for different purposes. This paper presents an approach to cooperatively deal with multiple intentions for a specific type of intelligent agent that interacts with AmI environments, denominated ambient agent (AA). The cooperative model in place is opportunistic, i.e., cooperation may exist if there are favorable conditions for it to occur. Since AmI environments provide dynamic contextual information, the AA's reasoning process must adapt to environment changes as efficiently as possible, while still making it possible for the AA to achieve its goals.

Adopting Multiagent Systems (MAS) to deal with contextual information in AmI applications is interesting because of its very nature, where different entities autonomously interact in a dynamic and uncertain environment [3, 15]. MAS whose agents follow the BDI model are specially tailored for AmI due to its inherent use of contextual information in the form of *beliefs*. These agents, however, may have different and independent goals, and still need cooperation to perform specific tasks. This can be illustrated in a real-world situation: if a person has an appointment but cannot be there in time, getting a ride from someone will make this goal achievable. Other similar AmI-related situations include cooperative collision warnings in vehicular ad-hoc networks [8] and opportunistic problem-solving using ontology negotiation [2].

While extensive research has been conducted in the field of intelligent agent cooperative planning [7, 18, 10, 13], most of the existing work relies on central coordination or communication mechanisms. Such restrictions can make it difficult to use the existing research in AmI scenarios where agents possess individual goals, but cooperation without previous commitment is still beneficial. With that in mind, we propose a cooperative planning mechanism that can be used by an AA to delegate specific plans to other agents. The objective is to provide cooperation in each of the agents' planning processes without having to commit to common goals, or make use of central planning engines of any kind. In terms of scientific contribution, this work also makes it possible to deal with multiple intentions while maintaining the collaboration mechanism among intelligent agents strictly opportunistic. In addition, we also improve the BDI agent reasoning mechanism by using a preemptive planning mechanism in conjunction with contextual planning. Thus, the proposed mechanism makes possible for different agents to cooperate among themselves to achieve goals otherwise unachievable by delegating the execution of specific plans. It is important to notice that the cooperation mechanism does not assume benevolent agents, nor it is based in existing commitments of any sort: AAs only cooperate among themselves if it is on their best interests.

This paper is organized as follows: Section 2 introduces relevant aspects of planning and cooperation used along this paper, with a description of the contextual planning system (CPS [6]). Section 3 describes the Collective CPS (CCPS) by presenting its formal structure and related algorithms. Section 4 presents, as proof of concept, an implementation of CCPS in a real-world based scenario, and describes the experiments performed. Finally, Section 5 concludes this paper and presents some perspectives for future work.

2 Background

In this section we present an overview of planning and cooperation in the domain of BDI agents, followed by the CPS, which is the contextual planning mechanism used as a basis for the present work.

2.1 Planning and Cooperation

The planning process used by a BDI agent involves choosing a set of actions to be executed and it is triggered whenever new information is perceived. Re-planning over new information allows the agent to adapt its plans accordingly and re-evaluate its goals at each interaction, so it can adopt the best course of action at the time.

Environmental changes can occur independently from the agent (such as the natural passage of time), or as a consequence of an executed action. One agent's actions can impact the perceptions of other agents in the same environment. If agents have different goals, one agent's actions may cause another agent's goals to become unachievable. Using coordination mechanisms can mitigate these impacts [9], and eventually increasing the overall planning efficacy or efficiency in face of a dynamic environment [12, 14]. It may also be the case that coordinated planning is a requirement for solving the problem at hand.

While there exist solutions aimed at solving situations that require cooperative planning processes [7, 18, 10], they usually consider joint goals situations, either relying on central coordination systems or being limited by the problem domain. AA could benefit from cooperative planning in situations where individual efficiency is affected by environmental conditions, ultimately executing certain tasks more efficiently. If multiple agents were to cooperate, context-dependent tasks could be delegated among them, and otherwise unachievable goals could be successfully achieved. Our approach aims at exploring situations where contextual-dependent tasks may be delegated by one AA as part of its planning process, giving the coordination process a contextual dynamic characteristic. We will detail this approach in the following paragraphs, from the formalism involved to the planning mechanism.

2.2 Contextual Planning System (CPS)

In the original BDI-agent model, an interpreter manages the agent states related to its beliefs (B), desires (D), and intentions (I) in order to achieve goals through planning. Let $\mathbb{A} = \{A_1, A_2, ..., A_n\}$ be the set of agents and $A_i \in \mathbb{A}$.

Chaouche *et. all* [6] proposed a model to deal with multiple intentions for an agent A_i. It represents an agent plan $\tilde{P}(A_i)$ as a tree structure composed of a set of intention plans ($\hat{P}_j$) and elementary plans ($P_{l,k}$). $\tilde{P}(A_i)$ is composed of multiple

intention plans $\hat{P}_j$, and each intention plan corresponds to the achievement of an agent's intention.

An intention plan $\hat{P}_j$ can be an alternate of multiple elementary plans: $\hat{P}_j = \{P_{j,0}, ..., P_{j,q}\}$. Alternating elementary plans allows achieving an specific intention in different manners, each one expressed by an elementary plan.

Elementary plans $P_{j,k}$ are described by *AgLOTOS expressions* [4], referring to behaviour expressions $E_{i,k}$ expressed as an ordered finite set of observable actions a_i to be executed by the agent. Any AgLOTOS expression is associated with contextual information related to the (current) BDI state of an agent. These elementary plans are obtained from a library of plans (*LibP*).

Using this model, Chaouche *et al.* [5] proposed a predictive planning mechanism that uses the contextual information to verify which among the actions known by the agent are feasible. "Feasibility" of an action is determined by any conditions the agent must met to execute it. For instance, moving from point A to point B is only feasible if agent is at point A. Once this verification occurs, CPS selects an execution path that maximizes the number of satisfied intentions [5], producing an optimal plan $\tilde{P}(A_i)$. In the next section we will explain how this plan is used by the Collective Contextual Planning System (CCPS) in conjunction with its cooperative mechanism.

3 Collective Contextual Planning System (CCPS)

The Collective Contextual Planning System (CCPS) structure is designed to process plans from multiple CPS-capable agents. The novelty here, in comparison with the CPS mechanism, is that when an agent plan cannot be executed for any reason, the agent can ask for help and delegates part of this plan to other agents, avoiding re-planning. By doing this, the agent eliminates the problem while maintaining the consistency of its goals. At the same time, all the parallelism and concurrence taken into consideration by the original CPS mechanism are maintained.

In the first step of the CCPS mechanism, an agent has an initial plan resulting from the CPS mechanism execution and verifies if it can be executed without the help of any other agent. If that is not the case, the agent detects which actions are preventing the plan from being executed. These actions are listed in a help request message that is sent to other agents. After receiving this message, each agent decides about offering help for each action and reply its answer. The requester agent and decides about accepting help and inform the helper agent its decision. In the case where no other agent is able to provide help, the agent tries to replace the intention associated with the plan that cannot be executed with a different one that could provide similar results. If no such intention is found, then the original intention is removed.

The idea behind this mechanism is similar to the contract net protocol [16], designed for distributed problem-solving. Aspects related to connection and contract negotiation, however, are simplified - either by considering that it is provided by the

agent's environment or by taking the opportunistic aspect into account. The concepts involved in CCPS will be formalized and explained in the next paragraphs.

3.1 Formal framework

In this section, we introduce some of the definitions necessary to formally describe the CCPS and its algorithm.

- Environment: $Env(t)$ is a set of logical propositions representing contextual information that can be perceived by any agent at time t.
- Action: An action a represents the finest granular activity performed by the agent in the environment. If an agent A_i moves from point X to point Y, there is an associated action $move(X,Y)$. Each action a is subject to three sets of logical propositions (as in STRIPS [11]): preconditions list ($Pre(a)$); delete list ($Del(a)$); and post-conditions list ($Post(a)$). An action a if *feasible* if all of its preconditions $Pre(a)$ are satisfied by the conditions of $Env(t)$. After an action a is executed, its conditions can be changed by $Del(a)$ or $Post(a)$- also altering $Env(t)$.
- Intention: Each agent $A_j \in \mathbb{A}$ has a set of intentions, and each intention is associated with a weight factor that serves as a comparator index, used to determine *order of importance* between two intentions. Thus, the intention function over the known agents is defined as $I : \mathbb{A} \to \mathbb{I}x\mathbb{W}$, and the set of intentions for an agent is expressed by:

$$I(A_j) = \{(i_0, w_0), (i_1, w_1), ..., (i_n, w_n)\}, A_j \in \mathbb{A}, i_0, ..., i_n \in \mathbb{I}, w_0, ..., w_n \in \mathbb{N}^*$$

  Two intentions are comparable according to their associated weight factors: given $(i_p, w_p), (i_q, w_q) \in I(A_j), 0 \leq p, q \leq n$, i_p is at least as important as i_q if $w_q \leq w_p$.
- Plan: Plan definition follows exactly the one from CPS, presented at 2.2.
- Plan Feasibility: Being a composite and partially ordered set of actions, the feasibility of an elementary plan $P_{i,j}$ can be given by the feasibility of its actions: $P_{j,k}$ is feasible in $Env(t)$ if and only if all of its actions are feasible in $Env(t)$. An intention plan $\hat{P}_j$ is feasible if and only if at least one of its elementary plans is feasible. In the case of an agent plan $\tilde{P}(A_i)$, we determine that it is completely feasible if and only if all of its intention plans are feasible.

There might be situations where it is necessary to consider partial feasibility of an agent plan $\tilde{P}(A_i)$. Suppose that $\tilde{P}(A_i)$ contains a set of intention plans $\{\hat{P}_1, ..., \hat{P}_q\}$ referring to $I_j = \{(i_0, w_0), (i_1, w_1), ..., (i_j, w_j)\}, I_j \subseteq I(A_i)$. In this case, if only $\hat{P}_j$ is unfeasible and w_j is relatively much lesser than the other intentions' weights, it could make sense for the plan $\tilde{P}(A_i)$ to be executed anyway, while the intention I_j is still unfeasible. However, if we have a situation where w_j is relatively much greater than the other intentions' weights, we should not want to proceed with $\tilde{P}(A_i)$ at all, and re-planning becomes necessary.

Let us also consider the following notations: the average weight of all intentions associated with all intention plans in $\tilde{P}(A_i)$ is given by $\overline{w_I}$, and the average weight of all intentions associated with all *feasible* intention plans in $\tilde{P}(A_i)$ is given by $\overline{w_F}$. With that in mind, we define the concept of partial feasibility for an agent plan: $\tilde{P}(A_i) = \{\hat{P}_1, ..., \hat{P}_j\}$ is partially feasible if $\overline{w_F} \geq \overline{w_I}$. In the rest of this work, we will consider an agent plan $\tilde{P}(A_i)$ unfeasible if it is not partially feasible. In the same manner, a plan $\tilde{P}(A_i)$ will be considered feasible if it is at least partially feasible.

It is also important to formalize a few aspects related to agent communication - in particular, the aspects involving the exchanged messages between agents, and their possible effects on the CCPS mechanism. When an agent A_j requests help to other agents, this is done through a message M containing at least one unfeasible action selected from $\tilde{P}(A_j)$. This message also contains *temporal limits* and a *deadline*. Temporal limits are associated with actions, and are formed by two different limits: a beginning (when an action should start), and an end (when an action should be finished). The deadline, on the other hand, is associated with the message and it is the time limit for the message to be answered. Once the deadline expires, the agent won't expect any replies to that particular message.

The concepts of complementary and equivalent intentions are also important to the CCPS mechanism since they are related to the cooperative process. Suppose that there are no other agents present in the environment, or no other agent can provide help. In this case, the agent asking for help tries to find any *complementary and equivalent intentions* that may exist within its original set of intentions. Complementarity and equivalence are defined considering that different intentions can have similar or equal results when their plans are executed. We consider that two intentions are complementary if at least one precondition of one of the intentions corresponds to a post-condition of the other. Additionally, two intentions are equivalent if at least one precondition of these intentions is identical, and if all the post-conditions of both intentions are the same.

4 Experiments

To verify the proper functioning of the CCPS structure, we implemented a proof-of-concept (PoC) version of the mechanism and tested it in conjunction with an application scenario. The CCPS PoC mechanism used the algorithm previously described and was implemented in Java. All interactions between the agents were simulated with the use of the JADE framework[1]. This implementation was evaluated according to a formal experimentation process appropriate for case studies in the Software Engineering domain [19]. The study protocol used in this evaluation was: (*i*) Using the concept of opportunistic cooperation as a basis, create a scenario with predictable outcomes; (*ii*) Defining the situations in which the scenario would be executed; (*iii*) Defining the theoretical outcome of the scenario for each of the situations; (*iv*)

---

[1] http://jade.tilab.com

Running the scenario for each of the situations, using the PoC implementation; (*v*) Analyzing the results, comparing the obtained data with the expected outcomes for each situation; and (*vi*) Discussing the results, pointing out any relevant issues or points of attention regarding the evaluation and its elements.

The objective of the evaluation was to observe if the CCPS mechanism could properly identify and act on situations involving *opportunistic cooperation*. Hence, we decided to study a specific scenario considering (*i*) isolated agents, with no communication in-between, and (*ii*) agents that are able to communicate and delegate individual tasks to each other. We used the CCPS formalization as a frame of reference to determine which would be the behavior in both situations, so that we could compare the outcome of the experiments with a theoretical baseline. By running the same scenario under different contextual conditions (simulating a dynamic environment), we wanted to compare how the use of the newly implemented opportunistic cooperation mechanism would affect the performance of each agent in terms of reaching individual goals. Our expectation was that the use of the CCPS mechanism would allow the agents to reach their individual goals more frequently when exposed to different environment conditions.

4.1 Scenario

To evaluate the CCPS mechanism in the domain of Ambient Intelligence (AmI), we chose to model a scenario related to Smart Cities, as described by Streitz [17]. According to this author, one of the goals of designing smart cities should be "enabling people to experience everyday life and work". Achieving this goal could be done through the use of location-based services and communication infrastructure used to connect different citizens among themselves and the city.

Having these concepts in mind, we based our scenario in a situation involving four different agents: Alice, Bob, Claire, and Damien. All agents play different roles within the campus of a University. Some of the agents could cooperate to achieve their individual intentions, such as retrieving a specific resource (book) or getting to a specific location (parking) in a specific time (taking the bus respecting the bus schedule). Cooperation among the agents was passively stimulated with the use of contextual restrictions, such as means of transportation, the agents' initial locations, temporal restrictions (duration of different actions, and time limits in which they should be performed), and so on. The scenario location was modeled after a university campus, and possesses different locations: a laboratory, three different offices, a parking lot, a bus stop, and the residence of all agents. Each of the agents possess the following routines (used to model their intentions): (*i*) *Morning preparation* (actions that take place before each agent leaves their residence); (*ii*) *Going to the university*; (*iii*) *Going to work*; and (*iv*) *Going back home*. Also, individual conditions were attributed to the agents (Alice and Claire have cars, but Bob and Damien take the bus; Alice and Bob work in the same office; and a few others).

4.2 Situations and expected outcomes

The scenario described above was executed in three different environment conditions (contexts), considering starting time (enough time to achieve time-sensitive intentions), inherently unachievable intentions, and the presence of equivalent intentions. Considering the aforementioned experiment situations, the expected results in the next paragraphs refer to the situations where communication between the agents is possible. The situation where the agents cannot communicate with each other was used as a reference to measure the efficacy of the implementation.

In the first context (1), the scenario started at 8 a.m., and each agent was given a set of intentions to be evaluated and eventually achieved along the day (e.g., going to work, working, coming back home). These intentions did not necessarily require cooperation of agents, and the temporal restrictions related to all actions involved were set in a manner that cooperation was not necessary at any part of the planning process. In this context, we expected that none of the agents would ask for any help.

In the second context (2), the starting time was kept at 8 a.m., but a few unsatisfiable intentions were also included (e.g., asking a book to the agents when none of them have it), with no equivalent or complementary intentions that could be used. In the case of the satisfiable intentions, in some cases we included equivalent intentions to reach the same goal (going back home by bus or by a ride in another agent's car). In this context, we expected that the agents with unsatisfiable intentions to ask the other agents for help; however, since they were designed to be absolutely unfeasible, we expected the agents to ask for help and - in the absence of any replies - to try and find any equivalent intentions that could be used. Ultimately, we expected that the agents to discard the unfeasible intentions and re-plan using the remaining intentions.

In the third context (3), we started the scenario at 9 a.m., but included unfeasible intentions with equivalent feasible intentions. This context was supposed (*i*) to allow opportunistic cooperation to effectively take place and (*ii*) to allow the agents to properly use the equivalent intention mechanism. In this context, we expected the agents to use more time in the planning process, but also to achieve more goals (compared to the non-communicating situation).

4.3 Results

Once properly coded and parametrized within the implementation, the scenario described in section 4.1 was executed, and the results were compared with the expected outcomes for each context.

In the first experiment (context 1), the agents behaved as expected. Both the non-communicating and the communicating agents were able to generate achievable plans for all of their intentions using the same amount of time. Our conclusion was that the verification involved in checking if cooperation was needed had no impact in the computational resources required.

In the second experiment (context 2), the communicating agents spent considerable additional time (about 200% more) trying to ask for help and trying to find an equivalent intention before reaching the same plan as the non-communicating agents. However, the outcome was still consistent with the expected results.

In the third experiment (context 3), the communicating agents also took considerably more time to reach a final plan, but in this case their plans also had much more achievable intentions than the non-communicating agents. In some cases, where all intentions were dependent on the achievability of a single one (duration and temporal limit for reaching specific locations), some non-communicating agents had considerable less intentions included in their plans, since the others were simply not achievable (as expected). On the other hand, the plan created by the communicating agents, while feasible, took a considerable amount of time to be generated.

5 Conclusion and future work

In this paper we formally presented CCPS, a cooperative contextual planning mechanism to be used by AA in AmI scenarios. CCPS was designed reusing an existing contextual planning mechanism (CPS) in conjunction with cooperation based on delegation. The resulting structure makes it possible for an agent to achieve goals otherwise unachievable through the cooperation with other agents running in the same environment. We were able to properly evaluate the CCPS mechanism by simulating a specific scenario in different contexts, using a proof-of-concept implementation and comparing the obtained results with the theorized expected situations. While we were able to confirm the proper functioning of the mechanism, there are a few considerations that must be taken into account.

The first one is related to efficiency: we observed that taking advantage of opportunistic cooperation situations also drastically increased the planning process time. In a real-world situation, this could negatively impact the goal achievement success rates of the agents using the CCPS mechanism. This situation is not completely unexpected, however, and we expect to address in future research.

Another consideration related to the experiments performed was related to the communication process. When the CCPS mechanism finds an unfeasible intention, the communication process that takes place is simple, but not optimal. I a worst-case scenario, the environment might be flooded by help requests. We also intend to study this limitation by testing different constraint mechanisms.

Finally, there are multiple aspects yet to be explored in the proposed CCPS mechanism. Different parts of its algorithm can be detailed studied, and concepts such as complementary and equivalent intentions can be refined. Implementation can also be extended to support multiple scenario variations in runtime, and more complex scenarios can be used to verify how the CCPS mechanism performs under different conditions. We intend to explore all of these aspects in future work.

References

1. Emile Aarts and Reiner Wichert. Ambient intelligence. In *Technology Guide*, pages 244–249. Springer, 2009.
2. Sidney C Bailin and Walt Truszkowski. Cooperation between intelligent information agents. In *International Workshop on Cooperative Information Agents*, pages 223–228. Springer, 2001.
3. Matteo Baldoni, Jörg P. Müller, Ingrid Nunes, and Rym Zalila-Wenkstern, editors. *Engineering Multi-Agent Systems - 4th International Workshop, EMAS 2016, Singapore, Singapore, May 9-10, 2016, Revised, Selected, and Invited Papers*, volume 10093 of *Lecture Notes in Computer Science*. Springer, 2016.
4. Ahmed-Chawki Chaouche, Amal El Fallah Seghrouchni, Jean-Michel Ilié, and Djamel Eddine Saïdouni. A Higher-order Agent Model with Contextual Management for Ambient Systems. In *Transactions on Computational Collective Intelligence XVI*, volume 8780 of *LNCS*, pages 146–169. Springer Berlin Heidelberg, 2014.
5. Ahmed-Chawki Chaouche, Amal El Fallah-Seghrouchni, Jean-Michel Ilié, and Djamel Eddine Saidouni. From intentions to plans: A contextual planning guidance. In *Intelligent Distributed Computing VIII*, pages 403–413. Springer International Publishing, 2015.
6. Ahmed-Chawki Chaouche, Amal El Fallah-Seghrouchni, Jean-Michel Ilié, and Djamel-Eddine Saïdouni. Learning from situated experiences for a contextual planning guidance. *J. Ambient Intelligence and Humanized Computing*, 7(4):555–566, 2016.
7. Angela Di Febbraro, Nicola Sacco, and Mahnam Saeednia. An agent-based framework for cooperative planning of intermodal freight transport chains. *Transportation Research Part C: Emerging Technologies*, 64:72–85, 2016.
8. Amit Dua, Seema Guide Bawa, and Neeraj Guide Kumar. *Efficient Data Dissemination in Vehicular Ad Hoc Networks*. PhD thesis, 2016.
9. Edmund H Durfee. Scaling up agent coordination strategies. *Computer*, 34(7):39–46, 2001.
10. Thorsten Engesser, Thomas Bolander, Robert Mattmüller, and Bernhard Nebel. Cooperative epistemic multi-agent planning for implicit coordination. *arXiv preprint arXiv:1703.02196*, 2017.
11. Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.
12. David Jung and Alexander Zelinsky. An architecture for distributed cooperative planning in a behaviour-based multi-robot system. *Robotics and Autonomous Systems*, 26(2-3):149–174, 1999.
13. Julien Nigon, Marie-Pierre Gleizes, and Frédéric Migeon. Self-adaptive model generation for ambient systems. *Procedia Computer Science*, 83:675–679, 2016.
14. Eiichi Osawa. A metalevel coordination strategy for reactive cooperative planning. In *ICMAS*, volume 95, pages 297–303, 1995.
15. Ferdinand Piette, Costin Caval, Cédric Dinont, Amal El Fallah-Seghrouchni, and Patrick Taillibert. A multi-agent solution for the deployment of distributed applications in ambient systems. In *Engineering Multi-Agent Systems - 4th International Workshop, EMAS 2016, Singapore, Singapore, May 9-10, 2016, Revised, Selected, and Invited Papers*, pages 156–175, 2016.
16. Reid G Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, (12):1104–1113, 1980.
17. N Streitz. Citizen-centred design for humane and sociable hybrid cities. *Hybrid City*, pages 17–20, 2015.
18. Alejandro Torreno, Eva Onaindia, and Oscar Sapena. Fmap: Distributed cooperative multi-agent planning. *Applied Intelligence*, 41(2):606–626, 2014.
19. Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.