



Macrophage+: A game with a purpose for applying human intelligence in control mechanisms

A. Tarihi, H. Haghighi*, F. Shams Aliee, and A. Setarenejad

Department of Computer Science and Engineering, Shahid Beheshti University G. C., Tehran, Iran.

Received 16 April 2018; received in revised form 25 October 2019; accepted 11 January 2020

KEYWORDS

Control mechanisms;
 Observer/controller
 pattern;
 Human-based
 computation;
 Game with a purpose;
 Crowdsourcing;
 Interactive game.

Abstract. Originally, control mechanisms were proposed to replace the need for human intervention in operational environments and, thus, enhance the precision and reaction time. Nowadays, new requirements in computer systems such as adaptation have made the design of control mechanisms more challenging. The observer/controller pattern is one of the control mechanisms proposed to control many interacting independent elements with intelligent decisions. An important challenge of designing these mechanisms is that the knowledge needed for decision-making is provided by experts; therefore, the process becomes time consuming and costly, depending on the availability of experts and their costs. This study hypothesizes that employing a game With A Purpose can help improve the process of providing knowledge in such control mechanisms by using crowd-sourcing and involving non-expert humans in an enjoyable manner. This hypothesis was investigated by Macrophage+, a game with a purpose implemented for this goal. A number of experiments were conducted to evaluate Macrophage+, focusing on both its applicability and effectiveness in the context of the observer/controller pattern as well as its enjoyability for the players. The results show that Macrophage+ is a successful game with a purpose that involves non-expert humans in the application of the observer/controller pattern.

© 2020 Sharif University of Technology. All rights reserved.

1. Introduction

While control mechanisms enable adaptive behavior in modern systems [1], their design is not a straightforward task. This complexity comes from the need for constant adaptation in unforeseen situations [2]. Adaptation is one of the sources of complexity in modern systems, although it makes them autonomous, requiring less human intervention [3] and more capable to react to changes accordingly [4]. Control mecha-

nisms in adaptive systems are closed-loop based on feedback from the system and the environment [3]. The common adaptation mechanism in realizing such systems is based on the analysis of data gathered from monitoring them, planning the proper actions based on the analyzed data, and finally executing the decision. This mechanism is commonly called the MAPE cycle (Monitoring, Analysis, Plan, and Execution) [4–9]. One of the control mechanisms that simplifies the MAPE cycle is the observer/controller pattern [6] in which the control mechanism is divided into two components: the observer and the controller. The combination of observer and controller (usually denoted as observer/controller) can perform effectively as a MAPE-based control mechanism [6]. Due to the simplicity of the observer/controller pattern, the focus of this paper will be on this specific pattern so that

*. Corresponding author. Tel.: +98 21 29904190;
 Fax: +98 21 2243 1804
 E-mail addresses: a.tarihi@sbu.ac.ir (A. Tarihi);
 h.haghighi@sbu.ac.ir (H. Haghighi); f.shams@sbu.ac.ir (F.
 Shams Aliee); a.setarenejad@sbu.ac.ir (A. Setarenejad)

we can avoid the inherent complexity of other control mechanisms.

While there are many approaches to knowledge discovery [10–13] in the context of control mechanism, especially in the realization of a rather straightforward control mechanism like the observer/controller pattern, expert knowledge must be employed in order to specify which data the observer collects and realize the decisions the controller makes. This results in introducing additional costs associated with modeling the system by the experts. A desirable solution for reducing the costs is to involve non-experts. Generally speaking, a non-expert individual cannot help realizing the control mechanism directly; however, from a theoretical point of view, the complex mechanism of information processing alongside heuristic thinking is common among humans [14]. It means that if the information processing capability is tackled correctly, non-experts can help solve the problems, similar to experts [15]. Specifically, regarding the observer/controller pattern, a means of interaction with the domain must exist in order to simplify and present it in a manner that the required knowledge can be extracted from the problem-solving capability of non-experts.

An interesting question arises here: “how can non-experts work with a model that aims to provide the required knowledge effectively for the observer/controller pattern as a specific subdomain of the control mechanisms?” In this regard, human-based computation is a growing field concerned with the application of human intelligence collectively in the form of crowd-sourcing so as to solve various problems [14–17]. Two rather popular genres of human-based computation are game With A Purpose (GWAP) and mechanized labor [18]. Mechanized labor, like an actual employment, requires payment for the involvement regardless of the expertise [19]. On the other hand, GWAP uses game and game-related concepts in order to keep the player motivated during the game that is actually solving the problem [20]. Currently, GWAP has been applied to a variety of problems and has been established as a successful realization for human-based computation [14]. For our problem, employing a GWAP can reduce the cost for knowledge acquisition to nothing, but designing the game as an interactive model of the problem has its own cost.

Hypothesis/problem: Regarding the above discussion, this study hypothesizes that GWAP can help solve the problem of knowledge acquisition in control mechanisms by crowdsourcing. This approach results in more availability of individuals capable of providing the knowledge. In addition, non-experts can be employed in a free-of-charge manner.

In order to show that our hypothesis holds, a few questions need to be addressed correctly:

Q1: What form of knowledge acquisition is presented as a game so that non-expert individuals can solve it?

Q2: How can our GWAP be designed so that it is both playable and enjoyable for the players?

Q3: How much effective is the application of the knowledge gathered by the designed GWAP for a control mechanism based on the observer/controller pattern?

Solution: The answer to question **Q1** involves a well-known skill-rule-knowledge model [21,22]. This model categorizes human performance in carrying out tasks (or processing the information) in a general manner. Regarding this model, the GWAP must capture the required knowledge. A GWAP, called Macrophage+, specially designed for a rule-based adaptation logic for those control mechanisms, is proposed in which a number of agents with limited actions to perform on some resources must be controlled. In relation to the skill-rule-knowledge model, such a rule-based adaptation logic for the control mechanisms simplifies the analysis of the players’ actions, provided that the game is easy to learn and mastered so that the players can reach the rule-based and skill-based levels in a reasonably short time. This enables the player to make decisions almost directly applicable to the actual control mechanism with the details of the operational environment hidden by the game. This makes the game a tool that collects the knowledge in the form of rules suitable to solve the real problem.

In order to hide the difficulty of the control mechanisms from the players, a different domain is applied for the GWAP. The chosen domain is biology and, accordingly, the concept of the macrophages was used so as to create a game capable of collecting the required knowledge for the control mechanisms without exposing the real purpose of the GWAP. In Macrophage+, the game environment is the human body in which the player is responsible for protecting against any harms made by external entities.

It is organized as a collection of levels assigned to the players which can be categorized into two groups: solved scenarios and semi-random levels. The former set is solved beforehand and given to the players for the game introduction. It must be noted that no solution is given to the players; thus, they still face the challenge of completing the levels successfully. The scenarios ensure that the game is playable for the players (i.e., there is certainly a solution for them). These levels are given to the players sorted by their difficulty (based on the solution we devised beforehand) so that the players’ skills are perfected as they progress in the game. Solving each of these levels may lead to either the same solution we already found or a new solution. If a new solution is found, it means that the player has found a potentially, but not necessarily, a more efficient

way of winning a solved scenario. This new solution may be used in extracting new rules.

The second set of levels is semi-random levels that resemble the challenges that the actual control mechanism must overcome in the operational environment. They are based on the scenarios with some variations so as to create totally new levels, with no knowledge of their solution beforehand. A mechanism, which will be discussed later, has been devised for ensuring that the players will not face unsolvable levels consecutively. Together, the first and second sets of levels ensure that the game is both playable and enjoyable for the players (regarding question Q2).

Evaluation methodology: An experiment was carried out in order to evaluate the game: some scenarios were created for solving cases in which a set of tasks was to be performed by a number of agents with the possibility of malfunction and loss. It involves two groups of volunteers: the non-experts and experts in the control mechanisms domain. The former was given the GWAP to play, while the latter was given the corresponding real domain problem. Both groups participated in the experiment individually and in parallel. The purpose of having two groups was to compare the performance of both groups in solving the problems. Apart from this comparison, in order to investigate that Macrophage+ is a useful GWAP in the control mechanism domain, three metrics introduced in [14] were used:

1. Throughput = average number of problem instances solved per human hour;
2. Average Lifetime Play (ALP) = the overall amount of time that each player plays the game averaged across all people who have played it;
3. Expected contribution = throughput \times ALP.

In addition, questionnaires were given to the non-experts regarding the experiment. They were asked about their idea of the real purpose of the game and whether the game was enjoyable or not. The reason for the first question was to determine whether the game was successful in hiding the real problem or not, while the second emphasizes the game aspects of Macrophage+. To show the effectiveness of the approach (regarding question Q3), the extracted rules were analyzed and tested on two sets of scenarios: the first consisted of the game scenarios played by the players, while the second included new (unused) scenarios. These scenarios represented a collection of scenarios (from the solved scenarios and semi-random levels) that were not employed in the game and were solved manually for the evaluation purpose. The purpose of the first set was to ensure that the collective knowledge could solve the context problem it was

extracted from, while the second was intended to show that the rules were generic enough for the control mechanism domain.

Result summary: The results showed that the in-game levels were solvable by the nonexperts. This emphasizes that the designed GWAP is applicable in the context of the problem it was designed for. Interestingly, based on the filled questionnaires, the real purpose of the game was not guessed by the players, and that the game was fairly enjoyable for them. In addition, Macrophage+, as a GWAP, was able to reduce the time needed for providing the knowledge compared to that of using experts.

Contribution: The main contributions of this paper can be summarized as follows:

- Employing GWAP concepts in the control mechanism domain;
- Providing a new solution for knowledge acquisition in the control mechanism domain;
- Shorter time and lower cost than that of using experts in knowledge acquisition.

Paper organization: In the following section, the background of this work is presented to create a common understanding of concepts that are used throughout the paper. Section 3 is dedicated to the related work. The game design is introduced in Section 4, presenting the principles and main concepts of the game. The evaluation of the proposed game is conducted in Section 5. The last section is devoted to the conclusion and some directions for future work.

2. Background

This section includes a brief introduction to the various decision-making methods in the control mechanisms and the biological concepts related to the proposed game design.

2.1. Knowledge acquisition and decision making in control mechanisms

Control mechanisms can be categorized based on their decision-making methods, namely model-based, rule/policy-based, goal-based, and utility-based [6,9,23].

Model-based approaches rely on modeling various aspects of the system such as states and goals. They utilize these aspects in order to make proper decisions based on them [9]. The majority of the existing research pieces fall into the model-based category [2,24–30]. Architectural models, model checking, and constraint evaluation are among the existing model-based concepts and techniques used in this category [2,24,26,30].

The rule/policy-based approaches specify the system behavior in various situations using “if-then” like constructs [9], whereas goal-based approaches specify the desirable states of the system to guide the system to achieve them [23,31–36]. Tools and techniques such as rule engines and policy enforcement are used for the rule-based and goal-based approaches, respectively [32,37].

The utility-based approaches use utility functions and try to maximize them [9]. They rely on these functions so as to provide a direction for the desired state of the system [38–41].

2.2. Macrophage

As the basis for the proposed GWAP, macrophages are a type of white blood cells that defend the body by digesting any unhealthy entity including foreign substances, dead cells, etc. [42]. They can digest a large number of substances in a short time using their specialized enzymes [43]. The macrophages can distinguish the entities based on the chemical substances in the environment [42].

The mapping between the macrophage notion, as a biology phenomenon, and the control mechanism concept, as the problem tackled by our proposed GWAP, is as follows: the external entities or invaders represent the resources that the system must finish processing before their deadline, while the macrophages indicate the computations performed on the resources. The concept of eating the invaders is analogous to finishing a process or computation on a resource. Detailed information about this mapping is given in Section 4.

3. Related work

While there is a considerable body of works on control mechanisms and their applications, the proposed systematic search did not reveal any GWAP designed for acquiring knowledge for the control mechanisms in general and the observer/controller pattern specifically. Therefore, a broader search criterion was employed to review the body of work concerning the GWAP notion in a more general scope. In this manner, the first subsection is dedicated to a general survey on research pieces on control mechanisms with a focus on the observer/controller pattern. The second subsection is devoted to different cases of research on GWAP, gamification, and serious games that are not related to control mechanisms.

3.1. Control mechanisms and the observer/controller pattern

The traditional systems relied on human operators for their knowledge of control [44]. Improvements caused by emerging technology aim at replacing the need for human intervention with advanced control mechanisms, making systems more autonomous and

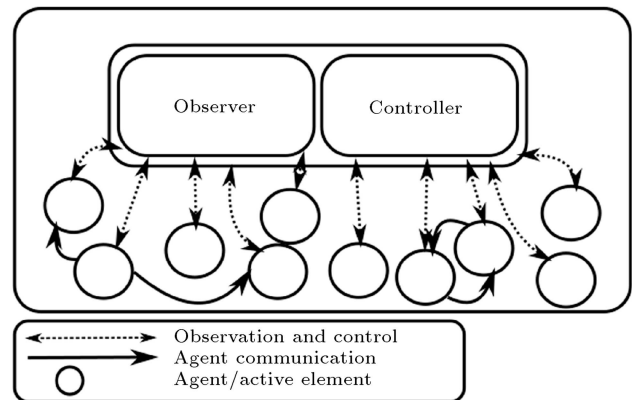


Figure 1. An overview of the observer/controller pattern.

less reliant on human operators [6,7]. In a control mechanism, the behavior of the system is used as a source of information that affects and/or corrects the future decision [45]. A control mechanism consists of two distinct layers. A lower (underlying) layer that provides the functionality of the system, while the second one guides the system [4].

The MAPE cycle [5] is a pattern that realizes such mechanisms. It consists of four stages, namely monitoring, analysis, planning, and execution. As mentioned earlier, the observer/controller pattern is a simple MAPE-based control mechanism that consists of two components [6,7,46]. Monitoring and analysis are carried out by the observer, while the controller performs the planning and execution [6,7]. Figure 1 depicts a system controlled by an observer/controller pattern. The system consists of communicating agent/active elements. The observer monitors the elements, while the controller commands the underlying layer (agents/active elements).

The original observer/controller pattern proposed in [46] employed a rule-based approach to decision-making, augmented by several computation models dedicated to modifying the rules. In addition, many other works [47–49] employed a similar rule-based approach.

Roth et al. [50] used the generic observer/controller pattern in a middleware with an architecture close to the MAPE cycle [51] that offers self-healing, self-protection, self-configuration, and self-optimization enablers to the ordinary services running on the middleware. Each ordinary service must implement a special interface to use the desired enabler. This work, like many others [45,52–54], employed a model-based approach to decision-making by involving the ordinary services in requesting the property they required.

In some other works, such as Nafz et al. [2] and Güdemann et al. [52], the generic observer/controller pattern was refined by using an approach called restore invariant approach, which tried to achieve system goals

by keeping its invariants and correcting the system behavior in case of any invariant violation. This approach can be categorized as a goal-based (keeping the invariants) approach to decision-making.

3.2. GWAP, gamification, and serious games

As stated in Subsection 3.1, no GWAP and no serious game designed for the control mechanism domain were found; therefore, the related works were chosen from different domains.

Some works did not propose a game; however, they contained ideas related to gamification [55,56]. In [55], a generic gamification enterprise platform was proposed that consisted of rules authored using a business rule management system. The rules were monitored via an analytic process. In this approach, the enterprise events were fed to the gamification platform. Gamification in [55] involved the people at an enterprise level as a part of the business process, which is not the case for the control mechanism. The demand for autonomous operation dictated minimized human intervention in the process.

Guy et al. [57] proposed a crowdsourcing platform for enterprise, in which individuals could create their own games using the collective intelligence of the enterprise. The games generated by this platform were used for data collection on various topics, helping the authors get ideas and enhancements from other colleagues. The proposed platform was competent for data gathering; however, in the case of knowledge acquisition for control mechanism, the scenarios are very specific to the domain and cannot fall under a generic platform.

Rani et. al. [58] proposed that the application of a feedback loop could help change the game difficulty regarding the player's skill. It helped maintain a high level of challenge for the player. They could keep the players engaged in the game with decreased boredom and anxiety compared to that of a normal challenge. The feedback was measured via biofeedback wearable sensors based on the physiological reactions of the players. In contrast, the idea of feedback at the selection level inside Macrophage+ was employed, but it is based only on the outcome of the levels in order to prevent successive losses caused by the semi-random levels.

Curator [59] was a GWAP devoted to collection recommendation. It involved the players participation in an online manner. The players were randomly paired and presented with collections of items to match. Scoring of the items was based on the preferences of both players, making each player consider the preferences of the other for a better score. The collections were analyzed for finding the related items for collection recommendation. This GWAP used the outcome of the players for the recommendation; however, because only the final result was recorded, it lacked analyzing

the way the player played the game. For extracting the rules for the control mechanism, recording players' actions are of vital importance, because they denote how the control mechanism must interact in new situations.

Chen et al. [60] proposed a GWAP for geotagging, in which the game server assigned tasks to the players involved in geotagging a location. There were two roles assigned to the players in the game: Requester and solver. The former asked geotagging questions that were replied by either the server or the other players. If the answer was not available, the latter group answered the question via an assigned task. There was no specific limitation on how the tasks were assigned to the solver group. Every player was free to choose the role based on his/her interest without restriction. The server employed a scheduling algorithm for task assignment in order to answer as many questions as possible. From the viewpoint of the knowledge acquisition problem, it was similar to the Curator GWAP.

Pheromander [61] is a serious game for improving human-computer interactions for a pheromone-based swarm. In this game, the swarm behavior is changed via pheromone placement by the player. The behavior is ranked with respect to the swarm goal and is displayed to the player in order to motivate him/her for better placement of pheromone and better working with the swarm from the user perspective. Pheromander simulates the swarm's behavior when pheromone is placed, engaging the player in solving the real problem. This approach cannot be used with the control mechanism, since it requires knowledge that the non-expert may not have.

KissKissBan [62] is a GWAP that uses human computation in both collaborative and competitive manners. This game involves identifying captions for images with three players in each round. A pair of players try to caption an image with the same words. The third player takes the role of blocker that tries to block some words to prevent other players from reaching a common strategy. The actual role of the blocker is to prevent cheating, thus increasing the accuracy of the labeled images. The blocker role changes between the players in 15 rounds of the game uniformly, resulting in each player taking the role of blocker for five rounds. In the knowledge acquisition problem, reaching a common strategy is desirable, because it denotes a common solution to the problem that needs to be extracted in the form of rules. Therefore, no blocker role is required for Macrophage+.

The ESP game [14] is a popular game for image tagging. Two players are randomly paired and given the same set of images to label. The goal of each player is to guess the label given by the other player, leading to a potentially agreed label for the image. There have been many improvements proposed for this popular

GWAP including new algorithms for puzzle selection [63]. Other games based on ESP have been proposed that use a similar approach to labeling [64,65]. The ESP game tries to reach a shared result between the players, while in the GWAP proposed in this paper, all players are seeking a common goal (i.e., solving the levels). The latter goal can be easier to communicate between the players, since the game can indicate how many levels are finished.

The Villain Villie game [66] is a GWAP specifically designed to mine human shape perception. Its main idea is to employ human computation to solve a computationally hard problem. The game places the player in the role of a detective that finds clues about criminals, ultimately leading them to characterize their shapes and emotional traits. The gathered data are analyzed in order to devise a human body-trait model. In this regard, Villain Villie has common elements with Macrophage+, but the focus of the former is to devise a model regarding a computationally hard problem, while the latter focuses on reducing costs using non-experts.

The Rings game [67] is a GWAP for generating test data required in software testing. The game converts the code under test into a pipe network, in which each decision statement in the corresponding code is converted into an obstacle with a specific shape representing the condition. Now, the player has to drop a ring with the proper size so that it can pass all the obstacles inside the pipe network (i.e., exits from the other side of the network). The size of the ring indicates the proper input data to test each path in the corresponding code. This game is very similar to Macrophage+, with the same goals, but unseen scenarios (represented as semi-random levels) are an important factor in Macrophage+.

Kondreddi et al. [68] combined the notion of human computation with information retrieval for better knowledge acquisition. They proposed a game that involves a sequence of interactions centered around knowledge represented in the form of triples or quads. The player completes the required knowledge, and answers are analyzed so as to create other meaningful questions in order to acquire more knowledge. In contrast, Macrophage+ is designed to extract knowledge from solutions provided by the players when playing the game. Therefore, it is not sensible to provide questions and answers to acquire knowledge from the players in this context.

Verbosity [69] is a GWAP in the form of quiz game that collects commonsense by the players. One player describes a word using a template, while the other tries to guess the word correctly. The answers are used for enriching the resulting commonsense facts.

In [12], employing the dynamic scripting technique was proposed. This work used automated

knowledge acquisition from available sources to play a real-time strategy game as an adaptive opponent. The sources used in this study include other artificial intelligence scripts employed in strategy games. The results showed that some general strategies could be generated using the proposed approach. Although the automation idea is very interesting, applying this approach in the control mechanism domain faces some difficulties. Most notably, acquiring external sources for control mechanisms is difficult due to the specific needs of each control mechanism and different approaches employed.

The focus of the study in [11] was on how the advances in communication technology allowed location-based knowledge acquisition using crowdsourcing. The proposed game was named Tell-us-where. It was in the form of a website that encouraged the players to describe their current location. Since the players were encouraged to participate in the game via a reward system (i.e., describing a place had a chance of receiving a gift), some mechanisms were implemented so as to prevent fake data from being collected. While Tell-us-where is effective in collecting data, it cannot capture the data in the control mechanism domain because the knowledge about them is not common and, therefore, it is difficult to collect effective scenarios using this approach.

TiE [70] is a virtual world for the promotion of artistic heritage. It engages people in micro games from which the player learns about the art. A typical micro game includes a puzzle of some piece of art. In this regard, the flow of information is opposite that of the knowledge acquisition problem, in which the knowledge is aggregated from the players.

In summary, it can be said that none of the related works can be applied readily to the problem of knowledge acquisition in the control mechanism domain. Some works such as [57,58,68–70] focused more on acquiring knowledge than applying it in a manner that can be applicable to another domain problem. Some other works [61,62,66,67] focused on finding a solution to their respective domain; however, these GWAPs are not specialized in the control mechanism domain. The aim of Macrophage+ is to provide an effective solution to the problem of knowledge acquisition in the control mechanism domain.

4. Game design

In this section, Macrophage+ will be presented from a practical point of view suitable to employ in an industrial environment. Hence, an illustrative example has been prepared in order to indicate the various aspects of a generic rule-based control mechanism capable of handling possible failure during the operations. This illustrative example has been applied to many

works related to the control mechanisms of such a category [2,6,71,72]. In this way, the applicability of Macrophage+ can become more tangible. The generic requirements devised from the illustrative example will be shown to be fulfilled by Macrophage+ by presenting various aspects of this GWAP. These aspects include the manner in which the levels are designed, how the levels are assigned to the players, and finally how the data are gathered to extract the related rules.

4.1. Illustrative example

The operational environment of the illustrative example consists of a collection of agents commanded by a control mechanism. Each agent has a few arms/tools capable of performing the following tasks: drilling a hole in the workpiece, inserting a screw in the hole, and tightening the screw (Figure 2(a)). The control mechanism must issue commands to the robots in a manner that when a failure occurs in a robot (e.g., a robotic arm is broken), the overall process can still continue. When such failure occurs, the control mechanism must change the assignment of the robots. An example of such change of assignment can be seen in Figure 2(b), in which a drilling tool of a robot has been broken and, therefore, the system has changed the assignment of the tools in order to sustain the processing of the workpieces. It should be noted that there are limits to the assignment of tasks, meaning that it is possible that no solution can be found at some point and the process may fail. Another complexity in this control mechanism is caused by lack of guarantee for the arrival time of the workpieces; thus, the system must process the incoming workpiece even when the load is more than what expected (one workpiece at a time). For instance, if two workpieces arrive instead of one workpiece, the system has to change the assignment of tools in a way that no delay happens in the process.

A desirable control mechanism for this environment must have the following major properties:

- P1:** Performing the required tasks on the workpieces using the agents;
- P2:** Recovering from failures of the agents;
- P3:** Handling of timing issues (e.g., workpieces not arriving orderly).

Realizing these properties can ensure a desirable control mechanism capable of operation in a real environment.

4.2. Macrophage+ overview

Macrophage+ is a mobile GWAP based on the macrophage phenomenon that tries to solve the problem of knowledge acquisition for control mechanisms. The game environment is designed to resemble inside the body invaded by external entities or simply invaders. The invaders cross the screen from the leftmost corner, giving the player an opportunity to see and interact with them before reaching the right side and doing “harm” in the form of game over (Figure 3).

The goal of the game is to “eat” all the invaders, while they are visible on the screen in order to protect the body from harm. The only means of interaction with the invaders is to touch a macrophage and drag it to one of the invaders; then, the macrophage “eats” the invader.

To do so, the player must interact by selecting

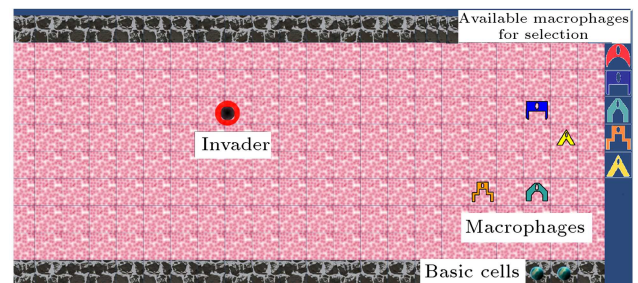


Figure 3. The annotated screenshot of the game indicating its various elements.

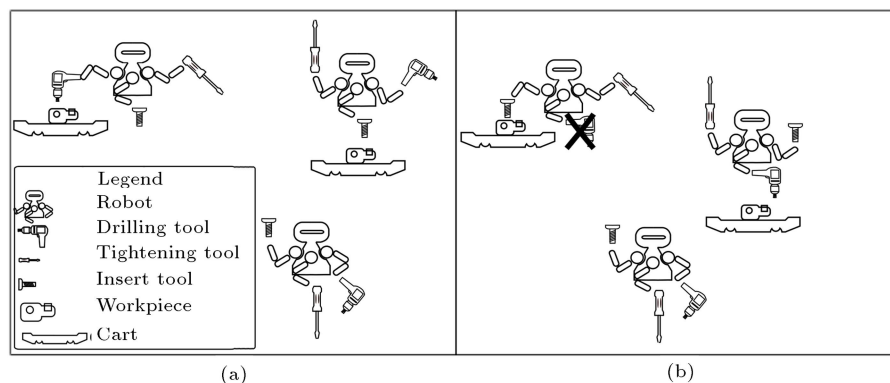


Figure 2. (a) The sample self-organizing resource-flow system in normal operation. (b) When one of the tools (i.e., the drilling tool in the leftmost robot) is broken, reconfiguration is performed and another robot begins to use its drill instead of the broken one.

and dragging macrophages to the invaders. These macrophages are created from the basic cells (indicated by small circles in the bottom right side of Figure 3). The number of the basic cells decreases by one when one of the macrophages on the right of the screen is dragged into the screen. This denotes that a macrophage is created from a basic cell. Limiting the interacting control to selecting and dragging macrophages simplifies the game and makes the game playable by a wider range of audience. Every level is finished when all the invaders are eaten by the macrophages.

In order to realize the major properties extracted from the illustrative example, the mapping between the game elements and the control mechanism domain must be explained. In doing so, the game design can be presented from the operational environment perspective.

The invaders represent the resources that the control mechanism has to control and manage so as to finish tasks on them before the corresponding deadline, while the macrophages denote the computation components that perform the task. Since each task may require different computations to be completed (e.g., read the file from the disk, perform computation, send the results), we represent each computation required to complete a task as a layer of protection around the invader. For instance, in the illustrative example, each of the invaders represents the resources that they process in which each may have up to three layers (drilling a hole, inserting the screw, and tightening it). Regarding this point of view, unless all the computations are finished (i.e., all the layers are removed), the resource cannot be processed (or in other words, the corresponding invader cannot be eaten). In this way, an invader may require more than one macrophage.

For indicating different computations, the layers are given distinct shapes and colors, which can be removed by the matching macrophage (which has the right color and shape). The shapes of the macrophages were designed in a way that the corresponding macrophages could be attached to the invaders, simulating the actual binding between a macrophage and its target invader (Figure 4(a) and (b)).

Figure 4(c) presents invaders with multiple layers and co-centered shapes. For example, the left invader has square and circular layers. The precedence of layers of a resource indicates the order in which the computations must be performed (e.g., drilling is carried out before placing the screw in the illustrative example). The player can choose from a limited set of small circular cells and convert them to the desired type of macrophage. This set of cells represents the basic cells that evolve to the actual macrophages. The player can convert each basic cell into a different type of macrophages for faster operation or deal with

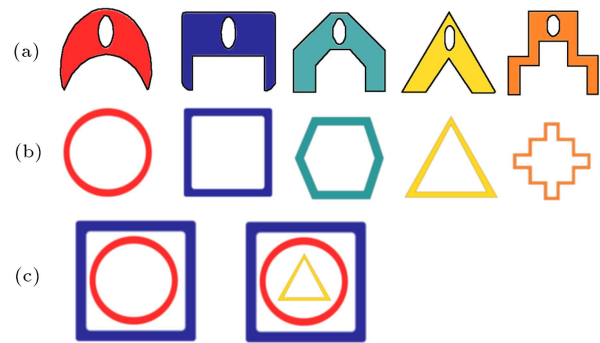


Figure 4. (a) Implemented types of macrophage. (b) The corresponding invader shapes. (c) Some sample invaders using the layers.

multiple invaders simultaneously (it is analogous with processing multiple resources at once in the illustrative example). Like real control mechanisms, the number of available computation components is limited; hence, the set of basic cells and the macrophages is limited. This design was chosen to make the player's decision as close as possible to real conditions. The combination of player actions regarding the invaders represents the commands that the control mechanism gives to the agents in order to perform all required tasks on the workpieces, fulfilling property P1 from the illustrative example.

The last feature of the game is the possibility of a macrophage death (i.e., disappearing from the screen). This feature has been added since, in a real control mechanism scenario, the computation components (either software or hardware) are prone to failures. For example, the robotic arms may be broken in the illustrative example. When a macrophage dies, it means that the player is unable to use it and has to replace it by another macrophage created from the basic cells. The actions performed by the player in response to such events are analogous to the required actions that the control mechanism should perform in the illustrative example when an agent malfunctions. This helps fulfill property P2 in the illustrative example.

Lastly, it is possible that more than one invader enters the screen which indicates a timing issue (i.e., some workpieces arrive concurrently, not sequentially). In such situations, the control mechanism is expected to perform some additional actions in order to complete all the required tasks on the workpieces. The knowledge that the player uses in these cases can help the actual control mechanism handle the timing issues encountered in the operational environment (property P3 in the illustrative example).

4.3. Game play and level design

The player begins each level with a predetermined set of macrophages and basic cells. In this way, the game

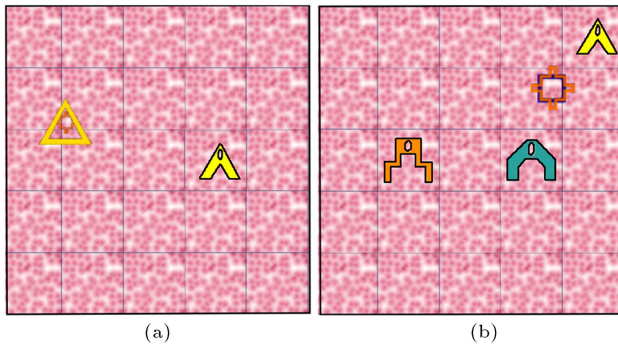


Figure 5. A sample attack on an invader by a macrophage in two screenshots: (a) Before attack and (b) after attack.

represents the initial state of a system having a fixed amount of operational components. When the game starts, the invaders begin to appear from the leftmost corner of the screen, each with visible layers surrounding them, indicating the required macrophages (computation tasks). Each correct macrophage attack causes a layer to be taken from the invader which means one of the tasks required to perform the computation is carried out. Figure 5 depicts a scenario in which a macrophage attacks an invader in two snapshots (before and after). In Figure 5(a), the invader has two layers (triangular and sawtooth). Afterward, it is attacked by the triangular macrophage, and only the sawtooth layer remains (Figure 5(b)).

The level has two possible outcomes: 1) The player wins when all the invaders are eliminated, and 2) The player loses when an invader has crossed the screen with at least one layer.

From the player's viewpoint, the levels are randomly generated with an increase in the difficulty as the game progresses. As mentioned before, there are two types of levels: solved scenarios and semi-random levels. The solved scenarios are the scenarios solved manually by experts beforehand and, then, converted into playable levels, while the semi-random levels are random levels created from the solved scenarios by changing their parameters randomly.

Each of these two types of levels can be categorized into three sets of levels, namely normal, challenging, and adaptive context levels. The “normal” scenarios are the usual levels at which no macrophage dies. These levels are mostly easy and used as introduction levels for the game.

The “challenging” scenarios are designed to match near-real-world problems faced by the control mechanisms in actual systems. The “adaptive context” scenarios focus on replacing the damaged components, resulting in a dynamic solution. Complicated scenarios are achieved by assigning a high death rate for the macrophages (around 50%). The last two types of levels may have no solution because they employ random

changes to the normal levels. Thus, their assignment to the players has to be made with special care.

The structure of scenarios is given to the Macrophage+ game via a simple human-readable file format that specifies the following major elements:

- The maximum number of available basic cells (from which the macrophages can be created);
- For each invader, the following items;
- Entrance time in seconds;
- Deadline, i.e., the time (in seconds) the invader should exit the screen;
- Layers by specifying their type and precedence.

The semi-random levels are created on the fly on the server side once and given to the players. Because of their random nature, the set of levels created from the same solved scenarios can vary in each initialization of the game.

Regardless of the level type, there are the following rules that govern both types of levels:

Rule 1: Each level must have at least one invader and one basic cell;

Rule 2: Each invader must have at least one layer;

Rule 3: Every invader should have an entrance time and a deadline time with $time < deadline$ holding;

Rule 4: The number of basic cells must be positive.

4.4. Data gathering

When a level is played by the player, every action of the player is recorded anonymously regardless of the outcome. This leads to recording the player failures which are also useful for deducing wrong actions. Each level is recorded separately that contains only the game play and no personal information in order to preserve the anonymity of the players. The major events recorded are given as follows:

- The result of the level;
- Creation of macrophages from basic cells;
- Consumption of layers by macrophages;
- Death of macrophages;
- Elimination of invaders.

4.5. Rule extraction

Rule extraction is currently the only method used for game play analysis in this work. The general form of rules used by control mechanisms is $R_i : C_i \rightarrow A_i$ in which R_i represents the rule, C_i the set of conditions that must be valid in order to apply the rule, and A_i the set of actions. The control mechanism uses these rules as the knowledge needed so as to make decisions. In order to simplify the rules presented in this paper,

a pseudo-code like syntax is used for the extracted rules. Such a representation can communicate the relationship between the game and the target domain. Therefore, there are verbs such as *Prepare* or *Use Tool* in the extracted rules:

$$R_1 : Task_1 \rightarrow Prepare Tool_1;$$

$$R_2 : Task_2 \rightarrow Prepare Tool_2;$$

$$R_3 : Task_3 \rightarrow Prepare Tool_3;$$

$$R_4 : Task_1 \rightarrow Use Tool_1;$$

$$R_5 : Task_2 \rightarrow Use Tool_2;$$

$$R_6 : Task_3 \rightarrow Use Tool_3.$$

Since the scenario is very basic, the extracted rules are trivial. They indicate that for each remaining *Task*, the *Tool* that can carry out the process must be prepared (either it exists like a macrophage that has been already created, or it must be created) and, then, the required tool must be used.

The recorded game plays are analyzed automatically based on the outcome of the game, trying to extract the right decisions in the form of the above rules (In extracting the rules, the notion of Decision Tree [73,74] is used which is a known machine learning technique capable of extracting the required rules). For instance, the actions in a game won by a player can be used as a basis for similar scenarios. This means that when the control mechanism has all or a subset of the tasks played by the player, the same or similar actions can be safely used.

Another application of the recorded actions is to extract patterns used for replacing the damaged components. Such actions can make the control mechanism exhibit adaptive behavior in unexpected situations.

A final step is considered for the unification of the rules. The necessity of unification is due to the fact that the game is played simultaneously by many players and, also, the levels have many similarities; this may lead to repeated rules. This unification consists of three steps:

1. Eliminating duplicated rules: Since the levels are shared among the players, it is natural to have similar or exact game plays, leading to the very same set of rules;
2. Normalizing the rules: Some rules may have the same semantics without having any similar condition or action (i.e., the same form of rule, yet with different symbols). Such rules are recognized and normalized manually. For instance, if there are two distinct levels at which there is only one invader with one layer, the solution is to use one macrophage with the corresponding color/shape (e.g., the above-mentioned rules R_1 , R_2 , and R_3 are unified);

3. Removing possible loops: It is possible to have rules like R_i , R_j , and R_k in which the actions of R_i lead to the triggering of R_j and, in turn, R_k is triggered, resulting in the triggering of R_i again. This loop will prevent the control mechanism from reaching a decision at all. The loops can be eliminated by further analysis such as decomposition of the rules and eliminating their cause.

Using the unification leads to the final set of rules applicable in the control mechanism.

5. Evaluation

The evaluation of the Macrophage+ game is performed from two major viewpoints:

1. The applicability of the rules extracted from the game for control mechanisms;
2. The evaluation of the game as a GWAP.

The second viewpoint is important for two reasons:

1. If the game is not motivating and interesting to the players, it will not be played by the players regardless of its applicability in the desired domain;
2. Another important issue is that the player must not be concerned with the complexity rooted in the actual domain. This goal is achieved by hiding the technical details from the game, making it appear not to be related to the real domain.

If the game cannot address the above issues, then it is not possible to solve the problem of knowledge acquisition in control mechanisms collectively in the form of crowdsourcing. Therefore, it is important to evaluate the game as a GWAP and study the related metrics.

5.1. Evaluation of the applicability for control mechanisms

Method: The evaluation of the applicability of the proposed GWAP in the control mechanism context has been performed through an experiment conducted by the cooperation of two groups of participants: experts and non-experts. The non-expert group played the game, while the experts were given the actual problem. Both groups performed their tasks individually and simultaneously. The game data were gathered using the game server and, then, analyzed in order to extract rules. The extracted rules were applied manually at two levels (solved scenarios and semi-random levels) given to the players. The results of both groups were used separately so as to solve new levels and ultimately two case studies. The extracted rules were applied to new scenarios that were not employed in the game to conduct the final evaluation of the applicability in actual control mechanisms.

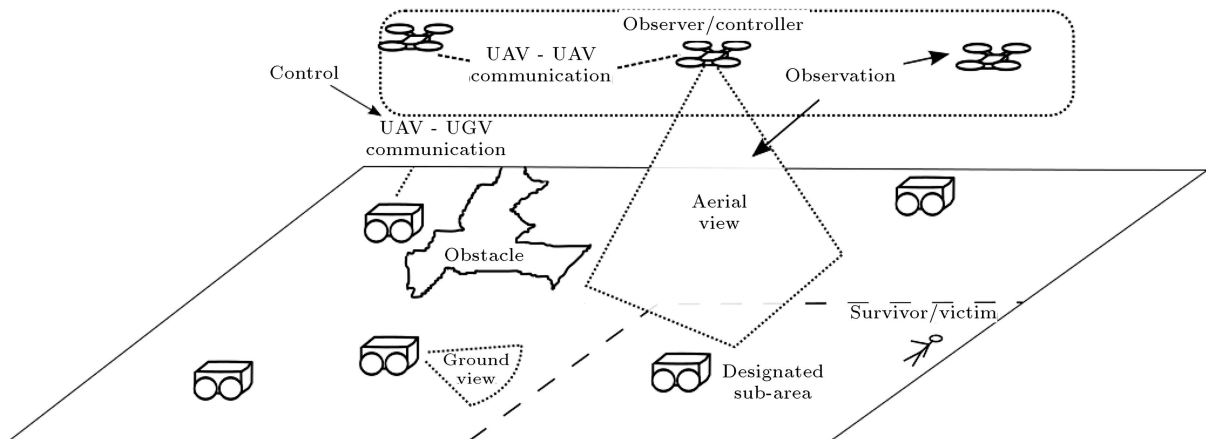


Figure 6. A schematic overview of the Search And Rescue (SAR) scenario, indicating various key elements.

Case studies: This study used the illustrative example as a case study. However, since this example is rather simple, another rather complicated scenario was applied.

Our chosen case study is a search and rescue (SAR) scenario in which a number of robots search for possible survivors and potential threats (e.g., wildfire) and, then, report them to the rescue base. Using robots in SAR has been an interesting problem studied in many research pieces [75–78]. In these missions, the goal is to find victims and survivors in an area, aid them or call for help as soon as possible with minimal exposure of the rescue team to the hazards in the environments. The nature of the hazards varies from environment to environment, generally resulting from a disaster happened in the area (e.g., wildfire, ruined building) with individuals incapacitated or trapped.

Many types of robots are used in these missions such as Unmanned Ground Vehicle (UGV) and Unmanned Aerial Vehicle (UAV), which are usually instrumented with the necessary equipment like sensors, cameras, and the Global Positioning System (GPS) [79]. They can detect victims using their cameras or other sensors [78,79]. In some of these missions, only one type of the mentioned robots is employed [80], while others have a hybrid approach to robots [75–78].

The scenario at focus here begins with the assignment of the target area. When the UGVs are deployed in the vicinity of the designated area, the mission starts. Due to the obstructed view of the UGVs and the size of the area, a UGV may achieve limited knowledge of the environment, potentially omitting some sub-areas and immediate threats. To compensate for this weakness, the use of UAV has been proposed in many works [75–78] so as to provide aerial support for the mission by performing the following tasks: 1) finding a path for UGVs in case of obstacles, 2) assigning higher priority sub-areas to UGVs (detecting these sub-areas can be done by using sensors such

as heat sensors employed for detecting fire), and 3) participating in searching inaccessible sub-areas such as ruined buildings, roofs, etc.

To perform the mentioned tasks, UAVs need to observe UGVs and cover all or a sufficient portion of the area. Since no UAV has a global view of the area, the required information is obtained using UAV to UAV communications. Figure 6 shows a schematic overview of the SAR scenario. The UAVs have an aerial view of the area including UGVs, obstacles, and survivors/victims. The UGVs employ their sensors to search the subarea to gain a more detailed view of their vicinity in comparison to the UAVs.

The UAVs communicate important data about the environment which facilitate realizing the mission goal, i.e., completely search the area in order to find the survivors/victims. They also communicate with the UGVs and give them orders to be followed. This leads to a feedback control loop similar to the observer/controller pattern in which the UAVs perform control tasks alongside normal SAR tasks (e.g., search a building). If we want to map this scenario to the Macrophage+, we have to identify the resources and the tasks. The major tasks in the control mechanism include aggregating data from the area using UAV to UAV communication and planning for a better area coverage based on the threat or rescue priority. The tasks that must be performed by the UGVs in the simplest scenario include: 1) getting path from UAVs, 2) searching the area and reporting the status of the area, 3) getting the next area assignment, and 4) moving to the next area. The tasks required from the UAVs in the same scenarios are as follow:

1. Calculating the path for UGVs;
2. Searching the area and reporting the status of the area;
3. Assigning the next area to search for the UGVs.

The resources in this scenario correspond to the sub-areas, which require different combinations of tasks to be processed. Using these tasks, different invaders representing the tasks required to navigate and search different areas were created in our experiment.

Ten scenarios were selected from both case studies, solved, and then converted as levels for the evaluation. These scenarios were arranged from single task scenarios in which no macrophage died to complicated scenarios with a high death rate for the macrophages. Moreover, each player played five unsolved levels in the form of semi-random levels based on the above ten scenarios. Finally, seven new scenarios were selected and solved, but not given to the players. The purpose of using these new scenarios was to evaluate the resulting rules in facing new situations (represented in the form of these seven scenarios).

Table 1 lists some of the extracted scenarios. For example, the sixth line denotes a scenario in which three invaders with the indicated layers must be eliminated. The complexity of this scenario is caused by the concurrent entrance of Macrophages #2 and #3. Since both macrophages have the same layers (layers 3, 4, and 5), the elimination of the layers must be carried out at the same time so that the invaders can be eliminated before their deadline. Therefore, an additional macrophage must be created by a basic cell for eliminating the first layer of the third invader, another two for eating the second layer, and finally two others for eating the last layers of the remaining invaders. Hence, a total of seven basic cells are at least required. It is interesting to note that if the invaders were slow enough, a better solution could be devised that involves four macrophages instead of seven. In such a case, there would be enough time to eliminate the first layer of invader #2; then, the same macrophage is used for eliminating the first layer of invader #3 while using a new macrophage for the second layer of invader #2. In this manner, the level

can be solved by using a total of four macrophages.

Evaluation criteria and metrics: The percentage of levels solvable by the extracted rules (for both scenarios and solvable semi-random levels) is the main metric. Another metric is the percentage of the new scenarios (not played by the players) solvable by the set of extracted rules. Finally, the total time for acquiring the knowledge by playing the game compared to that of the experts was considered as the last metric.

Participants: The participants were divided into two groups based on their experience with the control mechanisms and algorithms:

Players: A sample of sixteen people (six females and ten males) were recruited from outside of the university campus by public call. The only requirement was being familiar with playing simple games. This group was chosen in a manner that they had no or little knowledge of control mechanisms, such that they generally could not devise any effective solution to the problem systematically. This was ensured by selecting people with unrelated backgrounds or working experiences. The main reason for selecting the first group was to have non-expert people in control mechanisms, and to show that the game can have wide audience. The mean age of the participants was 30 years (ranging between 24–42, standard deviation of 4.3), and their average year of education was 5.7 (range 4–12).

Experts: The second group includes three experts with a good control mechanism and algorithm background who could devise a solution for the actual problem. They were asked to solve the actual control problem and extract the rules required. Only a detailed description of the operational environment was given to this group; therefore, no scenarios were needed. Their

Table 1. Some sample extracted scenarios.

Scenario no.	Summary	Invaders	Number of	Chance of
			basic cells	macrophage death
1	Show the game play	Single invader, one layer	1	0
2	Simple solved level	Two invaders, one layer each	1	0
3	Show how to replace the dead macrophage	Two invaders, one layer each	2	100
4	Simple solved level	Three invaders, two layers each	2	0
5	Semi-random level from row #4	Three invaders, three layers each	4	20
6	Complex solved level	Three invaders: -#1 with layers 1,2 -#2 with layers 2, 3, 4 -#3 with layers 2, 3, 4 after #1 enters, #2 and #3 enter at the same time	7	0

average age was 32.3 (range 32–34; standard deviation 1.14), and they had an average of 7 (ranging between 6–8) years of experience.

Materials: The game was run on a computer with an ordinary specification of common work station. Macrophage+ was developed using the PyGame cross-platform Python library. Images and graphics were created using Inkscape and Gimp software.

Tasks: The following two tasks have been performed:

Playing session: All players attended a single voluntary playing session in which they were free to quit the game when they wanted to. The players were asked to solve as many levels as they could in the session time, which could help with having game plays that result from motivated players for better results;

Control mechanism knowledge provision session: All experts attended a single session and were asked to manually provide the required knowledge for the control mechanism.

Procedures: The detailed procedures for the evaluation tasks are described here:

Playing session: Before starting the playing session, the participants were informed about the study, without mentioning the real purpose of the game. Then, the players were asked about agreeing to play the game. Afterward, the game mechanics were explained to the players orally, and all the questions were answered. When there was no question and all the players were ready, the game was installed on the smartphones and the game play began in an empty room.

Whenever any player finished playing Macrophage+, he/she was asked whether he/she agreed to contribute to the questionnaire regarding the game play.

Control mechanism knowledge provision session: After agreeing to perform the experiment, the three experts were asked to provide the required knowledge for the given specification so that the control mechanism could make its decision using this knowledge. There was no time limit for this session. Like the players, they were asked whether they agreed to participate in the evaluation process related to them.

Data collection: The game data were collected via the game server-side facilities. The output produced

by the experts was collected on the paper.

Results and discussion: The players solved 97% of the scenarios (the selected 10 levels) of the game. Due to the short time of the experience, no player left the session. The result of the semi-random levels was 80%.

Around 80% of the unused scenarios (i.e., scenarios prepared for evaluation purpose) were solved, which regarding the total time required for the experiment (around 10 minutes) was acceptable compared to the time the experts spent (around 2 hours).

The short levels and their successive assignment kept most of the players immersed in the game; however, a number of the participants stated that the game was much easy. It was expected that, due to the easiness of the game, most of the levels would be solved by the players. Interestingly enough, the semi-random levels that were designated to be unsolvable by the player were verified via manual analysis after the experiment. However, the players were not blocked by such levels due to the level assignment. The observation and questionnaires indicate that the players considered the game to have a random nature like many existing games. The details of the questionnaire will be studied in the next subsection.

Solving the puzzles by the players was faster than devising actual rules by the expert group. Table 2 shows two of the these extracted rules in pseudo-language in which *UseOr CreateTool* represents the action of creating or using existing tools, and *Process* denotes the action of processing the resource (using *UseOr CreateTool* or other means). On the other hand, in comparison with the players' output, the experts' output was capable of solving more complex scenarios and, generally, could be used to address a wider range of scenarios and operational environments.

The comparison of the results of the two groups (the output of the experts and the rules extracted from the game play of the non-experts) indicates that the GWAP is able to reach faster results with no cost. While the expert team reached a final specification for the control mechanism in a longer duration of time, the time required in an actual environment would be much longer. This increase can be attributed to the fact that the work of the experts was voluntary and might be carried out in a more casual manner than that in an actual environment. In addition, since the experiment was conducted on a volunteering basis, the experts agreed not to be paid; however, in an actual environment, we should accept considerable cost.

It appears that although the semi-random levels

Table 2. Some sample extracted rules.

$Resource(r) = \{Task(t_1)\} \rightarrow UseOrCreateTool(T_1)$
$Resource(r_1) < Resource(r_2) < Deadline(r_1) > Deadline(r_2) \rightarrow Process(r_2)$

were less than the scenarios, these levels played an important role in reaching the extracted rules, because the randomness led to the levels and difficulties similar to those of the experts' specifications.

Although this experiment was limited due to the participant number and time restrictions, it shows that Macrophage+ is applicable in the domain of the control mechanism and can be applied to this problem using crowd-sourcing and opening new opportunities for GWAPs in such a context.

Regarding the related works, geotagging proposed by Chen et al. [60], cannot be used for the control mechanism domain because of the question-and-answer theme of the geotagging game, which makes the application of the game impossible in the control domain. Pheromander [61] used the user interactions for improving the pheromone-based swarm. The player had to tackle the pheromone placing for optimum results. In the problem of knowledge acquisition, this requires domain knowledge and a simulation environment in which the player adds the rules. Such an approach limits the participants to the experts. Moreover, providing a simulation environment would add much higher costs than Macrophage+. The KissKissBan [62] game faced difficulties for proper application in the control domain. To do so, the collaborating players had to have detailed knowledge of the domain, which limited the participants to domain experts only. This applies to the ESP game [14]. The Curator GWAP [59], which was based on grouping of similar collections that could be used for grouping similar scenarios, included solving these scenarios, which limited the players to domain experts.

In summary, it can be said that Macrophage+ has been a better solution for knowledge acquisition so far than the existing GWAP. The incurred development cost of Macrophage+ is still lower than that of employing experts in the control mechanism domain.

5.2. Evaluation of the game as GWAP

The evaluation of Macrophage+ as a GWAP was carried out using some metrics calculated by us from the playing session and the questionnaires filled by the participants playing the game.

Evaluation method and metrics: The evaluation of game engagement in general has been the focus of many studies [81–83]. The studies show that the questionnaire is an applicable method for verifying this topic. Therefore, in the context of GWAP, the metrics were calculated as in [84] via questionnaire. Von Ahn and Dabbish [84] proposed three metrics for evaluating a GWAP:

1. Throughput as the average number of problems solved by the players in a period of time;

2. ALP as the average of total time the game played by the players;
3. Expected contribution as the product of the two previous items that denotes the average number of problems solved by the players in a period of time (e.g., an hour).

Through the gathered data, the three metrics could be calculated; however, these metrics can be used for determining which of the two GWAPs solving the same problem is better [84]. Because the contribution of using GWAP in providing the required knowledge for the control mechanism is new, calculating these metrics will not bring insight for evaluating the game. However, we will present the calculated metrics for comparison in the future works with the same problem. Further, due to some unique properties of the game, calculating the throughput metric needs careful consideration. The number of levels cleared is not a good way of calculating the throughput because the levels are shared across many players; therefore, they are repeatedly solved without any guarantee about the uniqueness of the solution.

Because of the above-mentioned restrictions, a questionnaire was used so as to see how much the players enjoyed the game. It is also important to know how much Macrophage+ has been successful in hiding the control problem from the players.

Participants: The same groups of the players in Subsection 5.1 cooperated in the second evaluation.

Material: To collect data, a questionnaire was designed (Figure 7). The required data for this questionnaire are given below:

1. Personal information of players (age, number of years spent in education, gender, previous experience in gaming, and smartphone model);
2. The degree which indicates how much the players enjoyed the game: (0 = not at all, 2 = a little, 3 = somewhat, 4 = good, 5 = very);
3. The real purpose of the Macrophage+ game from the player's point of view (the given choices include biology, control mechanisms, artificial intelligence, scheduling, reflex measurement, or to write other purposes).

The third part explores the ability of the GWAP to hide its real purpose from the players. The answer to this question is important because it can reveal whether or not the game is successful in hiding the technical details related to the target domain. In this manner, we can be sure that the game is usable by non-experts; therefore, the problem behind the game can be tackled via crowd-sourcing.

Questionnaire for Macrophage+ Game

A. Personal Information

Age	
Gender	
Years of Education	
Years of Experience in Gaming	

B. Game Experience

The game was engaging	1- yes	2- no
The game was boring	1- yes	2- no
I would have liked to continue playing the game	1- yes	2- no
While I play the game, I forgot about where I was	1- yes	2- no
The game was fun	1- yes	2- no
Playing the game was easy	1- yes	2- no

C. Game Developer's Intentions

Please choose the option showing the intention(s) of the game developers for creating such a game. Note that you can choose more than one choice.

<input type="checkbox"/>	Biology
<input type="checkbox"/>	Control Mechanism
<input type="checkbox"/>	Algorithm
<input type="checkbox"/>	Entertainment (No other purpose)
<input type="checkbox"/>	Improving reflexes
<input type="checkbox"/>	Other:

Figure 7. The questionnaire players filled.

Tasks and procedures: Having finished playing the game, each player was asked to participate in filling the mentioned questionnaire.

Results and discussion: All the problems were solved (12 levels: 9 scenarios, and 3 semi-random levels); therefore, the average number of problems solved was 12, which led to the extraction of 16 rules in 20 minutes or 0.8 of a problem per minute. The ALP metric was calculated as 17 minutes. The expected contribution was “ $0.8rule/min \times 17min = 13.6rule$ ”. It is worth mentioning that the problem was solved in parallel, which is the reason for the $0.8rule/min$ rate. Hence, it is possible to increase this number by adding more participants and levels without the need for a longer time.

As mentioned before, since Macrophage+ is the first attempt at utilizing the GWAP technique in the control mechanism domain, the calculated metrics are not applicable for making a sensible comparison. However, these measures were calculated for comparison with future works.

The questionnaire used in evaluating the game revealed that most of the players enjoyed the game. The game scored 84 out of 100 (standard deviation

of 12.96) by the players, which is an acceptable score regarding the early stage of the game. Among the 16 players, 12 people (75%) expressed their interest in playing Macrophage+ again. An important finding is that no player could guess the real intention of Macrophage+ (the answer to the third question in the questionnaire), which indicates that the game has been successful in hiding its real purpose. Reflex measurement, algorithm (optimization), and biology were respectively the highest selected areas for the game by the participants.

6. Conclusion and future works

Game With A Purpose (GWAP) is a technique used in the human-based computation domain in which humans help solve problems that are costly or hard to solve. This technique has been applied to many problems. In the control mechanism, the knowledge required for decision-making is usually provided by experts, a lengthy and time-consuming approach. This study proposed a GWAP, called Macrophage+, that solved this problem by disguising it as a simple game playable by non-expert people.

Macrophage+ is loosely based on real

macrophages that exist in the body eating the external entities. Each task to be performed by the control mechanism is simulated by an invader with many layers. The player uses the macrophages for eating the invaders layer by layer. Component failure in the system is simulated by dying macrophages and the recovery mechanism is simulated by the player's ability to replace them with new components. The levels are complete when all the invaders are "eaten" by the macrophages.

The results show that the game is successful in achieving its goals. However, there are still shortcomings to the proposed method. One of these shortcomings is that Macrophage+ only provides knowledge for the rule-based control mechanisms and there is no straightforward way to extend it to the other types of control mechanisms. The addition of metrics can adapt the game for goal-based adaptation logic for control mechanisms. In order to achieve this, the condition of winning or losing the levels can be extended by adding some other conditions describing the desired metrics (e.g., the number of failed components lower than 25%). In this regard, such a type of the game may introduce a particular type of challenge besides the normal levels to the players. This can increase the motivation of players to try them.

Devising a feedback mechanism for gathering scenarios from the operational environment, especially operational failures in the agents, may provide scenarios unforeseen by the game. This can be realized by enriching and structuring the logging mechanisms of the agents and sending the logs to the system, thus enabling the structured logs to be parsed so as to extract the required scenarios for later use in the GWAP. Hence, providing the GWAP with new challenges can enrich the game and provide possible solutions to the domain by analyzing the game plays. This may be achieved by gathering mechanism and monitoring logs generated by the involved systems.

Another shortcoming is that the tasks (i.e., layers of invaders) are only performed sequentially; hence, no useful rule can be gathered for performing concurrent tasks. The idea of hybrid layers can address this problem. By hybrid layers, we mean that some layers of the invaders consist of two or more different shapes (e.g., half of a layer is shaped like a square, while the other half is circular) and, therefore, require more than one macrophage in order to destroy them (i.e., one for the circular half and one for the square part).

References

- Shaw, M., Andersson, J., Litoiu, M., Schmerl, B., Tamura, G., Villegas, N.M., Vogel, T., Weyns, D., Baresi, L., Becker, B., Bencomo, N., Brun, Y., Cukic, B., Desmarais, R., Dustdar, S., Engels, G., Geijs, K., Goschka, K.M., Gorla, A., Grassi, V., Inverardi, P., Karsai, G., Kramer, J., Lopes, A., Magee, J., Malek, S., Mankovskii, S., Mirandola, R., Mylopoulos, J., Nierstrasz, O., Pezzé, M., Prehofer, C., Schäfer, W., Schlichting, R., Smith, D.B., Sousa, J.P., Tahvildari, L., Wong, K., and Wuttke, J. "Software engineering for selfadaptive systems: A second research Roadmap", De Lemos, R., Giese, H., Müller, H.A., Shaw, M., Editors, In *Software Engineering for Self-Adaptive Systems II. Lecture Notes in Computer Science*, **7475**, pp. 1–32, Springer, Berlin, Germany (2013).
- Nafz, F., Steghöfer, J.P., Seebach, H., et al. "Formal modeling and verification of self-* systems based on observer/controller architectures", In Camara J., de Lemos R., Ghezzi, C., and Lopes A., Editors, *Assurances for Self-Adaptive Systems, Lecture Notes in Computer Science*, **7740**, pp. 80–111, Springer (2013).
- Salehie, M. and Tahvildari, L. "Self-adaptive software: Landscape and research challenges", *Trans. Auton. Adapt. Syst.*, **4**(2), pp. 1–42 (2009).
- Macías-Escrivá, F.D., Haber, R., del Toro, R., et al. "Self-adaptive systems: A survey of current approaches, research challenges and applications", *Expert Systems with Applications*, **40**(18), pp. 7267–7279 (2013).
- Kephart, J.O. and Chess, D.M. "The vision of autonomic computing", *Computer*, **36**(1), pp. 41–50 (2003).
- Müller-Schloer, C. and Tomforde, S., *Organic Computing - Technical Systems for Survival in the Real World*, Springer, Berlin, Germany (2017).
- Müller-Schloer, C., Schmeck, H., and Ungerer, T., Editors, *Organic Computing - A paradigm Shift for Complex Systems*, 1st Edn., Springer, Berlin, Germany (2011).
- Seebach, H., Ortmeier, F., and Reif, W. "Design and construction of organic computing systems", In *Evolutionary Computation*, CEC 2007, pp. 4215–4221, Singapore (2007).
- Krupitzer, C., Roth, F.M., VanSyckel, S., et al. "A survey on engineering approaches for self-adaptive systems", *Pervasive and Mobile Computing*, **17**, pp. 184–206 (2015).
- Momtazi, S. and Moradiannasab, O. "A statistical approach to knowledge discovery: Bootstrap analysis of language models for knowledge base population from unstructured text", *Scientia Iranica*, **26**, pp. 26–39 (2019).
- Winter, S., Richter, K.F., Baldwin, T., et al. "Location-based mobile games for spatial knowledge acquisition", In *Proc. of the Workshop on Cognitive Engineering for Mobile GIS*, Belfast, Ireland (2011).
- Ponsen, M., Spronck, P., Munoz-Avila, H., et al. "Knowledge acquisition for adaptive game AI", *Science of Computer Programming*, **67**(1), pp. 59–75 (2007).

13. Bimba, A.T., Idris, N., Al-Hunaiyyan, A., et al. "Towards knowledge modeling and manipulation technologies: A survey", *Intl. Journal of Information Management*, **36**(6), pp. 857–871 (2016).
14. Von Ahn, L. "Games with a purpose", *Computer*, **39**(6), pp. 92–94 (2006).
15. Cooper, S., Khatib, F., Treuille, A., et al. "Predicting protein structures with a multiplayer online game", *Nature*, **466**(7307), pp. 756–760 (2010).
16. Law, E.L.M., Von Ahn, L., Dannenberg, R.B., et al. "Tagatune: A game for music and sound annotation", In *Proc. 8th Intl. Conf. on Music Information Retrieval (ISMIR)*, **3**, pp. 361–364, Vienna, Austria (2007).
17. Quinn, A.J. and Bederson, B.B. "Human computation: A survey and taxonomy of a growing field", In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI'11)*, pp. 1403–1412, New York, USA (2011).
18. Sabou, M., Bontcheva, K., Scharl, A., et al. "Games with a purpose or mechanized labor?: A comparative study", In *Proc. of the 13th Intl. Conf. on Knowledge Management and Knowledge Technologies*, p. 19, Graz, Austria (2013).
19. Wang, A., Hoang, C.D.V., and Kan, M.K. "Perspectives on crowdsourcing annotations for natural language processing", *Language Resources and Evaluation*, **47**(1), pp. 9–31 (2013).
20. Weng, L., Schifanella, R., and Menczer, F. "Design of social games for collecting reliable semantic annotations", In *16th Intl. Conf. on Computer Games (CGAMES)*, pp. 185–192, Louisville, KY, USA (2011).
21. Reason, J., *Human Error*, Cambridge University Press (1990).
22. Rasmussen, J. "Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models", *Systems, Man and Cybernetics, IEEE, Transactions on*, **3**, pp. 257–266 (1987).
23. Lalanda, P., McCann, J.A., and Diaconescu, A. "Autonomic computing", In *Principles, Design and Implementation*, 1st Edn., Springer, Germany (2013).
24. Garlan, D., Schmerl, B., and Cheng, S.W. "Software architecture-based self-adaptation", In *Autonomic Computing and Networking*, pp. 31–55, Springer, Germany (2009).
25. Garlan, D., Cheng, S.W., Huang, A.C., et al. "Rainbow: Architecture-based self-adaptation with reusable infrastructure", *Computer*, **37**(10), pp. 46–54 (2004).
26. Oreizy, P., Medvidovic, N., and Taylor, R.N. "Architecture-based runtime software evolution", In *Proc. of the 20th Intl. Conf. on Software Engineering*, pp. 177–186, Kyoto, Japan (1998).
27. Canal, C., Poizat, P., and Salaün, G. "Model-based adaptation of behavioral mismatching components", *IEEE Trans. on Software Engineering*, **34**(4), pp. 546–563 (2008).
28. Filieri, A., Hoffmann, H., and Maggio, M. "Automated design of self-adaptive software with control theoretical formal guarantees", In *Proc. of the 36th Intl. Conf. on Software Engineering*, pp. 299–310, Hyderabad, India (2014).
29. Nafz, F., Ortmeier, F., Seebach, H., et al. "A generic software framework for role-based organic computing systems", In *Software Engineering for Adaptive and Self-Managing Systems, SEAMS'09. ICSE Workshop on*, pp. 96–105, Cape Town, South Africa (2009).
30. Fischer, P., Nafz, F., Seebach, H., et al. "Ensuring correct self-reconfiguration in safety-critical applications by verified result checking", In *Proc. of the 2011 Workshop on Organic Computing, OC'11*, pp. 3–12, New York, USA (2011).
31. Kaiser, G., Parekh, J., Gross, P., et al. "Kinesthetics extreme: an external infrastructure for monitoring distributed legacy systems", In *Autonomic Computing Workshop*, pp. 22–30, Seattle, Washington, USA (2003).
32. Georgas, J.C. and Taylor, R.N. "Towards a knowledge-based approach to architectural adaptation management", In *Proc. of the 1st ACM SIGSOFT Workshop on Self-managed Systems*, pp. 59–63, Newport Beach, California, USA (2004).
33. Fox, K.L., Henning, R.R., Farrell, J.T., et al. "System and method for assessing the security posture of a network using goal oriented fuzzy logic decision rules", April 19 2005. US Patent 6,883,101 (2005).
34. Berry, D.M., Cheng, B.H.C., and Zhang, J. "The four levels of requirements engineering for and in dynamic adaptive systems", In *11th Intl. Workshop on Requirements Engineering Foundation for Software Quality (REFSQ'05)*, p. 5, Porto, Portugal (2005).
35. Cheng, B.H.C., Sawyer, P., Bencomo, N., et al. "A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty", In *Intl. Conf. on Model Driven Engineering Languages and Systems*, pp. 468–483, Denver, Colorado, USA (2009).
36. Van Lamsweerde, A. and Letier, E. "Handling obstacles in goal-oriented requirements engineering", *IEEE Transactions on Software Engineering*, **26**(10), pp. 978–1005 (2000).
37. Sykes, D., Heaven, W., Magee, J., et al. "From goals to components: a combined approach to selfmanagement", In *Proc. of the 2008 Intl. Workshop on Software Engineering for Adaptive and Selfmanaging Systems*, pp. 1–8, Leipzig, Germany (2008).
38. Menascé, D.A., Ewing, J.M., Gomaa, H., et al. "A framework for utility-based service oriented design in sassy", In *Proc. of the First Joint WOSP/SIPEW Intl. Conf. on Performance Engineering*, pp. 27–36, San Jose, California, USA (2010).

39. Shen, C.C., Srisathapornphat, C., and Jaikaeo, C. "An adaptive management architecture for ad hoc networks", *IEEE Communications Magazine*, **41**(2), pp. 108–115 (2003).
40. Esfahani, N., Elkhodary, A., and Malek, S. "A learning-based framework for engineering feature-oriented self-adaptive software systems", *IEEE Transaction on Software Engineering*, **39**(11), pp. 1467–1493 (2013).
41. Brinkschulte, U., Pacher, M., and Von Renteln, A. "An artificial hormone system for self-organizing realtime task allocation in organic middleware", In *Organic Computing*, pp. 261–283, Springer, Berlin, Germany (2009).
42. Lodish, H., Berk, A., Kaiser, C.A., Krieger, M., Bretscher, A., Ploegh, H., Amon, A., Martin, K.C., *Molecular Cell Biology*, 8th Edn, W.H. Freeman and Company, New York, USA (2008).
43. Alberts, B., Bray, D., Hopkin, K., Johnson, A.D., Lewis, J., Raff, M., Roberts, K., Walter, P., *Essential Cell Biology*, 4th Edn, Garland Science, Taylor & Francis Group, New York, USA (2013).
44. Sugeno, M. and Nishida, M. "Fuzzy control of model car", *Fuzzy Sets Systems*, **16**(2), pp. 103–113 (1985).
45. Schmeck, H., Müller-Schloer, C., Cakar, E., et al. "Adaptivity and self-organization in organic computing systems", *ACM Trans. Auton. Adapt. Syst.*, **5**(3), pp. 1–10 (2010).
46. Richter, U., Mnif, M., Branke, J., et al. "Towards a generic observer/controller architecture for organic computing", *GI Jahrestagung*, **93**, pp. 112–119 (2006).
47. Komann, M. and Fey, D. "Marching pixels-using organic computing principles in embedded parallel hardware", In *Intl. Symposium on Parallel Computing in Electrical Engineering (PARELEC'06)*, pp. 369–373, Bialystok, Poland (2006).
48. Hartmann, J., Stechele, W., and Maehle, E. "Self-adaptation for mobile robot algorithms using organic computing principles", In Kubátová, H., Hochberger, C., Daněš M., Sick B., Editors, In *Intl. Conf. on Architecture of Computing Systems (ARCS'13)*, pp. 232–243, *Lecture Notes in Computer Science*, **7767**, Springer, Berlin, Germany (2013).
49. Branke, J., Mnif, M., Müller-Schloer, C., et al. "Organic computing-addressing complexity by controlled self-organization", In *Second Intl. Symposium on Leveraging Applications of Formal Methods, Verification and Validation (Isola 2006)*, pp. 185–191, Paphos, Cyprus (2006).
50. Roth, M., Schmitt, J., Kiefhaber, R., et al. "Organic computing middleware for ubiquitous environments", In *Organic Computing A Paradigm Shift for Complex Systems*, pp. 339–351, Springer, Berlin, Germany (2011).
51. Murch, R., *Autonomic Computing. Information Management*, IBM Press, California, USA (2004).
52. Gudemann, M., Nafz, F., Ortmeier, F., et al. "A specification and construction paradigm for organic computing systems", In *Second IEEE Intl. Conf. on Self-Adaptive and Self Organizing Systems*, pp. 233–242, Venezia, Italy (2008).
53. Kasinger, H. and Bauer, B. "Combining multi-agent-system methodologies for organic computing systems", In *16th Intl. Workshop on Database and Expert Systems Applications (DEXA'05)*, pp. 160–164, Copenhagen, Denmark (2005).
54. Tomforde, S., Rudolph, S., Bellman, K., et al. "An organic computing perspective on self-improving system interweaving at runtime", In *IEEE Intl. Conf. on Autonomic Computing (ICAC)*, pp. 276–284, Wurzburg, Germany (2016).
55. Herzig, P., Ameling, M., and Schill, A. "A generic platform for enterprise gamification", In *Joint Working IEEE/IFIP Conf. on Software Architecture and European Conf. on Software Architecture*, pp. 219–223, Helsinki, Finland (2012).
56. Khanzadia, M., Shahbazia, M.M., Arashpourb, M., et al. "Lean design management using a gamified system", *Scientia Iranica*, **26**(1), pp. 15–25 (2019).
57. Ido, G., Hashavit, A. and Corem, Y. "Games for crowds: A crowd sourcing game platform for the enterprise", In *Proc. of the 18th ACM Conf. on Computer Supported Cooperative Work and Social Computing, CSCW'15*, pp. 1860–1871, Vancouver, British Columbia, Canada (2015).
58. Rani, P., Sarkar, N., and Liu, C. "Maintaining optimal challenge in computer games through real-time physiological feedback", In *Proc. of the 11th Intl. Conf. on Human Computer Interaction*, **58**, pp. 22–27, Bonn, Germany (2005).
59. Walsh, G. and Golbeck, J. "Curator: A game with a purpose for collection recommendation", In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, pp. 2079–2082, Atlanta, Gorgia, USA (2010).
60. Chen, L.J., Wang, B.C., and Zhu, W.Y. "The design of puzzle selection strategies for esp-like gwap systems", *IEEE Trans. on Computational Intelligence and AI in Games*, **2**(2), pp. 120–130 (2010).
61. Kerler, S., Vilsmeier, J., Edenhofer, S., et al. "Pheromander: Real-time strategy with digital pheromones", In *8th Intl. Conf. on Games and Virtual Worlds for Serious Applications (VS-GAMES)*, pp. 1–4, Wurzburg, Germany (2016).
62. Ho, C.J., Chang, T.H., Lee, J.C., et al. "Kisskissban: A competitive human computation game for image annotation", *ACM SIGKDD Explorations Newsletter*, **12**(1), pp. 21–24 (2010).
63. Syu, Y.S., Yu, H.H., and Chen, L.J. "Exploiting puzzle diversity in puzzle selection for esp-like gwap systems", In *IEEE/WIC/ACM Intl. Conf. on Web Intelligence and Intelligent Agent Technology*, **1**, pp. 468–475, Washington, District of Columbia, USA (2010).

64. Otterbacher, J. "Crowdsourcing stereotypes: Linguistic bias in metadata generated via gwap", In *Proc. of the 33rd Annual ACM Conf. on Human Factors in Computing Systems*, pp. 1955–1964, Seoul, Republic of Korea (2015).
65. Harris, C. "Cluemein: Enhancing the esp game to obtain more specific image labels", In *Proc. of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts, CHI PLAY '18 Extended Abstracts*, pp. 447–452, Melbourne, Victoria, Australia (2018).
66. Ashraf, G., Why, Y.P., and Islam, T. "Mining human shape perception with role playing games", *GSTF Journal on Computing (JoC)*, **1**(1), pp. 58–64 (2018).
67. Amiri-Chimeh, S., Haghighi, H., Vahidi-Asl, M., et al. "Rings: A game with a purpose for test data generation", *Interacting with Computers*, **30**(1), pp. 1–30 (2017).
68. Kondreddi, S.K., Triantafillou, P., and Weikum, G. "Combining information extraction and human computing for crowdsourced knowledge acquisition", In *IEEE 30th Intl. Conf. on Data Engineering*, pp. 988–999, Chicago, Illinois, USA (2014).
69. Von Ahn, L., Kedia, M., and Blum, M. "Verbosity: A game for collecting common-sense facts", In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems (CHI '06)*, pp. 75–78, Montreal, Quebec, Canada (2006).
70. Bellotti, F., Berta, R., De Gloria, A., et al. "Exploring gaming mechanisms to enhance knowledge acquisition in virtual worlds", In *Proc. of the 3rd Intl. Conf. on Digital Interactive Media in Entertainment and Arts*, pp. 77–84, Athens, Greece (2008).
71. Fisch, D., Jänicke, M., Müller-Schloer, C., et al. "Divergence measures as a generalized approach to quantitative emergence", In *Organic Computing-A Paradigm Shift for Complex Systems*, pp. 53–66, Springer Berlin, Germany (2011).
72. Seebach H., Nafz F., Steghofer J.P., et al. "How to design and implement selforganising resource-flow systems", In *Organic Computing-A Paradigm Shift for Complex Systems*, pp. 145–161, Springer Berlin, Germany (2011).
73. Quinlan, J.R. "Induction of decision trees", *Machine Learning*, **1**(1), pp. 81–106 (1986).
74. Kohavi, R. and Quinlan, J.R. "Data mining tasks and methods: Classification: decision-tree discovery", In *Handbook of Data Mining and Knowledge Discovery*, pp. 267–276. Oxford University Press, Inc. Oxford, United Kingdom (2002).
75. Nakamura, S., Nakagawa, H., Tahara, Y., et al. "Towards solving an obstacle problem by the cooperation of uavs and ugvs", In *Proc. of the 28th Annual ACM Symposium on Applied Computing*, pp. 77–82, Coimbra, Portugal (2013).
76. Venkatesh, S., Lobo, R., and Sugunan, N. "Controlling unmanned ground vehicle using stationary airborne system", *Intl. Journal of Innovation and Applied Studies*, **8**(3), p. 958 (2014).
77. Grocholsky, B., Keller, J., Kumar, V., et al. "Cooperative air and ground surveillance", *IEEE Robotics & Automation Magazine*, **13**(3), pp. 16–25 (2006).
78. MacArthur, E.Z., MacArthur, D., et al. "Use of cooperative unmanned air and ground vehicles for detection and disposal of mines", In *Proc. of Intl. Society for Optics and Photonics*, **5999**, pp. 94–101 (2005).
79. Liu, Y. and Nejat, G. "Robotic urban search and rescue: A survey from the control perspective", *Journal of Intelligent & Robotic Systems*, **72**(2), pp. 147–165 (2013).
80. Anderson, B. and Crowell, J. "Workhorse auv-a cost-sensible new autonomous underwater vehicle for surveys/soundings, search & rescue and research", In *Proc. of OCEANS 2005 MTS/IEEE*, pp. 1–6, Washington, District of Columbia, USA, IEEE (2005).
81. Mayer, I., Bekebrede, G., Harteveld, C., et al. "The research and evaluation of serious games: Toward a comprehensive methodology", *British Journal of Educational Technology*, **45**(3), pp. 502–527 (2014).
82. Brockmyer, J.H., Fox, C.M., Curtiss, K.A., et al. "The development of the game engagement questionnaire: A measure of engagement in video game-playing", *Journal of Experimental Social Psychology*, **45**(4), pp. 624–634 (2009).
83. Norman, K.L. "Geq (game engagement/experience questionnaire): a review of two papers", *Interacting with Computers*, **25**(4), pp. 278–283 (2013).
84. Von Ahn, L. and Dabbish, L. "Designing games with a purpose", *Commun. ACM*, **51**(8), pp. 58–67 (2008).

Biographies

Ali Tarihi received his BS and MS degrees in Software Engineering from Shahid Beheshti University, Tehran, Iran. His research interests include bio-inspired computation, software architecture, and GWAP. He is currently a PhD student at the Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran.

Hassan Haghighi received his BS, MS, and PhD degrees in Software Engineering from Sharif University of Technology, Tehran, Iran. Testing, Gamification, GWAP, and formal methods are his main academic interests. He is currently an Associate Professor of the Faculty of Computer Science and Engineering at Shahid Beheshti University.

Fereidoon Shams Aliee received his BS and MS degrees from Shahid Beheshti University and Sharif University of Technology, respectively. He received his PhD in Software Engineering from the Department of

Computer Science, Manchester University, UK. Software architecture, SOA, and enterprise architecture are the main focus of his research. He is currently an Associate Professor of the Faculty of Computer Science and Engineering at Shahid Beheshti University.

Amirmehdi Setarenejad received his BS degree in Software Engineering from Shahid Beheshti University, Tehran, Iran. He is a graduate MS student from the Faculty of Computer Science and Engineering, Shahid Beheshti University.