A methodology to develop awareness in computer supported collaborative work using policies

Amir Talaei-Khoei\textsuperscript{a,∗}, Suchada Vichitvanichphong\textsuperscript{b}, Terje Solvoll\textsuperscript{c}, Pradeep Ray\textsuperscript{d}, Amir Hossein Ghapanchi\textsuperscript{e}

\textsuperscript{a} School of Systems, Management and Leadership, University of Technology, Sydney, Australia
\textsuperscript{b} Discipline of Informatics, Faculty of Arts and Business, University of the Sunshine Coast, Australia
\textsuperscript{c} Norwegian Centre for Integrated Care and Telemedicine, University Hospital of North-Norway, Norway
\textsuperscript{d} Asia-Pacific ubiquitous Healthcare research Centre (APuHC), The University of New South Wales, Sydney, Australia
\textsuperscript{e} School of Information and Communication Technology, Griffith University, Australia

A R T I C L E   I N F O

Article history:
Received 23 May 2013
Received in revised form 16 January 2014
Accepted 4 March 2014
Available online 28 April 2014

Keywords:
Computer supported cooperative work
Awareness
Intelligent agents
Policy

A B S T R A C T

Cooperation is significantly influenced by participants’ awareness of relevant information. The objective of this paper is to propose a methodology to design and develop applications that assist individuals to identify their awareness. Through extending Cooperative management Methodology for Enterprise Networks (CoMEN), this article introduces Policy-based Awareness Management (PAM) – a software engineering methodology that proposes the use of existing policy rules as a source to identify awareness. The methodology has been built on the logic of general awareness, and implements Directory Enabled Networks (DEN) policy structure. The contribution of the paper is illustrated through the wireless communication system at a hospital in Norway. We conclude that theory of general awareness and in particular PAM as an extension of CoMEN, is effective to identify relevant information for agents.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

People increasingly work and live in cooperative environments, where by growing communication technologies they are overloaded with irrelevant or loosely relevant information [1]. In addition, cooperative environments such as social networks, B2B e-commerce, healthcare and disaster management teams are emerged in line with the concept of information uncertainty [2], which requires adapting new approaches in cooperation. The evolution of cooperative environments has been marked by the methods that utilize intelligent IT tools to enhance cooperation among participants [3]. As such, cooperative environments have to deal with information uncertainty [4,5], and one issue that has arisen from the use of IT is that individuals are often overloaded with irrelevant or loosely relevant information [1]. This requires methods to identify the relevance of information as new, certain information comes to the fore. Research and design practices in Computer Supported Cooperative Work (CSCW) emphasize the role of awareness in understanding the relevance of information [3,6–8]. Daneshgar and Wang [9] encourage researchers to work on definitive methods to identify such awareness.

∗ Corresponding author.
E-mail addresses: amir.talaei@uts.edu.au (A. Talaei-Khoei), Suchada.Vichitvanichphong@research.usc.edu.au (S. Vichitvanichphong), Terje.Solvoll@telemed.no (T. Solvoll), p.ray@unsw.edu.au (P. Ray), a.ghapanchi@griffith.edu.au (A.H. Ghapanchi).

http://dx.doi.org/10.1016/j.jcss.2014.03.003
0022-0000/© 2014 Elsevier Inc. All rights reserved.
The initial ideas of this article have been borrowed from Cooperative management Methodology for Enterprise Networks (CoMEN) [3]. The contribution is motivated by the following drawbacks of CoMEN. It is regarded as uncontroversial that the dynamic and uncertain nature of everyday life overloads individuals with irrelevant or loosely relevant information [3,4]. Once new, certain information comes to the fore, a method to identify awareness is essential. This is missing in CoMEN. In addition to that, Computer Supported Cooperative Work (CSCW) has recently evolved to embrace the complexity-based paradigm [5]. This paradigm replaces deterministic perspectives of the internal and external views of systems by agency principles [6]. The agency principles emphasize the role of individuals in a system. Zacarias et al. [5] define the agency relationship as interactions between individuals and software agents to perform tasks on individuals’ behalf. Much research proposes the use of agents to aid individuals maintaining their awareness [7–9]. Although CoMEN proposes the use of software agents to implement CSCW-based applications, it lacks authoritative stages to design and implement such applications to identify awareness in run-time. By extending the initial ideas from CoMEN, Policy-based Awareness Management (PAM), the proposal of this paper, employs software agents to assist individuals identifying their awareness. The main product of this work is a software engineering methodology for developing such applications that are able to assist individuals to become aware of relevant information in cooperation.

To this end, we extend Cooperative management Methodology for Enterprise Networks (CoMEN) [3] to enable cooperative roles to identify the relevant information. We employ intelligent agents to assist individuals obtaining awareness. We describe a framework – the Policy-based Awareness Management (PAM) – to model agents’ awareness based on existing policies. This framework provides a foundation borrowed from literature in which the process towards policy-based awareness is grounded. We also propose a step-wise process to identify awareness from given sets of policies.

The paper is organized in the following way: Section 2 motivates the work by comparing different cooperative management methodologies. This section also provides the background knowledge of the work. Section 3 presents methodological guidelines of PAM to extend CoMEN. The section also presents the background knowledge of the work. Section 4 presents an interpretive case study at St. Olavs Hospital. This section applies PAM and the methodological guidelines to the wireless communication system at this hospital. The section also discusses some experiments on how the wireless communication system in the hospital can benefit from PAM. Section 5 discusses the implications and limitations of PAM as well as our conclusions and directions for future research.

2. Related work and background

The need for management of cooperative systems has been an objective of research in recent years [10]. Cooperative management involves formalizing the cooperation among number of individuals and organizational entities [3]. One of the major questions facing the management is how to utilize the emerging Information Technologies (IT) to facilitate cooperation [9]. There have been several methodologies proposed for the use of IT in cooperative management. We evaluate the different approaches in the literature based on the following criteria. These criteria are based on the work presented in [11]:

- **Concept management** that involves (1) understanding cooperative roles’ definitions for the same or similar concepts and (2) understanding the relationship of concepts defined by different cooperative roles, (3) ability to define local concepts as well as global concepts.
- **Application management** that involves the ability of one software system run by a cooperative role to interact to the software run by another role.
- **Scenario management** that involves recognizing the sequences of interactions between cooperative roles.
- **Event management** that involves recognizing events that initiate the scenarios.
- **Awareness management** that involves realizing the relevance of information.

A summary of our analysis is shown in Table 1.

2.1. Cooperative management Methodology for Enterprise Networks (CoMEN)

CoMEN uses Computer Supported Cooperative Work (CSCW) techniques to analyze scenarios. This involves a top–down analysis of the system in the following steps:

1. **Overall system study** that involves study the overall “big picture” of the system including the activities and environment.
2. **Logical component identification** that involves recognizing roles, possible actions and information structure in the system.
3. **Process study** that consists of study possible scenarios
4. **Cooperation enhancement** that involves identifying the relevant information, called awareness in each scenario for the roles. The basic idea is that a cooperative role becomes aware of information, if she/he can identify the relevance of that. While the information is not available to the cooperative role, he/she interacts with others who already know the required information. Therefore the role will know the information. This process is called cooperation enhancement.
Table 1
Cooperative management methodologies.

<table>
<thead>
<tr>
<th>Concept management</th>
<th>Application management</th>
<th>Scenario management</th>
<th>Event management</th>
<th>Awareness management</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internet Simple Network Management Protocol (SNMP) [13]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Open System Interconnection (OSI) [14]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Reference Model of Open Distributed Processing (RM-ODP) [15]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Telecommunications Management Network (TMN) [16]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Internet Engineering Task Force (IETF) [17]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Cooperative management Methodology for Enterprise Networks [9]</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

Table 2
Definitions for different types of awareness.

<table>
<thead>
<tr>
<th>Awareness type</th>
<th>Definition</th>
<th>About artifact</th>
<th>About people</th>
<th>About activity</th>
<th>About context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workspace Awareness</td>
<td>Up-to-the-minute information about the existence of entities in a shared workspace.</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Common-sense Awareness</td>
<td>General sense of who is around and what belong to them.</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Group Awareness</td>
<td>Understanding of people in the group, their responsibilities and their status.</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Social Awareness</td>
<td>Information about presence of people and their activities.</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Context Awareness</td>
<td>Cognizance of an internal or external entity that causes change in a situation.</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
</tbody>
</table>

CoMEN leaves us the following two questions:

1. How to identify awareness? Daneshgar and Wang [9] state that currently there is no definitive method to determine required awareness and they encourage research on methods for identifying awareness. Talaei-Khoei et al. [12], in a systematic review of literature, have come up with a process on Awareness Maintenance. They believe awareness identification is the missing step in the literature.

2. How to develop a system that enables roles to identify awareness? The system should assist individuals to identify awareness to relevant information and help them perform actions that reveals truth or falsity of the relevant information. Ray et al. [9] state that use of intelligent agents can be useful to assist cooperative roles recognizing awareness. However the details of the development have been ignored.

2.2. What is awareness?

There exist a number of reviews for awareness in the literature of CSCW [13–15]. They define five types of awareness: (I) Workspace Awareness [16], (II) Common-sense Awareness [17], (III) Group Awareness [17], (IV) Social Awareness [18] and (V) Context Awareness [19], also known as Situation Awareness [20,21]. Table 2 provides the proposed definitions for the different types of awareness and compares them in terms of types of information they cover.

Omoronyia et al. [15] argue that context awareness cuts across the other types of awareness. This awareness considers the changes in states of the other awareness types. People work on different artifacts and activities, in different situations and with different people. Such awareness is highly contextual and cannot be addressed by the other types of awareness. [3,23] define context awareness as an understanding of relevant information that is required for an individual. Omoronyia et al. [15] define the relevant information to an individual as information that fully characterizes the desired situations of that individual. The desired situation is the intended situation in which the individual’s behavior will result. Therefore, awareness is relevance of information that an individual is required to know. By accepting this definition for awareness and considering the above mentioned understanding of relevance, we take the both accounts of awareness introduced by Ray et al. [3] – emphasizing on individuals, and introduced by [19] – emphasizing on situation. This relates the two views for context awareness in the literature. Dey [6] considers context as information about the entity of which the individual is

\[1\] A mediator for any sort of interaction among participants. An artifact can be physical, such as a report, or can be cognitive, such as skills and experiences. For more details about the concept of artifacts, readers must refer to [22].
aware. Although the notation of context awareness shows the relevance of the context, it does not refer to the knowledge about validity of the context [15,24,25]. Our concept of awareness is most closely related to context awareness [39].

2.3. Awareness in agents

Research in intelligent agents has been interested in the natural semantics for awareness as a mental attitude. The classical approach is the possible-worlds model in which a state can be considered possible in addition to true or false [26].

The literature in possible-worlds model of agents presents logic of general awareness [24,25]. The essential idea is relevance of knowledge. Under the possible-worlds interpretation, a valid sentence and its consequences are true at every world that the agent considers possible. However, a known sentence and its known consequences may or may not be relevant. Therefore, in the logic of general awareness, an agent implicitly knows all the valid sentences, but it changes its implicit knowledge to the explicit knowledge if and only if, the agent is aware of the sentence. Sillari [25] defines awareness of a propositional sentence as the relevance of that sentence to a situation. Therefore, the notation of awareness does not refer to validity of a sentence. Regardless that a sentence is valid or not, an agent becomes aware of a sentence if and only if it identifies the relevance of the sentence to the situation. Halpern and Pucella [27] argue that with the logic of general awareness we must explain how awareness can be obtained. Policy-based Awareness Management (PAM) uses policies as a source for identifying awareness.

2.4. Policy structures

There are three main policy structures widely used in distributed cooperative systems: (1) Directory Enabled Networks – next generation (DEN-ng) [28], (2) TeleManagement Forum (TMF) [29], and (3) Distributed Management Task Force (DMTF) [30]. TMF and DMTF lack an awareness view, while DEN-ng presents its policy model to support awareness in distributed systems. Therefore, we borrow our policy structure from DEN-ng. It provides a meta-model to implement awareness with policy rules; however, it does not address the use of policies to identify awareness.

DEN-ng advises that a policy is a composition of different policy rules, where each rule defines “event-condition-action” semantics. The semantics is such that the rule is evaluated when an event occurs. Once the condition clause is satisfied, the modality of action will be applied which may or may not result in executing the action. Sloman [31] defines modalities of actions as permitting, forbidding, requiring and deterring. Following the DEN-ng, we define policy rules as rule-type properties.

3. PAM: an extension of CoMEN

The aim of Computer Supported Cooperative Work is to facilitate cooperative activities among several interacting participants. CoMEN pays a great attention to interactions among cooperative individuals while PAM adopts the concepts in intelligent agent systems and DEN-ng policy structure. For more details about PAM, readers can refer to [38]. Therefore, the following steps are recommended for developing cooperative applications to assist individuals identifying awareness.

3.1. Scenario analysis

This step, borrowed from CoMEN, provides a top–down study of given cooperative work. It encompasses activities relating to requirement analysis. CoMEN defines a method laid on the foundation for the holistic analysis of information systems, incorporating human factors. CoMEN provides “overall system study”, “process study” and “logical component identification”. While we are adopting the concepts associated with PAM, we apply the principles of general system theory. In general system theory, a system is a set of interrelated parts. Therefore, we cannot understand the system as-a-whole by understanding the components, because interactions between components affect the system as-a-whole. Therefore, we study the system “as-a-whole” i.e. holistic view and then we zoom in and study the system as a composite i.e. composite view. Taking this into consideration and applying the practical concepts of CoMEN, we propose the following phases for scenario analysis:

Phase 1: Holistic system study: This phase is equivalent to “overall system study” in CoMEN. The objective is to capture the cooperative environment including activities, policies, initiative events, and situations carrying various system properties.

Phase 2: Holistic process study: This phase is equivalent with “process study” in CoMEN. The objective is to capture possible scenarios. In each scenario, this phase identifies a sequence of situations derived from Phase 1. In each of these situations, holistic process study also finds the truth or falsity of the system properties identified in Phase 1.

Phase 3: Composite system study: This phase is equivalent to the “logical component identification”. The objective is to capture the different roles involved in the cooperative work. This phase also presents each role in terms of a set of possible plans that the role can take for different situations as well as the set of actions that role is able to perform. Composite system study also identifies the available forbidding, permitting, requiring and deterring policy rules for the different roles.

Phase 4: Composite process study: This phase presents the “cooperation enhancement” in CoMEN. The objective is to capture the required awareness as well as available information in each process identified at Phase 2. This phase deploys the process by which roles know when a policy is relevant to the situation, if it’s relevant then the conditions in the policy represent items of what to be aware, so take actions to determine the truth or falsity of those conditions.
3.2. System design

This step compasses the design and implementation of cooperative applications. The research on application of information technology in CSCW state that use of intelligent agents is beneficial to assist roles. The cooperative application, as the output of this step, must support a knowledge based environment. This environment facilitates cooperation considering relevant information leading to the successful resolution of the scenarios. PAM is based on literature of intelligent software agents and distributed policies. The phases taken in PAM are followed the scenario analysis phase in CoMEN. Here, we have explain the four different models of systems in PAM:

**Phase 1: Holistic model at design-time:** The objective of this model is to capture the macro-level features of the system at design-time. Fig. 1 shows the holistic model at design-time. In this phase, we present possible situations as possible worlds. In a set of all possible worlds, each world can be connected to the other worlds. This realizes a single past multiple-future perspective; as time goes by, an agent can take one of the possible actions and accordingly go to one of the possible worlds. At each world, there is a set of events that are possible to receive. Propositional properties present propositional sentences in the system. The value for a propositional property, or in short property, can be true or false. The interpreter holds a set of true sets for each property. A true set is a set of worlds that a property is true at them. Policies, in PAM, are modeled as properties. Therefore, at a given world, a policy can be true or false. This gives us the opportunity of modeling the applicability of a policy in a given world. Therefore, a policy is applicable in a world, if and only if it is true at that world.

**Phase 2: Holistic model at run-time:** The objective of this model is to capture the macro-level features of the system at run-time. Fig. 2 shows the holistic model at run-time. In this model, the path is defined as a sequence of accessible worlds from the current world. A fact is a valued property in a given world. Properties present propositions that can be true or false, but fact are those propositions, at a given world, which we have already calculated their values. As we mentioned above a property is true in a given world, if and only if the interpreter has the world in its true set for the property.

We design the structure of the worlds according to branching time model for agents [32], with a single past and multiple tree-like futures, called branching-time model. In a branching-time model, the worlds are connected together by Actions. An example of branching-time model is given in Fig. 6. Therefore, actions can be defined as connectors between different worlds.

We add unary modal operators next and eventually while we calculate the value of a propositional property. A propositional property can be true in the next time instant if it is true in the next world. Similarly, a proposition can be “eventually”
true, if eventually there is a world in the future that the property is true at that. We also define operator Until such that we say a property is true until another one is true if and only if the first property is true till the second one becomes true. Readers should refer to [26] for more details.

We may have situations where we are not sure what will happen in the future, i.e., there can be more than one possible path. Therefore, we define two additional operators to represent the modality of a formula [26]: (1) inevitable, which is to be true of a formula at a particular point in a branching-time tree if the formula is true of all paths starting from that point. (2) We also define optional, which is true for a formula at a particular point in a time tree if it is true of at least one path starting from that point.

Phase 3: Composite model at design-time: The objective of this model is to capture the micro-level features of the system at design-time. Fig. 3 shows the composite model at design-time.

All actions are inherited from the class Basic Actions, which allows parallel execution. Executable actions are inherited from the class Action. They implement what the actions are supposed to do. Each agent holds its current world, the interpreter and a plan. A plan is defined as a set of rules. Each rule has an action and conditions. Conditions comprise properties, events, and worlds in which the rule would be applicable. The plan indicates to the agent that what action is to be performed at the current world. Plans are represented by a branching-time model of the worlds.

As shown in Fig. 3, the forbidding, requiring, deterring and permitting policy rules inherit from the class Property, and they implement the interface Policy Rule.

Phase 4: Composite model at run-time: The objective of this model is to capture the micro-level features of the system at run-time. Fig. 4 shows the composite model at design-time.

Based on the logic of general awareness, an agent implicitly knows that a propositional property is true if and only if at the given world there is a fact associated with the property, which is true. Facts and properties are explained above. Similarly, the agent does not implicitly know a property if and only if at the given world, the associated fact is false. The agent is aware of a property if and only if the property is relevant. An agent explicitly knows a property if the agent has implicit knowledge as well as awareness to the property. Therefore, if the agent does not know the property or is not aware of the property, it consequently does not explicitly know the property. As such, being aware of a property does not mean that the agent implicitly knows the property. It only means that the property is relevant to the agent at the world.

We define Done Action to be an action that has been performed to transfer the agent from a world to the current world. We also add the method Done that returns true if and only if the agent has already performed the associated action.

When an agent receives the event, it puts it in the Received Events list of the world. Therefore, we define the method Received if and only if the given event is included in the list of the agent at the current world.

We have so far addressed how to develop a system based on PAM using agents. However, we have not said how agents become aware of propositions. PAM defines the notion of a policy-aware agent and proposes a three-step process to (1) recognize relevant policy rules, (2) recognize the relevant information in the relevant policy rules, and (3) change behavior based on the recognized relevant information. PAM uses policy rules as a source to obtain awareness.
Step one: Recognize relevant policy rules: The objective of this step is to show how agents identify a policy rule as relevant to a given situation. We define policy-aware agent as an agent that obtains awareness of a policy rule when it identifies there is a possibility that it may violate the rule sometime now or in future. We now provide the formal definition of a policy-aware agent.

Definition: A policy-aware agent is an agent who obtains its awareness to a policy rule if and only if the agent recognizes a possibility to violate the policy rule now or in the future.

Given this definition, when the agent is not going to violate a policy rule, the agent simply follows its plan. Because permitting and deterring are not in force [28,31], violating the policy rules can only happen in forbidding and requiring policy rules.

Based on the definition of policy-aware agent, and considering that only requiring and forbidding policy rules are in force, we identify two situations in which an agent violates a policy rule. Therefore, in the following situations, the agent needs to become aware of the rule:

The agent receives an event referenced in a requiring policy rule, upon which there is a possibility now or in the future that the agent will implicitly know that (a) the policy rule as a property is valid, (b) the event has been received, (c) the policy’s condition is satisfied, and (d) there will be an option to not do the required action given in the policy rule. In such a situation, there is a possibility to violate the policy rule.

The agent receives an event referenced in a requiring policy rule, upon which there is a possibility now or in the future that the agent will implicitly know that (a) the policy rule as a property is valid, (b) the event has been received, (c) the policy’s condition is satisfied and (d) there will be an option to do the forbidden action given in the policy rule. In such a situation, there is a possibility to violate the policy rule.

Step two: Recognize the relevant information in the relevant policy rules: The objective of this step is to show how agents become aware of relevant information that is required to invoke the policy rule. If a policy-aware agent has received an event referenced in a policy rule, it identifies information awareness from the conditions referenced in the rule. In the logic of general awareness, if an agent is aware of other agents’ awareness, or aware of their implicit or explicit knowledge about a propositional property, then the agent becomes aware of that property. Therefore, if an agent is aware that another agent aware of a policy rule i.e., awareness of awareness, then it will be aware of the event and condition of that particular policy rule. This awareness can be useful in cooperative work.

Step three: Change behavior based on recognized relevant information: The objective of this step is to show how awareness can change the behavior of agents. We have so far discussed recognizing the relevance of a policy rule and information but we have not addressed how awareness updates knowledge leading to a change in an agent’s behaviors.

In order to update knowledge based on awareness, a volitional agent has been proposed by [31]. The volitional agent eventually will implicitly know that a property is true or false, if and only if it is aware of the property. Following this strategy, being aware of a property, the agent selects a path that takes it to the implicit knowledge about this property. The problem is that there might be more than one path to this knowledge. Therefore, proposing a shortest path strategy, we define that policy-aware agent eventually comes to implicit knowledge of the property by the shortest path possible, if and only if it is aware of that property.

3.3. From scenario analysis to system design

Table 3 presents how we map between concepts in scenario analysis and system design. This gives us the opportunity to develop the constructs in system design based on what are analyzed in scenarios. Ray et al. [33] state that CSCW applications
developed based on CoMEN should be heavily based on scenario analysis. In fact, the relevance of information to create new interactions can be found by analyzing different scenarios in the system. Taken this point into account, we propose Table 4 to map concepts associated with scenario analysis to the constructed used to develop cooperative application. In a world, the agent’s neighborhood is a set of worlds that the agent has access to their models. Intuitively, neighborhood of an agent, in a world, means a set of worlds that the agent has access to their information. The agent implicitly knows information if and only if it is available in the agent’s neighborhood. This gives the intuition that agents, in the same world, can have different knowledge based on their neighborhoods. Explicit knowledge has also been added to the system design. The agent explicitly knows a property, if and only if the agent is aware of that and implicitly knows it.

4. Case study: wireless communication system at St. Olavs Hospital

In this section, we present an interpretive case study [34] conducted at St. Olavs Hospital, Trondheim University Hospital, Norway. We found physicians were being called by wireless devices when they should not be interrupted. In this section, we are going to apply PAM and its methodological guidelines to develop a cooperative management application for wireless communication at the hospital.

4.1. Scenario analysis

In order to conduct the first step of the methodological guidelines, one of the authors moderated an interpretive data collection [34]. This consists of participatory observations, non-structured and mostly ad-hoc interviews, and discussions. The study was conducted among a selected group of physicians at various levels of hierarchy and roles. Regarding sensitive information gathered during the study, a non-discloser agreement had been signed before data collection.

Observations: The moderator followed the independent work of 11 physicians at the clinics, for a total of 135 hours in May and June 2009. The moderator took the role of a first year medical student to collect in-depth data. He followed each physician in their everyday work for at least two workdays/nights/duties. The moderator contacted each physician to make an appointment in the morning meeting at each clinic. The moderator recorded every call/page/message, type of device, reaction and physicians’ situation.

Interviews and discussions: During the observation at each clinic, the moderator had an open office with other assistants. This gave opportunities for several discussions on communication scenarios that the moderator discovered. The moderator did not use any pre-defined interview guidelines. However, he asked related questions to what had been observed. The moderator asked two types of questions from interviewees: (1) specific questions to the role of interviewees, (2) similar question to everybody and then he compared the answers.
4.1.1. Holistic system study

St. Olavs Hospital in Trondheim, Norway, has 950 beds with more than 90% accuracy rate, and treats 413,000 patients a year. It is aiming to become the most technologically advanced hospital in northern Europe by 2014 [35].

The hospital recognizes the information and communication technologies as the key for becoming patient-centric. St. Olavs is using a single, converged Internet Protocol (IP) network based on Cisco 7921G for wireless communication devices to improve productivity, decrease operating costs, and enhance the potentials for service integration. The physicians carry at least a wireless IP-phone, an on-call-duty pager and a private mobile phone. Some of them, mostly assistants, also carry a personal pager. However at evenings and nights, a backup GSM phone has to be taken. To use the wireless IP-phone, the physicians have to personally log on to the phone, and if they have an administrative role, they have to log on the administrative role as well.

Here, our focus is on wireless communication among health professionals. We focus on communication notices during outpatient examination duty of physicians, which is a common activity in both clinics. The initiative events in the hospital are related to calling a health professional or informing the availability of people. The moderator found that there is a clear policy about the way the communications should be managed. However, this policy is often unused. The policy says that health professionals, working in the hospital, should try their best to not interrupt physicians when they are not available. Therefore, the system property that the moderator considered as the most important factor in communication is availability of the person that is being called and a list of delegators that can answer the calls instead of the physician. The moderator identified different situations that should be considered as wireless communication interruptions. Some of these situations are listed in Table 4.

\[ \text{Table 4} \]

<table>
<thead>
<tr>
<th>Situation Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$ The nurse is taking care of patients.</td>
</tr>
<tr>
<td>$s_1$ There is an issue needs the physician consultation.</td>
</tr>
<tr>
<td>$s_2$ The nurse is waiting for the availability of the physician.</td>
</tr>
<tr>
<td>$s_3$ The physician is busy with outpatient examination duty.</td>
</tr>
<tr>
<td>$s_4$ The physician has been interrupted by a call.</td>
</tr>
<tr>
<td>$s_5$ The physician has already finished the phone.</td>
</tr>
<tr>
<td>$s_6$ The physician is available.</td>
</tr>
<tr>
<td>$s_7$ The physician is not available.</td>
</tr>
<tr>
<td>$s_8$ The call has been forwarded.</td>
</tr>
<tr>
<td>$s_9$ The call has been made to the physician.</td>
</tr>
</tbody>
</table>

4.1.2. Holistic process study

The moderator, during the data collection, mainly on his observations found several scenarios related to interruption via the wireless communication in the clinics.

Here, due to space restrictions, we focus on one of the observed scenario that happened once the moderator was observing a physician examining a patient ($S_3$). The doctor was in sterile dressed and gloves. The physician received a call from a nurse ($S_4$), the moderator observed that the physician stopped the examination, took off his gloves, and answered the phone. Then, after finishing the call, he washed his hands and treated the hands in antibacterial liquid, took the new gloves on, and started the examination all over again ($S_5$).

The moderator asked the physician for his opinion about the call. The call was about the medicine that one of the patients should take, which could be answered by another person. The physician believed that most of the phone calls that he receives during his examinations can be answered by his available colleagues. As a matter of the fact, the physician was not available and the call should have been diverted to another physician. To do so, the nurse should have asked for the availability of the physician ($S_2$) from the communication center, before calling.

4.1.3. Composite system study

Looking at the communication system and based on the scenario that we analyzed, the moderator identified the role of the physician and the nurse. The moderator found that the nurse could simply asked the communication center about the physician's schedule. In St. Olavs Hospital, there is a list of delegators for each physician that has been provided to nurses. The list states to whom the call can be forwarded if a physician is not available. The moderator was told that in practice, once an issue comes, if the doctor is not available and he has already assigned a delegator, the caller must call the delegator instead.

4.1.4. Composite process study

The moderator found that one of the problems is that the availability of physicians is only stores at the communication center. Therefore, anybody that wants to know the availability of a physician has to contact the communication center. In the scenario, the nurse received the issue that needed the physician's consultation. At the time she made the call, the information about the availability of the physician was not present to the nurse. However, she could understand that if possibly the physician was not available then she was going to violate the aforementioned policy. Therefore, she should have taken the policy rule relevant and becomes aware of that. Being aware of the rule, she also became aware of the availability
of the physician as relevant information. However, she did not know whether the physician is available or not. This required cooperation enhancement in which the nurse could contact the communication center to realize the availability of the physician before calling him.

The cooperation enhancement could be automatically realized by assisting the nurses with an intelligent cooperative application. Once the nurse received an issue which needed calling the physician, it could have looked at the policy rules and become aware of the availability of the physician. Therefore, the application automatically would contact the communication center. If the physician was not available, the application would divert the call to the delegator. In the following section, we show the system design step for such an application.

4.2. System design

Having done the scenario analysis, we follow the system design steps to develop a cooperative application. This assists health professionals to understand the relevance of the physicians’ availability before calling them. We follow PAM to develop the system. The agents use policies as a source for obtaining awareness. Fig. 5 shows the system design phase for the mentioned scenario in St. Olavs Hospital. The system design, as explained earlier, is conducted based on the following phases: (1) holistic model at design-time, (2) holistic model at run-time, (3) composite model at design-time, and (4) composite model at run-time.

In the holistic model at design-time, the different situations of the system are presented by worlds. Those worlds that directly derived from holistic system study are presented by w_s, while those showing the change of the system environment are presented by w.

Table 5 shows the list of the worlds. In fact, neighborhood defines that agents can be in the same world, although they implicitly know about different worlds. As such, they may have different knowledge. The considerable properties in this system are two propositions: the physician is required, and the physician is not available. The interpreter stores the worlds in which these properties are true. There are two different initiative events: (1) an issue with the need of physician consultation, (2) call.

Based on the holistic process study conducted in scenario analysis, Fig. 6 presents the branching-time model of the system. The figure shows how the worlds are connected together. When an issue with the need for the physician consultation comes up, the system goes to w_1. At w_1, the nurse can simply call the physician (w_8) or asks the communication center for the availability of the physician. If the physician is available (w_3), the call will be made (w_5) and the physician will answer (w_7). If the physician is not available (w_4), then the call will be forwarded (w_6).

In composite model of the system at design-time, we identify nurse agent, physician agent and communication center agent. While the agents are in a certain world, the neighborhood presents all the worlds that each agent knows about. In fact, an agent in a world knows about some worlds that may be different than the worlds that another agent in the same world knows about. We also identify the different actions that can be carried out by each agent. The composite model of the system at design-time presents the plan of each agent based on rules. The rules say which action should be taken by the agent according to the received event and the captured propositional properties. Plan can be presented in terms of branching-time model [32].

The composite model of the system at run-time presents the mental attitudes of agents such as awareness, implicit knowledge and explicit knowledge. Looking at the system in w_1, PAM proposes that the nurse agent should take the following three steps to obtain awareness:

Step 1: Recognize relevant policies: The nurse agent implicitly knows that the physician is required. However, the agent does not know whether the physician is available or not. Following the nurse agent’s plan, if it simply calls the physician, it goes to w_8. Looking at the branching time model and the nurse agent’s plan, the agent finds out that there is a possibility at w_4 that the agent implicitly knows about the physician's unavailability. Therefore, by calling the physician and going to w_8, there is a possibility for the agent to do the forbidden action given in the mentioned policy rule. The policy rule says once an issue comes which requires a physician consultation, if the physician is not available; the nurse must not call the physician. Therefore, calling to the physician when he/she is not available is forbidden and breaks the policy rule. As such, the agent takes this policy rule as relevant and becomes aware of that (see Fig. 6).
Step 2: Recognize the relevant of information: Being aware of the policy rule, the nurse agent needs to implicitly know the conditions for the rule, which makes the availability of the physician relevant. Therefore, the nurse agent needs to become aware of the physician’s availability.

Step 3: Change behavior based on recognized relevant information: Being aware that the physician is not available, the nurse agent needs to implicitly know whether “the physician is not available” is a true or false property. Therefore, it performs an action that eventually brings the agent to this knowledge. The nurse agent asks the communication center and goes to w2. This is because that the truth or falsity of the property is given in w4 and w3, which are accessible from the path begins from w2.

In the next section, we present the overall deployment of the system on the wireless communication at the hospital.
4.3. Deployment

PAM can be developed using intelligent communication systems such as Cisco Call Manager. An objective of intelligent communication systems is to provide communications processing that includes (a) a Service Logical Program (SLP) to receive users’ requests and (b) a Service Logical Interpreter (SLI) to execute the users’ requests and return the results to the users. The SLP and SLI can be implemented by software agents. This implementation can benefit from using PAM to recognize the relevance of information and request it from the users. For example in the wireless communication system at St. Olavs Hospital, the nurse uses a SLP agent that is implemented by the PAM. This would give him/her the ability to identify the relevance of the physician’s availability. The SLI can be implemented on top of the Cisco Call Manager server, which stores the availability of physicians in the hospital.

Fig. 7 shows the mentioned scenario as an example on how the developed system can be deployed on the Cisco Call Manager at the hospital. Every day, the physicians give their availability schedule to the communication center via their wireless phones. Then, once a nurse dials the physician, the wireless phone checks the availability of the system with the SLI agent sitting on the server at communication center. If the physician is available, the nurse SLP agent will continue dialing. If the physician is not available, then the nurse agent based on the uploaded list of delegators calls an alternative physician. Complying one of the procedures at the hospital, the delegators are people who must be available at any time. Therefore, the nurse agent does not need to check the availability of the delegator.

4.4. Experiments on the wireless communication system

We have, so far, explained how we can develop the wireless communication system for the hospital following the methodological guidelines in PAM. In this section, we run some experiments on the developed system.
Table 6  
Configuration parameters.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>The number of actions, or in the other words, the number of worlds.</td>
</tr>
<tr>
<td>Complexity</td>
<td>The arrangement of the worlds meaning how flat the branching-time model of the system looks like. We measure the complexity by the maximum number of the worlds coming from the same world in branching-time model. Therefore, the more horizontal worlds the branching-time model has, the more complex it is.</td>
</tr>
<tr>
<td>Availability of policy rules</td>
<td>The number of the policy rules that have been triggered by events and actions involved in the scenario.</td>
</tr>
</tbody>
</table>

4.4.1. Methodology

In the observations, interviews and discussions, the moderator found 43 different scenarios. These scenarios were our input data for the experiments. By varying each of the parameters in Table 6, the success of the system can be tested for experimental purposes. In each experiment, we sorted the scenarios based on the selected parameter from Table 6.

We compare developing SLP and SLI agents using PAM against using plans without awareness. We measure the success of agents in each scenario as achieving a given certain world.

4.4.2. Size

We measure the size of the scenarios by number of different worlds involved in the scenario. We divided the scenarios based on the number of the involved worlds. There was a range of 4 to 26 worlds contributing in the scenarios. We repeated the experiment for each specific number of the worlds 100 times with random inputs to the scenarios. We ran the system with awareness approach and without awareness. Fig. 8 shows the results of the experiment.

The experiment shows that regardless of using awareness, increasing the size of the scenarios drops the success of the system. In the other words, increasing the number of the worlds in a scenario increases the need for identifying the relevance of information to decide which action should be taken. Fig. 8 shows awareness curve stays above the other one. This shows the positive impact of using awareness on success of the system. In Fig. 8, by increasing the size of the system, the curves diverge from each other. This shows that despite the fact that increasing the number of the worlds decreases the success rate, PAM becomes more beneficial by increasing the size of the system.

4.4.3. Complexity

We define the complexity of a scenario as the number of the actions that can be taken at the world. Therefore, we measure the complexity of the scenarios by the maximum number of the horizontal worlds in the branching-time model. In this experiment, we categorized the scenarios in regard to the maximum number of horizontal worlds. We could find scenarios that were flat, while there were also scenarios with 7 horizontal worlds. We repeated the experiment, for each specific complexity, 100 times with random inputs to the scenarios. We ran the system with awareness approach and without awareness. Fig. 9 shows the results of the experiment.

The experiment shows that by increasing the complexity of the scenarios the success rate falls. In the other words, increasing the number of the horizontal worlds in a scenario clearly increases the number of possible actions that can be taken. Therefore, the need for identifying the relevance of information grows. Fig. 9 shows awareness curve stays above the other one, which illustrates the positive effect of using awareness on success of the system. In Fig. 9, by increasing the complexity of the system, the curves diverge from each other. This means that the drop of success is more radical without
Fig. 9. Effect of complexity on success; comparing awareness approach with not using awareness.

Fig. 10. Effect of availability of policy rules on success.

4.4.4. Availability of policy rules

The moderator during his observation, interviews and discussions at St. Olavs Hospital found that the health professionals at the hospitals are the following several rules. He identified 24 rules related to the selected communication scenarios. We ran the system developed by PAM with zero policy rules, with one policy rule, with two policy rules and so on to 24 policy rules. At each time, we ran all the 43 identified scenarios and recorded the success of system. Then, we calculated the success rate. Fig. 10 shows the results of this experiment. Increasing the number of policy rules gradually lifts up the success of the system. This confirms the basic idea of PAM saying that using policies as a source to obtain awareness is effective. In fact, increasing the number of policy rules improves identification of relevant information that the agent may need to select among different actions.

4.4.5. Summary and outlook

The first experiment indicates that by increasing the size of the system, the success rate has an overall decrease. Fig. 8 shows the success faces a more dramatic fall in bigger systems. Therefore, by increasing the size of the system, the rate of success drops more tremendously. The figure illustrates that using PAM in comparison with ignoring awareness is a beneficial approach. The gradual divergence between two curves indicates the role of awareness becomes much more significant in bigger systems.
The second experiment shows that the structure of branching-time model has a considerable effect on success of the system. The flatter branching-time model has less possibility of actions. Therefore, complexity drops the success rate. Fig. 9 shows a general improvement by using awareness. This figure also exhibits a gradual divergence between two curves by increasing the number of horizontal worlds in the branching-time model. This demonstrates the benefit of using PAM in more complex systems, where standard approaches without awareness suffer a significant fall in success rate.

The third experiment shows that the system has an improvement by increasing the number of policy rules when we use PAM. The basic idea of PAM – the use of policies as a source of awareness, is seen to be beneficial. In fact, because PAM is working with policies, increasing the number of policy rules has a positive effect of the success rate.

5. Discussion and conclusion

In this section, we present the contributions of PAM in theory as well as in practice. We also present the limitations of PAM that can serve as directions for future research in the field.

Community websites and social networks tools profoundly affect many facets of people's lives. These types of tools are supported by information technology and provide a means to cooperation while they recommend certain types of decisional information leading to a particular behavior for individuals. However, the recommending tools bombard the participants in the community by a large amount of irrelevant or loosely relevant information. Therefore, the lack of a definitive method to recognize the relevance of information in recommender tools can be obviously seen. PAM provides a method to identify awareness in cooperative application.

We borrowed the concept of awareness from CoMEN. However, there is currently no definitive method for recognizing the required awareness. Given policy-aware agents, participants in cooperative endeavors can recognize the relevance of policy rules and the information that they require to enhance cooperation.

PAM is based on the logic of general awareness. While the literature on this logic emphasizes the importance of a method to identify awareness, PAM proposes policies as a source to identify and obtain awareness resulting favorable behaviors.

Although Directory Enabled Networks – next generation (DEN-ng) is being used to implement awareness in agents, the use of policies as a guideline to find the required information for awareness has been ignored. This gap is addressed by PAM. In fact, DEN-ng policy structure gives a meta-model in which a given set of awareness can be implemented by generating and trigging DEN-ng policy rules. However, PAM is not a method to generate policy rules. Instead, given a set of policy rules, it identifies the relevant information into the situation; that is awareness.

PAM can be developed with intelligent communication systems. As we explained in the case of St. Olav Hospital, the implementation of the Service Logical Program (SLP) can be benefited from PAM. This gives the ability to the participants to identify the relevance of information. It decreases the unnecessary communications, as shown in St. Olav Hospital case.

Research in software agents typically addresses awareness in terms of programming intelligent agents. Such software agents should reason and make suggestions to assist cooperative roles. The case of St. Olav Hospital shows that programming software agents have two weaknesses which can be solved by considering policies in agent-base systems: First, the suggestions made by software agents are dependent on their understanding of the situation and limited by their implementation. As such, agents are not able to recognize the relevance of information to a situation. Therefore, policies can act as guidelines. Second, standard approaches to software agents are technology dependent, but involved cooperative roles often use different technologies simultaneously. Therefore, integrated cooperation can be difficult to achieve. Policies can help as integrated common guidelines, which are given to all the roles in cooperation.

There are four limitations associated with our approach that we would like to point out. These can be the direction for the future research in the field.

PAM is not a process to design policies; rather, it assumes a given set of policies. Therefore, PAM relies on the quality of the policies, which indicates the following issues:

- **Interactions and conflicts among policies:** policy rules may interact with each other, and a new added policy rule may conflict with existing previous rules.
- **Refining high-level policies to computational policy rules:** high-level policies in ordinary English need to be translated into machine-readable form before applying PAM.

PAM is untried in mass production environments. However, an initial proof of concept as an exemplar is made in the wireless communication system at St. Olav Hospital. While exemplars are a common way to provide initial validation to a new method in software engineering, further analysis, productions, and real-world experimentation are needed for the future research [36].

The paper as explained in Section 1 aims to extend CoMEN as a methodological approach for development of agent-based cooperative applications that are able to identify relevance of information. Since CoMEN itself lacks fundamental formalism, in order to formalize PAM we need to ground a framework to define formal semantics. This will complicate the paper while it is not in the scope. The main objective here is to provide a software engineering methodology to develop cooperative applications. For detailed formalization of the approach, refer to [37].
Acknowledgments

A part of this research is supported by the Research Council of Norway, grant No. 176852/S10. We would also like to thank all the physicians at St. Olavs Hospital for all their helps and collaborations.

References