# A novel context-aware augmented reality framework for maintenance systems

S. akbarinasaji[a] , E. homayounvala[a,*]
[a]*Cyberspace Research Institute, Shahid Beheshti University, Evin, Tehran, Iran*

**Abstract. A**ugmented Reality (AR) bridges the gap between real and virtual world by bringing virtual information to real environment as seamlessly as possible. The need for better perception of knowledge-intensive complex maintenance tasks and access to large amounts of documents and data makes the use of AR technology promising in a maintenance domain. Context-awareness enhances the usability of such AR applications, i.e. the output and behavior of the system will be adapted according to different contexts such as user location, preferences, devices, etc. to afford higher level of personalization. The adaptation needs to be efficient in terms of performance and speed. This paper presents an optimized framework which combines context-awareness and AR for training and assisting technicians in maintaining equipment in an industrial context to improve field workers effectiveness. Ontology is used to model a maintenance context and Semantic Web Rule Language (SWRL) provides logical reasoning. This optimized framework utilizes a behavior network to select suitable actions collection based on ongoing task current step and apply context-based inferred information from ontology to each member of this collection. Evaluation results comparing the performance of the proposed framework with conventional ontology alone in a maintenance domain confirmed that the proposed framework in this research provides the same results as ontology in terms of content, but it runs much faster in terms of run-time and performance. The proposed context-aware framework is quite valuable especially in case of response time and performance of maintenance systems with large number of maintenance activities.

Keywords: Context-Awareness, Augmented Reality, Behavior Network, Ontology, Operation and Maintenance

## 1. Introduction

Efficient operation and maintenance guarantees reliability, safety, and energy saving of equipment. Operation and maintenance includes actions required to upkeep equipment and facilities and helps to lengthen equipment lifecycle [1][2]. The complex structure of equipment face technicians with difficulties to conduct maintenance activities [3] so an access to technical paper-based manuals is required especially when these activities are performed infrequently [4].

Computing technologies has enabled developers to support maintenance activities faster, more reliable and more intuitive. As a consequence traditional paper-based manuals are on the verge of extinction [5]. AR technology which has been used in various domains including maintenance [6] is the technology which can be employed to assist users completing a maintenance task steps by overlaying virtual objects on a real-time equipment scene and providing human-machine interaction as well [7][6]. AR technology improves user's perception of maintenance activity by superimposing multimedia alerts and providing bi-directional visual interaction between the user and the system [2]. However context-awareness which is an integral part of improving systems usability needs to be employed in order to adapt the system's behavior and output based on user's context [3].

Context is defined as a set of states and settings in the environment that determines an application's behavior or in which an application event occurs and

*Corresponding author. Tel. (+98)-21-29905482; E-mail: e_vala@sbu.ac.ir.

is interesting to the user [8]. Context can be divided to three groups, namely, computing context (network bandwidth, communication cost, etc.), user context (user profile, people nearby, etc.) and physical context (light, noise level, etc) [9]. To make it clear some examples of context is stated as follows. A user location can be used to provide alerts on future preventive jobs about nearby equipment to the user. As another example user's expertise level, defined as user context, affects details of the information to be shown, i.e. a user with high expertise level needs less details to perform a task but a novice user requires more details to help him finishing the job. Or suppose a user which runs the system on his tablet device. If the battery level of his device goes below a certain value for example 25% or the CPU usage of his device goes higher than a threshold for example 85% then the system should avoid overlaying 3D models which consume more battery and need more processing resources, instead textual overlay should be used.

In addition one should keep in mind that the goal of providing context-awareness within AR-based maintenance systems is to improve efficiency and reduce the time, cost and effort required to complete maintenance tasks, but achievement of this goal should not affect the performance of the proposed systems in case of run-time and resources, i.e. mentioned systems need to be able to run as fast as possible even on devices with limited computational resources and should not take an intolerable time to provide output for the final user.

This paper presents an optimized context-aware AR-based framework to assist technicians conducting maintenance tasks. The proposed framework makes use of collected raw context data and perform reasoning on them to obtain new inferred data and provide adaptive behavior using an ontology. On the other hand the responsibility of selecting the collection of information instances to provide assistance to the technician for each step of ongoing maintenance routine is granted to a behavior network which inconsiderably improves the performance of the system instead of running time-consuming SWRL rules.

The rest of this paper is organized as follows. Section 2 gives a review of context-aware systems in various domains. Section 3 shows the architecture of the proposed framework in this paper. An experiment of the proposed framework is presented in section 4. Results of employing the proposed framework is provided in section 5. Section 6 summarizes the paper and introduces future works.

## 2. Related work

Conducting activities including scheduling, periodical controls, performing preventive and predictive procedures to ensure that property and equipment achieve their intended goals, i.e., preventing breakdown, increasing efficiency, reliability and safety is the main purpose of operation and maintenance activities [5][1].

AR technology can be utilized to train complicated procedures and provide maintenance instructions as well as visual interactions to the onsite technicians and it's proved to be useful in a maintenance context [3].

Wang et al developed a framework for an AR-based system to conduct complicated maintenance tasks cooperatively and efficiently through a local network infrastructure between technicians and remote experts [10]. Schwald et al. proposed an AR system to aid technicians in maintaining equipment in operational condition as well as instructing them how to use facilities [4]. AROMA-FF encompasses an AR-based interface in order to superimpose 3D assistive models and a maintenance information model including equipment settings, sub-systems, components, actors, maintenance history, sensor-based live data stream, geometric representations [5].

To reach the goal of improving the usability of AR maintenance systems there is a need to utilize context-aware computing [11] in order to discover contextual information and use them to provide more practical information to the users [8], in consequence, virtual information will be changed according to various context [3] and the intelligent behavior in a smart environment will be improved [12].

Gouin-Vallerand et al. employ different types of contextual information including user profiles, device profiles, etc. and propose a context-aware service provision to be used in smart environments [13]. CoMotion project at MIT exploits GPS-based user location as a context parameter and plays back a recorded message automatically as soon as a recipient arrives at the desired location [14]. CyberGuide, an information provider service, takes advantage of both user location and orientation to provide information such as directions, background information, places of interest and people comments for the tourists [15][16]. A notepad application is developed by Schmidt et al. for a PDA device to adapt the font size and backlight based on the user recognized context including moving state and light level. In another scenario mobile phone profile is changed depending on the current phone state [17]. Oh et al. suggests CAMAR which is a mixture of mobile AR and context-awareness. CAMAR makes use of unified user context information and allows users to view and tailor augmented information to share them with other end users [18]. Mileo et al. present a context model that makes use of information from pervasive sensors placed in the environment and patient explanatory feedback as well to interpret complex collected data and reason about available knowledge using logic programming language in order to support patient well-being [19]. Byun et al. employed context history beside user modeling and machine learning algorithms to provide adaptation. The proposed system collects and learns user preferences about the office window state in different room conditions, afterwards user desired window state will be induced based on gathered data i.e. user behavior [20].

Context-awareness is not widely used due to the lack of effective infrastructure to support context-aware applications [12]. Ontology has been widely used for context modeling which defines concepts in a specific domain and relation between concepts [21]. Ontology provides benefits such as knowledge sharing, logic-based reasoning and inference, and knowledge reuse [22]. Entities and their relations are introduced in form of classes, subclasses and properties in ontologies. Each class can have several subclasses, individuals and properties. Properties are divided to two types, object and data properties. Object properties defines a relation from an individual to another individual. Data properties defines a relations from an individual to a data value such as String, Boolean, Date, Time, etc [23].

A formal general reusable ontology-based context model including default vocabulary and extendable structure is provided by Hoque et al. to support ontology-based reasoning and rule-based reasoning in a smart home environment [24]. CONON takes care of context modeling and ontology reasoning in addition to user defined reasoning in a pervasive computing environment. It defines a basic context ontology consist of general concepts and the ability to extend ontology for a specific domain hierarchically [22]. A combination of first order probabilistic logic and web ontology language (OWL) is implemented as an agent-based middleware by Qin et al. to gain contextual information, knowledge demonstration, usage and sharing in context-aware applications [12].

Ong et al. present a novel ontology-based context-aware AR maintenance system to support provision of context related information to the technicians. In addition the system provides a real-time collaborative interface for technicians to ask for help from remote experts or vendors as well [2]. Another context-aware maintenance application based on ontology equipped with AR technology with the ability to run on a tablet PC is introduced by Flatt et al. This software superimpose maintenance information beside live operational data as virtual information on screen and allow users to attach sticky notes to the equipment components enriched with position information as well which make them recognizable later even in different camera positions [7]. Zhu et al. introduced an Authorable Context-aware Augmented Reality System (ACARS) using ontology as a context modeler to support technicians in maintenance jobs and enable them to interact with AR contents. As a result they can create new augmented information themselves or revise formerly created information [3]. Authoring for Context-Aware AR (ACAAR) is another authoring tool which afford approaches to create augmented context-aware information in maintenance AR applications with little programming skills employing the help of ontology. ACAAR allows the technicians to correlate virtual information with different contexts [11],

Behavior Network takes responsibility for the problem of action selection in an agent-based environment. Each agent faces a collection of competence modules. Changes in the activation level of each module will lead to its activation. The agent will select the modules with the highest activation level greater than desired threshold [25].

Song and Cho present a context-adaptive user interface for the simplification of using different control devices containing different functions on each device in a home environment. The proposed system takes advantage of a Bayesian network to predict the desired devices in current user situation and then employed a behavior network to select the required functions at each step hierarchically [26].

Literature review shows that a lot of context-aware systems have been designed and implemented in various domains, however context-awareness has been disregarded in maintenance domain and there is not much research done in this field, thus most of maintenance AR applications have been deprived of promising benefits of context-awareness such as improved usability. But while improving usability due to context-awareness, performance of the system needs to be considered. Conventional ontology-based context-aware maintenance systems are in charge of defining what needs be shown in each step of a maintenance task using SWRL rules. These rules are time-consuming and as they grow the performance of the system decrease. In an industrial environment with

10,000 facilities that each facility has more or less maintenance tasks with 10 steps or more to complete the job, applying rules and inferring the relevant output of each step by using reasoner will be time-consuming and it is not bearable for the user to wait minutes for the system to start working.

In this paper we have proposed a context-aware framework as a base for AR maintenance application. The proposed system employs an OWL ontology for the sake of context-awareness and uses behavior network to select actions to be conducted at each step as a supplement to improve run-time speed. Thus the ability of running application on devices with limited processing resources will be provided and the system starts working in a rational acceptable time.

## 3. System overview

This section describes the process of development of a context-aware AR maintenance system. As stated before AR brings to completion the ability to superimpose maintenance information instances on real equipment and describe virtual equivalent of specific type of information such as paper-based manuals, usage instructions, repair procedures, etc. to the users.

### *3.1. Classification of information instances*

Information instances are those virtual objects to be superimposed on environment using AR. To provide information instances for maintenance systems two questions need to be taken into account. First one is the time when a procedure should be performed and the second one is how a procedure should be carried out. In this research information instances are categorized in three groups based on maintenance activity type which will give the answers to the two questions stated above.

#### *3.1.1. Static information*

In break-down activities a specific routine consists of definite steps should be performed on an equipment that has stopped working due to a failure reason to return that equipment back to normal functional condition. Operational activity however is not a maintenance activity but is included in this research as a maintenance routine containing definite steps advised by the manufacturer on how to run an equipment correctly. These type of information are named as static information considering the fact that they do not vary a lot during equipment lifecycle.

#### *3.1.2. Periodical information*

Technicians need to perform some procedures on equipment based on the manufacturer schedule in order to avoid failure and further break-down cost. This schedule introduces the conditions to perform a job or to send an alert to the technicians. These conditions include predefined intervals such as every eight operating hour, every day, every thousand kilometers traveled, etc. or certain parameters such as pressure, current or noise level higher than a certain threshold. When one of the conditions stated above are met some specific steps should be executed on that equipment. Since information included in this category are performed based on a schedule, they are called periodical information.

#### *3.1.3. Dynamic information*

Monitoring activities make engineers able to control live operational data as well as statistical reports. Due to the fact that these information change every moment, this category is named as dynamic information

### *3.2. AR-based maintenance system*

Different type of information instances described in 3.1 can be presented using AR technology. For this to happen there should exists three phases in each AR system. These phases are shown in Figure 1 namely Identification (P1), Retrieval (P2) and Visualization (P3).

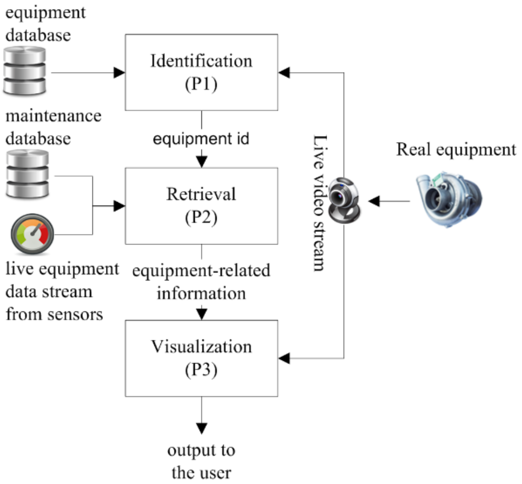


Fig. 1. Phases of AR systems

Identification (P1) is responsible for recognizing equipment in the environment automatically by means of equipment natural features or visual markers attached to equipment employing computer vision-based object recognition algorithms, RFID devices, GPS-based location and orientation, etc. or manual input of user into the system specifying the equipment. The proposed system in this research uses natural features to detect and track equipment in the environment. To do so, the system need to be trained in advance. Therefore an image of the equipment should be fed to the system for the sake of features extraction. Finishing this procedure will create a dictionary containing natural features of equipment which can be used for further detection and tracking purposes. Figure 2 shows an original image taken from an equipment and Figure 3 shows the extracted features of this equipment.

Fig. 2. Original equipment image

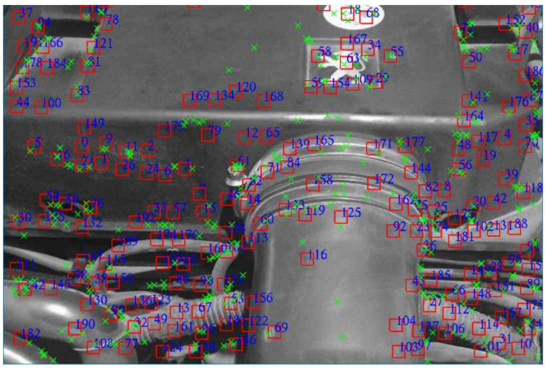

Fig. 3. Equipment image feature set

The second phase in AR-based systems is Retrieval phase (P2) which is in charge of obtaining maintenance related information or live data through various databases and pass them to the next phase. Visualization (P3), the last phase, integrates retrieved information and superimposes them on real equipment scene obtained through live video stream. AR technology fulfills the requirements to implement these phases.

### 3.3. Adaptation scheme

The important fact is that showing static virtual information to the user is not satisfactory. So output information needs to be selected based on user's context such as expertise level, skill name, interested color, location, etc. in Retrieval phase (P2) which context-awareness seek to fulfill in AR-based maintenance systems. Therefore an adaptation engine is considered within Retrieval phase (P2) as shown in Figure 4. This scheme is the basis for the provision of relevant information to the users in the terms of content and presentation.

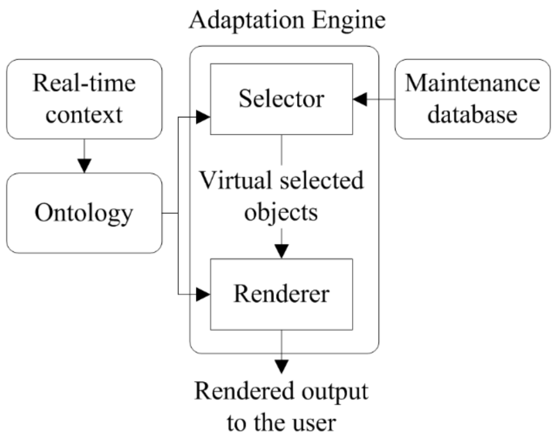


Fig. 4. Adaptation engine

Adaptation engine as shown in Figure 4 is the heart of Retrieval phase (P2) and is divided into two subsections referred to as Selector and Renderer. Selector receives real time context and predefined logical rules and instances from ontology as input. As a consequence of applying logical rules on received context, Selector chooses which combination of virtual information need to be retrieved from maintenance database to be displayed on the scene using a behavior network. This combination is next passed to the Renderer which applies rendering properties such as color, transparency, etc. considering the rules defined in the ontology by system developers. Output generated by this module will be delegate of what is needed to be displayed on the screen.

### 3.4. System architecture

The proposed system consists of five different modules, namely, System Core, Context Collector, Behavior Network, Reasoner, Visualization and databases to provide context-aware information instances to the final user as shown in Figure 5. System Core module is called initially whenever the system starts working and establishes the link between different modules. Context Collector is responsible for providing real time context to the System Core. Behavior Network prepare combination of instructions to be displayed from maintenance database based on the rules step, individuals and properties defined in the system ontology and return them back to System Core module. Applying the rules defined in ontology over individuals and sending back the results to the Behavior Network is the Reasoner duty. Visualization module receives combination of output information from System Core and presents them on the screen. Embedded modules interact with each other as well as some other external libraries. A description of each module is provided as follows.

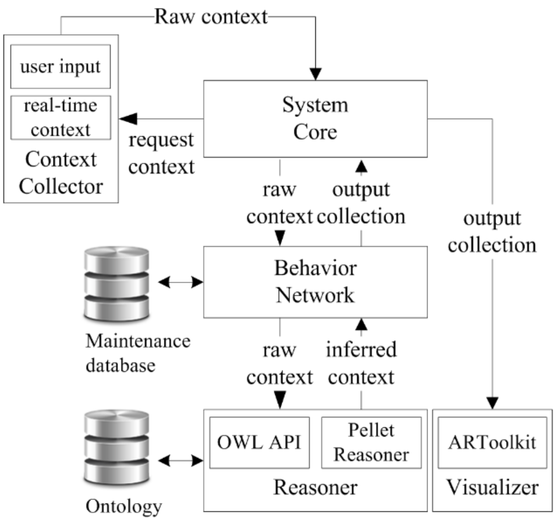

Fig. 5. System architecture

#### 3.4.1. System core module

As Figure 5 illustrates System Core module is the start point of the system and will be called initially whenever the system starts. For the system to be functional there is a need for a controller to provide a link between different modules. This controller fires other modules whenever needed and retrieves results after process completion in the fired module. System Core plays the role of a controller in the proposed system.

First and foremost, System Core calls for Context Collector to obtain real time context. After collection of these raw context data, they are passed to Behavior Network module via System Core to choose output information accordingly. Acquired combination of information will be passed to Visualization module in order to be displayed on the screen.

#### 3.4.2. Context collector module

This module has a main class referred to as Context Provider which is called by System Core module. This class asks for real time context which can be obtained in two methods.

First method is by reading live sensors data such as time of the day, current date, battery level, CPU usage level and etc. automatically. The second method is gathering information through manual user input such as person expertise level, person preferred color and etc. Context Provider class transmits these raw context collected data to System Core for further processing.

#### 3.4.3. Reasoner module

In order to describe maintenance environment, context modeling is needed which defines entities and relations between them. Context modeling plays an important role in the process of implementing context-aware applications in order to provide information according to the status of people, equipment, activity and etc. Following section provides a description of the context modeling used in our proposed system.

*Ontology*

This research adopts CONON ontology proposed by Wang et al. as the base ontology [22]. Some customization and extension is being done upon it to model a maintenance context in order to provide a common definition for certain maintenance concepts and their inter-relations. The proposed ontology in this research is named CAMON as an abbreviation for Context Aware Maintenance ONtology. As illustrated in Figure 6 CAMON consists of five base classes named Activity, CompEntity, Information-Instance, Location and Person. Activity subclasses represents different maintenance activities described in 3.1. CompEntity includes CompDevice, Equipment and SensorDevice as subclasses. CompDevice represents user's computational devices such as mobile, tablet, etc. which is used to run the system during a maintenance activity. Equipment represents facilities that the technician will perform an activity on. SensorDevice holds AR markers placed in the environment. Location is divided to IndoorSpace and OutdoorSpace. Person represents everybody that is involved in a maintenance activity. BDM_Engineer are those engineers who carry out a corrective maintenance activity. PM_Engineer represents those who perform a preventive activity routine on an equipment. Manager is a delegate of persons who are involved in monitoring activities and Operator represents persons who have decided to run an operation on an equipment. InformationInstance holds the related rule name of each activity to be displayed over the scene.

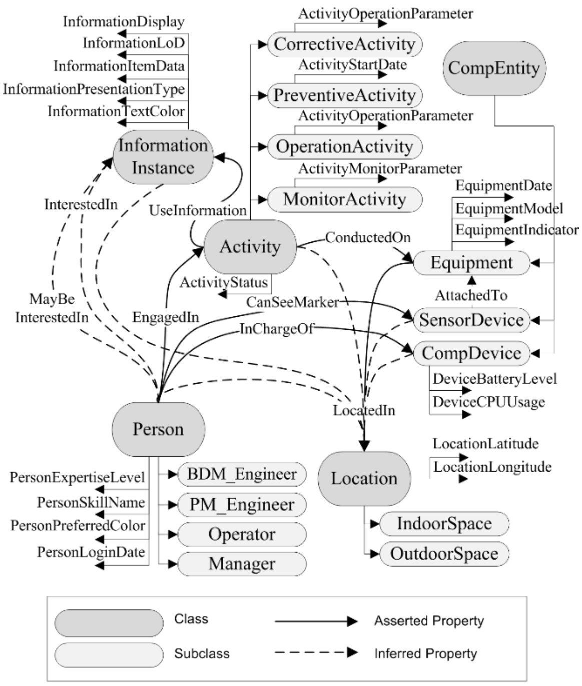

Fig. 6. CAMON ontology

Based on the maintenance environment these classes can be changed or extended and individuals may be inserted into the ontology. For example the definition for ChangeFilter as an individual of class Activity is illustrated in Figure 7. This activity is conducted on an equipment named CarEngine, uses information named ChangeFilter_Rule as information instance, failure reason is stated as DirtyAirFilter and current status of activity is at Step 1.

As another example an individual named John is created as an individual of Person class as illustrated in Figure 8. John is engaged in ChangeFilter activity. He is in charge of iPhone6 device. His expertise level is defined as Low, preferred color is saved as Red and skill name is stated as Mechanic.

```
<rdf:Description rdf:about=”ChangeFilter”>
    <rdf:type rdf:resource=”Activity” />
    <ConductedOn rdf:resource=”CarEngine” />
    <UseInformation
        rdf:resource=“ChangeFilter_Rule” />
    <ActivityFailureReason
        rdf:datatype=”XMLSchema#string”>
        DirtyAirFilter
    </ActivityFailureReason>
    <ActivityStatus
        rdf:datatype=”XMLSchema#string”>
        STEP1
    </ActivityStatus>
</rdf:Description>
```

Fig. 7. Create an individual of Activity class

```
<rdf:Description rdf:about=”John”>
    <rdf:type rdf:resource=”Person” />
    <EngagedIn rdf:resource=”ChangeFilter” />
    <InChargeOf rdf:resource=”iPhone6” />
    <PersonExpertiseLevel
        rdf:datatype=”XMLSchema#string”>
        LOW
    </PersonExpertiseLevel>
    <PersonPreferredColor
        rdf:datatype=”XMLSchema#string”>
        RED
    </PersonPreferredColor>
    <PersonSkillName
        rdf:datatype=”XMLSchema#string”>
        MECHANIC
    </PersonSkillName>
</rdf:Description>
```

Fig. 8. Define an individual of Person class

*Logical rule definition*

Logical rules which describe environment in detail will be defined using SWRL rules. Any SWRL rule consists of two parts named Body and Head. Body contains a set of preconditions and Head contains set of actions which will be taken into account whenever the Body preconditions are satisfied completely. An example of SWRL rule is presented in Figure 8. In this example if atoms A(?x) and B(?x) are satisfied, action C(?x) will be taken and the result will be true. A, B and C can be classes, object properties or data properties and x introduces a variable in this example.

SWRL built-in functions such as swrlb:equal can also be used in SWRL rules.

A(?x) ^ B(?x) -> C(?x)

Fig. 9. An example of SWRL rule

SWRL rules provide the ability for adaptation and context reasoning, i.e. inferring high-level context from raw context data. In this research logical rules are divided into three types:

- Environment Rules: This type of rule defines complex relation between entities and introduces environment in more detail. An example of this type of rule is illustrated in Figure 10. System developers will declare these type of rules based on the environment which the system will run inside.

```
Person(?p) ^ Activity(?a)
    ^ InformationInstance(?i) ^ EngagedIn(?p, ?a)
    ^ UseInformation(?a, ?i) -> InterestedIn(?p, ?i)
```

Fig. 10. An example of environment rule

- Context Adaptation Rules: These rules provide the ability to deduce high level context from raw context data. An example of this type of rule is shown in Figure 11. This example uses real time user's device CPU usage level and says that if the CPU usage of user's device is greater than 80 percent then related information instance presentation type need to be Text-based in order to avoid CPU overload because textual presentation consumes less computing resources.

```
Person(?p) ^ CompDevice(?c)
    ^ InformationInstance(?i) ^ InChargeOf(?p, ?c)
    ^ InterestedIn(?p, ?i) ^ DeviceCPUUsage(?c, ?u)
    ^ swrlb:greaterThan(?u, 80)
    -> InformationPresentationType(?i, "TEXT")
```

Fig. 11. An example of context adaptation rule

Different types of raw context data can be used to infer more usable information and provide a robust adaptation.

- Static Adaptation Rules: The rules which introduce the requirements for a specific maintenance activity to be fired will be described here. Actually these rules are simple ones which says which rule should be considered in which situation. But SWRL rules regarding what should be displayed in each step of the selected routine is omitted in our proposed system and Behavior Network is responsible for this job. An example of static adaptation rule is illustrated in Figure 12.

```
Activity(?a)
    ^ ActivityFailureReason(?a, "DirtyAirFilter")
    ^ InformationInstance(ChangeFilter_Rule)
    -> UseInformation(?a, ChangeFitler_Rule)
```

Fig. 12. An example of static adaptation rule

The rule shown in Figure 12 says that if there is an activity which failure reason is DirtyAirFilter, the related information instance will be ChangeFilter-_Rule.

*Reasoner module*

Reasoner module performs two types of reasoning. First one is Ontology Reasoning. Using the Ontology Reasoning method the reasoner classifies individuals based on their definition and assumes each individual as an individual of a class. As an example, in the ontology it is stated that if an activity has a failure reason it is assumed to be a corrective activity therefore if activity1 belongs to class activity and it has a data property of FailureReason set to some certain value then the reasoner infers that this activity is a corrective activity.

Second reasoning method receives raw context data from Behavior Network module and applies predefined SWRL rules on the ontology and transmits the result information instances which need to be displayed with data properties included such as status, level of detail, text color, transparency, etc. to the behavior network.

### *3.4.4. Behavior network module*

Once we determined the output information instance name such as activity1_rule with its relevant properties, a behavior network will be generated using activity set {S} as sensors like Eq. (1), activity set {G} as goals like Eq. (2) and activity instructions as middle nodes {N} like Eq. (3). Each activity holds corresponding properties of InformationInstance class stated in the proposed CAMON ontology.

$$S = \{activity_1, activity_2, activity_3, \ldots\} \quad (1)$$

$$G = \{activity_1\_done, activity_2\_done, \ldots\} \quad (2)$$

$$N = \{node_1, node_2, node_3, \ldots\} \quad (3)$$

Each middle node has a structure like Eq. (4). $c_i$ Shows a list of preconditions for $node_i$ to be activated, $a_i$ is the node add list and $d_i$ holds its delete list.

$$node_i = \{c_i, a_i, d_i, \propto_i\} \quad (4)$$

A specific collection of nodes show different steps of a maintenance procedure. If an activity contains seven steps for completion, seven distinct stages containing middle nodes will be considered in the network for that activity and based on the value of LevelOfDetail property each stage can have one or more parallel middle nodes. Figure 13 illustrates an example of middle nodes in a maintenance task with three steps in a behavior network when the value of LevelOfDetail is inferred as high which needs more detailed instructions.

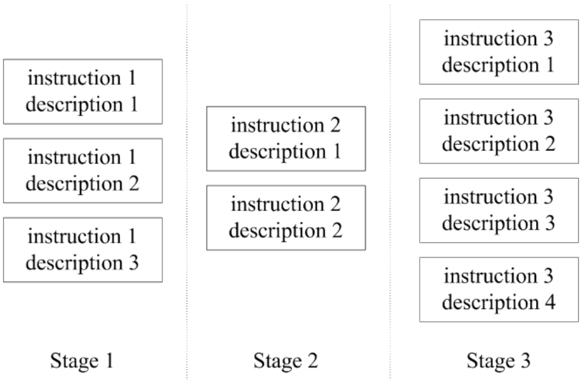


Fig. 13. Middle nodes in a behavior network

After generating middle nodes, successor and predecessor links will be generated for each middle node. There exists a successor link from node x to node y if node y is in the add list of node x and node x is a precondition of node y, i.e. for every proposition $p \in a_x \cap c_y$ there is a successor link from node x to node y. A predecessor link will be created from node x to node y when there is a successor link from node y to x or for every proposition $p \in c_x \cap a_y$. Each node has an activation level $\alpha_i$. When a node activation level surpasses a threshold that node will be activated which causes the node source code to be executed.

This will cause to generate a logical path from each member of sensor set {S} to the relevant member of goal set {G}. The concept is to start the process from sensors and spread an energy through links to activate desired middle nodes in the network en-route selected maintenance task specific steps. Behavior network in this research is implemented using a graph which is surfed using Best First Search (BFS) method up to the time when module arrives to the ongoing activity step.

As shown in Figure 14 considering that Task 1 sensor with high level of detail is the desired sensor and the technician is conducting this procedure at step 2 therefore the relevant goal is Task1_done and the behavior network returns stage 2 middle nodes including instruction2 (description1) and instruction2 (description2) with their predefined properties. First the add list of selected sensor which are stage 1 middle nodes will be activated and they activate middle nodes of stage 2 which are intended nodes.

At last a collection of middle nodes will be sent to the Visualization method. The structure of such output collection is shown in Figure 15. Each member of this collection includes three properties, namely, item data, text color and presentation type. Item data represents the content of the output. Text color and presentation type are display properties.

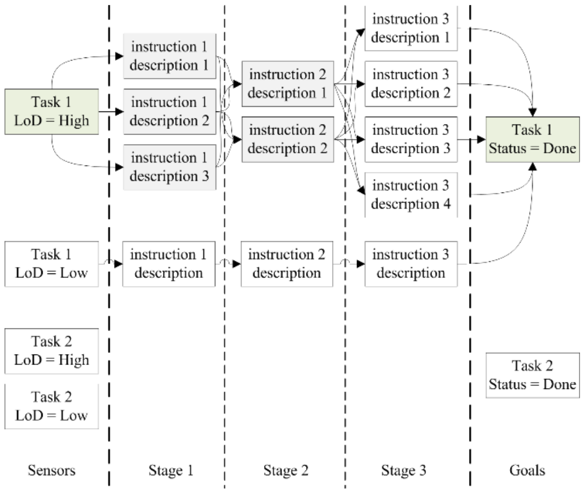


Fig. 14. Process of retrieving instruction steps

### *3.4.5. Visualization module*

Behavior Network module prepare and sends a set of output objects like the one showed in Figure 15 to the System Core module. System Core module receives this set and transmit them to the Visualization module. Input video stream will be obtained using user's device camera in Visualization module and members of the received output information list will be augmented on this video stream. Then the processed video stream including augmented information will be displayed on the user device display screen. Therefore the final user can make use of maintenance scene enriched with virtual added guidance in order to complete the intended instruction.

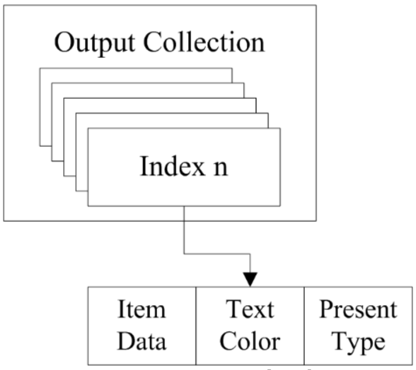

Fig. 15. Transmitted pack

*3.5. Implementation*

This section provides the details about implementation of the proposed architecture in this research. System Core, Context Collector, Behavior Network and Reasoner modules are developed using Java Development Kit (JDK) 1.8. IntelliJ Idea is used as an IDE for developing source codes. Microsoft SQL Server 2005 serves as database server. Behavior Network module connects to SQL Server using jdbc-4.0 external library in order to obtain information instances and create required behavior network. Reasoner module uses owlapi-osgidistribution-4.1.0 external library for the sake of reading ontology classes, properties and individuals. Com.-clarkperisa.protege.plugin.pellet-2.2.0 library is used by Reasoner module for further Ontology and SWRL reasoning. Visualization module is implemented using C++ programming language and uses ARToolkit to implement AR and superimpose virtual objects over environment.

## 4. Experiment

The proposed framework provides assistance to the users to perform maintenance tasks based on AR technology. An experiment has been conducted in a simulated maintenance environment to validate the output results of the proposed framework. In the simulated environment the user tries to change the air filter of a car engine in different contexts. The results of the experiment is as follows.

In the first experiment a novice BDM engineer is changing the air filter of a car engine. The engineer preferred color is set as red and device CPU usage of his device is obtained as 75 percent. As Figure 16 illustrates a red box and a guidance string introducing the location of container of air filter is shown to the user at first step based on user preferred color and symbolic presentation due to high CPU usage.

The engineer continues to the next step of task in the second experiment with a different to previous step that CPU usage is reduced to 30 percent and battery level shows 80 percent, therefore intuitive presentation is selected to help user open the screws and circled arrows will be showed over the screws location to show the user that the screws need to be opened anticlockwise at this step (Figure 17). A caution sign is also presented on the scene as an alert to the novice user of the system.

Fig. 16. Change air filter - step 1

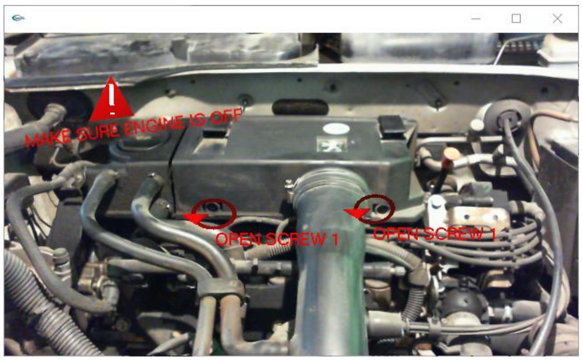

Fig. 17. Change air filter - step 2

In the third experiment, user device battery level dropped off to 25 percent at second step and it caused to change presentation type to symbolic and also the user changed his preferred color to green. So, arrows and caution sign are deleted and simple green circles and a text are presented on the scene as Figure 18.

The forth experiment starts when the user surfs to the third step of task. Considering user preferred color as blue, CPU load as 25 percent and battery connected to AC power, intuitive presentation will be selected as presentation type. Therefore blue 3D arrow with related guidance to open clamps is presented on the scene as Figure 19.

Fig. 18. Change air filter - step 2

Fig. 19. Change air filter - step 3

## 5. Results

A general maintenance environment is being modeled in this research. We implemented a middle-ware to produce sample data to evaluate the proposed framework. Using this middle-ware same sample data is produced for two cases: One for the time when the ontology is used alone and the other for the time when the proposed framework in this paper including a mixture of ontology and behavior network is used.

Sample data in each ontology is divided to two categories: First, when an expert technician is running the system resulting in low level of detail. Second, the time that a technician with low expertise level is using the system which results in high level of detail output information.

For each of these cases the relevant ontology is filled with desired number of individuals as Table 1 in order to evaluate the performance of the proposed framework.

Table 1. Number of individuals

| Class Name | No. of individuals |
|---|---|
| Person | 1 |
| Device | 1 |
| Location | 18 |
| Equipment | 20 |
| Activity | 1, 5, 10, 15, 25, 35, 50, 75, 100, 125, 150 |

A performance evaluation is conducted in order to compare the system run-time using each method. For first time the comparison is made between ontology alone and proposed framework in this paper considering user expertise level as high. Run-time to achieve output information instance and its related properties in this case is shown as Table 2.

Table 2. Run-time comparison when LoD is low

| No. of Activity | Ontology Alone (seconds) | Proposed Framework (seconds) |
|---|---|---|
| 1 | 2.334 | 3.479 |
| 5 | 3.255 | 3.695 |
| 10 | 3.402 | 3.766 |
| 15 | 3.608 | 3.847 |
| 25 | 4.427 | 4.048 |
| 35 | 5.457 | 4.060 |
| 50 | 7.282 | 4.313 |
| 75 | 7.517 | 4.364 |
| 100 | 9.246 | 4.496 |
| 125 | 15.126 | 4.648 |
| 150 | 17.019 | 7.638 |

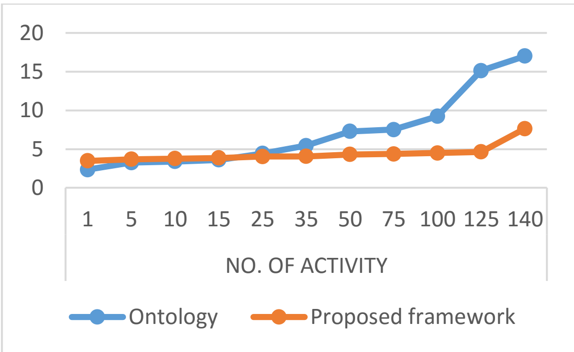

Fig. 20. Run-time comparison when LoD is low

As shown in Table 2 and Figure 20 in the beginning run-time of ontology alone method is better than the proposed framework in this research, but as the number of activities grows the application run-time using ontology increases exponentially, on the other hand the run-time of proposed framework in this paper increases slightly and the average reduction rate of 2.756 seconds in run-time is obtained when using this method.

This evaluation is conducted in another case when the user's expertise level is low which leads the activity level of detail to be high and causes greater number of information instances to be retrieved. Run-time in this case is shown in Table 3.

Table 3 . Run-time comparison when LoD is high

| No. of Activity | Ontology Alone (seconds) | Proposed Framework (seconds) |
|---|---|---|
| 1 | 2.438 | 3.437 |
| 5 | 3.180 | 3.636 |
| 10 | 3.660 | 3.737 |
| 15 | 3.721 | 3.827 |
| 25 | 5.470 | 3.965 |
| 35 | 5.749 | 4.197 |
| 50 | 7.296 | 4.535 |
| 75 | 8.097 | 4.547 |
| 100 | 9.916 | 4.601 |
| 125 | 20.264 | 4.766 |
| 150 | 21.886 | 8.324 |

As shown in Table 3 and Figure 21 the proposed framework achieves better score in both cases. The average reduction rate in run-time considering this case is 3.828 seconds.

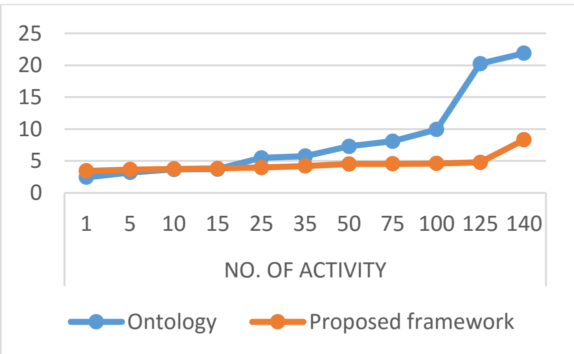

Fig. 21. Run-time comparison when LoD is high

We achieved a total average reduction rate of 42% in the initial run-time when the proposed framework in this paper is used which is more satisfactory to the end users.

## 6. Conclusion and feature work

In this research a review of simple AR-based maintenance systems, context-aware systems and AR-based context-aware maintenance systems has been conducted. A context-aware framework for AR systems in maintenance domain is proposed. The proposed framework employs ontology and behavior network.

Ontology models the maintenance context and defines entities in the real environment and their inter-relations. Applying logical SWRL rules and reasoning based on the current user's context in order to infer and provide context adaptive output to the final user is also the responsibility of ontology and the implemented reasoning engine.

The need for defining SWRL rules to select information collection according to the ongoing activity step is ignored in the ontology and turned over to behavior network instead which selects the collection of output information to be superimposed on the screen based on the current activity step. This will cause to reduce number of SWRL rules defined in the ontology and decrease the time required to run the framework.

There are things remained in this research to be improved as future work. First, However context-awareness helps to improve the usability of AR systems there is lack of research in this field in maintenance contexts and there are different context parameters that can be employed to provide a better adaptive user interface to the users. Second, the lack of an authoring tool is sensed in this research in order to provide a bi-directional interaction tool between technicians and the system to provide maintenance procedure steps by technicians and make corrections on data provided by developers. Third, by the help of context-awareness a suggested list of technicians with the same skill and location is provided, but there is a need for a tool to provide remote collaborative communication between technician and members of this list. Finally, there is a need to evaluate the proposed adaptive user interface in terms of usability, learnability, time required for job completion, etc.

## References


[1] G. P. Sullivan, R. Pugh, A. P. Melendez, and W. Hunt, "O&M Best Practices - A Guide to Achieving Operational Efficiency (Release 2.0)," Richland, WA (US), 2004.

[2] S. K. Ong and J. Zhu, "A novel maintenance system for equipment serviceability improvement," *CIRP Ann. Technol.*, vol. 62, no. 1, pp. 39–42, 2013.

[3] J. Zhu, S. K. Ong, and A. Y. C. Nee, “A context-aware augmented reality system to assist the maintenance operators,” *Int. J. Interact. Des. Manuf.*, vol. 8, no. 4, pp. 293–304, 2014.
[4] B. Schwald and B. DeLaval, “An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context,” in *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2003.
[5] S. Lee and Om. Akin, “Augmented reality-based computational fieldwork support for equipment operations and maintenance,” *Autom. Constr.*, vol. 20, no. 4, pp. 338–352, 2011.
[6] D. W. F. Van Krevelen and R. Poelman, “A Survey of Augmented Reality Technologies, Applications and Limitations,” *Int. J. Virtual Real.*, vol. 9, no. 2, pp. 1–20, 2010.
[7] H. Flatt, N. Koch, C. Röcker, A. Günter, and J. Jasperneite, “A context-aware assistance system for maintenance applications in smart factories based on augmented reality and indoor localization,” in *20th IEEE International Conference on Emerging Technologies and Factory Automation*, 2015, pp. 1–4.
[8] Tr, G. Chen, and D. Kotz, “A Survey of Context-Aware Mobile Computing Research,” 2000.
[9] B. N. Schilit, N. Adams, and R. Want, “Context-aware Computing Applications,” *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*. pp. 85--90, 1994.
[10] J. Wang, Y. Feng, C. Zeng, and S. Li, “An augmented reality based system for remote collaborative maintenance instruction of complex products,” in *2014 IEEE International Conference on Automation Science and Engineering*, 2014, pp. 309–314.
[11] J. Zhu, S. K. Ong, & a Y. C. Nee, and a Y. C. Nee, “A context-aware augmented reality assisted maintenance system,” *Int. J. Comput. Integr. Manuf.*, vol. 28, no. 2, pp. 213–225, 2015.
[12] W. Qin, Y. Shi, and Y. Suo, “Ontology-Based Context-Aware Middleware for Smart Spaces,” *Tsinghua Sci. Technol.*, vol. 12, no. 6, pp. 707–713, 2007.
[13] C. Gouin-Vallerand, B. Abdulrazak, S. Giroux, and A. K. Dey, “A context-aware service provision system for smart environments based on the user interaction modalities,” *J. Ambient Intell. Smart Environ.*, vol. 5, no. 1, pp. 47–64, 2013.
[14] N. Marmasse and C. Schmandt, “Location-Aware Information Delivery with ComMotion,” in *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing*, 2000, pp. 157–171.
[15] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton, “Cyberguide : A Mobile Context-Aware Tour Guide,” *Wirel. Networks*, vol. 3, no. 5, pp. 421–433, 1997.
[16] S. Long, R. Kooper, D. Abowd, and G. Atkeson, “Rapid Prototyping of Mobile Context-Aware Applications : The Cyberguide Case Study,” in *Proceedings of the 2nd Annual International Conference on Mobile Computing and Networking*, 1996, pp. 97–107.
[17] A. Schmidt, K. Asante, and A. Takaluoma, “Advanced Interaction in Context,” in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, 1999, pp. 89–101.
[18] S. Oh and W. Woo, “CAMAR : Context-aware Mobile Augmented Reality in Smart Space,” in *Proc. of IWUVR*, 2009, vol. 9, pp. 48–51.
[19] A. Mileo, D. Merico, and R. Bisiani, “Support for context-aware monitoring in home healthcare,” *J. Ambient Intell. Smart Environ.*, vol. 2, no. 1, pp. 49–66, 2010.
[20] H. E. Byun and K. Cheverst, “Utilizing Context History To Provide Dynamic Adaptations,” *Appl. Artif. Intell.*, vol. 18, no. 6, pp. 533–548, 2004.
[21] A. Shehzad, H. Q. Ngo, K. A. Pham, and S. Lee, “Formal Modeling in Context Aware Systems,” in *Proceedings of the First International Workshop on Modeling and Retrieval of Context*, 2004.
[22] X. Wang, D. Zhang, T. Gu, and H. K. Pung, “Ontology Based Context Modeling and Reasoning using OWL,” in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, pp. 18–22.
[23] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, “A Practical Guide to Building OWL Ontologies Using Protégé 4 and CO-ODE Tools Edition 1.2,” *Univ. Manchester*, 2009.
[24] M. R. Hoque, M. H. Kabir, K. Thapa, and S.-H. Yang, “Ontology-based Context Modeling to Facilitate Reasoning in a Context-Aware System : A Case Study for the Smart Home,” *Int. J. Smart Home*, vol. 9, no. 9, pp. 151–156, 2015.
[25] P. Maes, “How to do the right thing,” *Conn. Sci.*, vol. 1, no. 3, pp. 291–323, 1989.
[26] I. J. Song and S. B. Cho, “Bayesian and behavior networks for context-Adaptive user interface in a ubiquitous home environment,” *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1827–1838, 2013.