

Hybrid recommendations and dynamic authoring for AR knowledge capture and re-use in diagnosis applications

Iñigo Fernández del Amo, John Ahmet Erkoyuncu^{*}, Maryam Farsi, Dedy Ariansyah

Cranfield Manufacturing, School of Aerospace, Transport and Manufacturing, Cranfield University, UK

ARTICLE INFO

Article history:

Received 26 February 2021
Received in revised form 8 December 2021
Accepted 11 December 2021
Available online 18 December 2021

Keywords:

Augmented Reality
Failure diagnosis
Authoring systems
Knowledge capture
Ontology-based reporting

ABSTRACT

In Industry 4.0, integrated data management is an important challenge due to heterogeneity and the lack of structure of numerous existing data sources. A relevant research gap involves human knowledge integration, especially in maintenance operations. Augmented Reality (AR) can bridge this gap, but it requires improved augmented content to enable effective and efficient knowledge capture. This paper proposes dynamic authoring and hybrid recommender methods for accurate AR-based reporting. These methods aim to provide maintainers with augmented data input formats and recommended datasets for enhancing the efficiency and effectiveness of their reporting tasks. The proposed contributions have been validated through experiments and surveys in two failure diagnosis reporting scenarios. Experimental results indicated that the proposed reporting solution can reduce reporting errors by 50% and reporting time by 20% compared to alternative recommender and AR tools. Besides, survey results suggested that testers perceived the proposed reporting solution as more effective and satisfactory for reporting tasks than alternative tools. Thus, proving that the proposed methods can improve the effectiveness and efficiency of diagnosis reporting applications. Finally, this paper proposes future works towards a framework for automatic adaptive authoring in AR knowledge transfer and capture applications for human knowledge integration in the context of Industry 4.0.

© 2021 Published by Elsevier B.V.

1. Introduction

Industry 4.0 is bringing digitalisation of complex equipment to improve availability and sustainment for achieving more intelligent and adaptive industrial operations [1]. Technologies like Artificial Intelligence (AI) and Digital Twins (DT) are enabling real-time and automatic data capture and analysis, monitoring and optimisation for faster and improved decision-making [2]. These require integrated data management that is still challenging due to heterogeneity of data formats (e.g. audio or video) and lack of structure of numerous data sources (e.g. manuals or reports) [3]. In this context, human knowledge capture is fostering research attention due to its potential to create valuable data streams [4]. A relevant area for knowledge capture is maintenance diagnosis. Digital diagnosis systems aim to automate fault-finding tasks through sensor data capture and analysis to identify failures' root causes [2,5]. One example of industrial applications is the development of AI-driven Augmented Reality Diagnostic by IBM which was focused on supporting the technician to diagnose the hardware error [6]. But when these systems fail to determine the cause of an identified failure like in No-Fault-Found (NFF) events, experts are still required to do so [7].

When maintainers conduct diagnosis procedures, they are usually asked to report them along with failure modes, and conditions identified [8]. These reports contain valuable knowledge to be re-used to enhance diagnosis systems [9] and consequent reporting tasks where those modes and conditions reappear [10]. Although capturing such knowledge seems relevant to enhance diagnosis systems, that should be done in a structured manner to enable knowledge to be re-used [11].

Augmented Reality (AR) technologies can embed digital information in human operations in co-existence with real-world objects [12]. They have been widely applied to support complex human tasks (e.g. maintenance repair or medical surgeries) because of their abilities to transfer knowledge [13]. AR has also been applied for knowledge capture and re-use in manual maintenance operations like repair and assembly [14]. However, there is very little academic evidence of AR knowledge capture and re-use applications in abstract operations like diagnosis reporting. A possible reason relates to the complexities associated with representing abstract knowledge and data input forms through augmented content, which can result in incorrect reports [15]. Another reason involves diagnosis reporting tasks that require the selection of items from extensive datasets like fault conditions [11]. These can overload the augmented scene and the user's field of view if large selection lists are augmented without prior filtering or recommendation of relevant items [16].

^{*} Corresponding author.

E-mail address: j.a.erkoyuncu@cranfield.ac.uk (J.A. Erkoyuncu).

In order to enhance knowledge capture and re-use in diagnosis reporting tasks, this research proposes hybrid recommendations and dynamic authoring for AR maintenance applications. This proposal includes the following contributions:

1. A method to provide context-aware and ontology-based AR recommendations to reduce extensive selection lists in reporting applications through knowledge re-use for decreasing reporting time.
2. A method to dynamically create and allocate content in augmented scenes to avoid augmented content overload and ease data input tasks for reducing reporting time and errors.

This research aims to demonstrate that AR knowledge capture and re-use applications can achieve sufficient reporting effectiveness and efficiency to enable structured knowledge capture. Thus, allowing to re-use it in consequent diagnosis reporting activities. Therefore, this research validates the proposed contributions against conventional alternative reporting solutions for diagnosis reporting tasks. Inspired by similar research works [4,17–19] and well-established frameworks in design research [20], this research employed the following methodological steps including (1) Objectives identification based on literature and industrial user studies as presented by Booth, Papaioannou and Sutton (2012), (2) Solution design: “create a solution to satisfy the research opportunities”, (3) Solution demonstration: “prove the solution’s use to satisfy the research opportunities”, (4) Research validation: “measure the solution’s impact on research opportunities”. Efficiency experiments evaluated to impact of this research’s proposals on two reporting tasks through comparison against alternative reporting tools.

The rest of this paper’s structure is as follows. Section 2 presents a literature review on diagnosis applications and recommendation techniques in AR research to detect current research gaps. Section 3 describes the research’s contributions. Section 4 presents this research’s validation protocol along with experimental methods and cases of study, along with the validation’s results are analysed and discussed. Finally, Section 5 presents this research’s conclusions and advances future works to enable human knowledge integration in extended Industry 4.0 contexts.

2. Literature review

2.1. Diagnosis applications in augmented reality

Diagnosis can be described as the series of human activities that determine the causes of asset’s failures or abnormal behaviours [21]. Such activities include different tasks such as manual’s consultation, fault-finding investigation and reporting [22]. Diagnosis of a complex asset is a complicated task that involves an assessment of various types of information and accurate judgement, which is knowledge intensive, and therefore, requires a knowledge support system to be effective [23,24]. AR technologies have the ability to contextualise virtual information and so can improve the efficiency and effectiveness of diagnosis tasks [25]. For example, Longo, Nicoletti and Padovano [4] described an AR-Digital Twin integration model to augment sensor data and service records for enhancing fault-finding tasks in shop-floors. A similar approach was utilised by Avelle et al. [26] to support fault detection on industrial robots. Ghimire, Pattipati and Luh [27] and Khalil et al. [28] use integrative models of equipment to understand relations between sensor data from different components and visualise remaining useful life indicators (e.g. fatigue). Others like Priya and Vasudevan [29], Wang et al. [30] or Das, Dong and Scherer [31] made use of extended

AR tracking capabilities to detect missing or mis-shaped components for enhancing visual inspection tasks. These studies took advantage of AR technological capabilities (e.g. tracking) to develop advanced data-driven prognostics for enhancing real-time sensor-data visualisation that supports diagnosis tasks.

Besides data-driven diagnostics, AR research have also focused on knowledge-based methods. Due to its knowledge transfer abilities, academics identified AR as a suitable technology to provide remote expert support for diagnosis tasks. Authors like Masoni et al. [32], Rambach et al. [33] or Zenati-Henda et al. [34] focused their efforts on user interfaces that allowed remote experts to seamlessly integrate indications for on-site technicians to conduct fault-finding tasks. Others like Oyekan et al. [35], Hadar et al. [36] and Mourtzis, Vlachou and Zogopoulos [37] provided intelligent fault-tree algorithms to support early diagnosis stages and reduce remote expert’s workload. These studies aimed to ease the knowledge transfer from remote experts to on-site technicians, although few papers analysed the effects of models utilised to send such knowledge through AR. One of them, proposed by the authors [38], described a message structure and associated automatic content-creation rules to enhance knowledge sharing in remote diagnosis scenarios.

Overall, AR research in diagnosis applications has focused on improving sensor-data visualisation and remote expert knowledge transfer for several diagnosis tasks such as fault-finding or visual inspection. However, there is little academic evidence on AR applications to support diagnosis reporting tasks. Having a reliable fault reporting system is a paramount to ensure high quality reusable information [39], minimising NFF incidents [40,41], and human mistakes in safety-related system [42]. Although, few papers made use of AR abilities for knowledge capture [4,34,38] or demonstrated its potential benefits in terms of knowledge re-use [43].

2.2. Recommendation techniques in augmented reality

Knowledge re-use can be described as the series of processes to capture, package and share knowledge [44]. Hence, knowledge can be seen as a set of items (e.g. documents or ontology’s instances) to be classified, evaluated and suggested for a given purpose in a given context. That is the purpose of knowledge-based recommendation systems [45]. Recommender techniques have been widely applied in AR research to filter and contextualise augmented content for reducing augmented scene’s overload and improving task’s support [46]. For example, Schaeffer et al. [46] proposed a content-based recommender algorithm to inform consumers on product’s ecological impact at the point of sale. Collaborative-filtering (user-based) recommenders have also been used in AR, Lin and Chen [47] described a deep-learning algorithm to advise learnings tasks and methods through AR according to their learning stage. The same applies to knowledge-based recommenders, Torres-Ruiz et al. [48] described a pattern-matching algorithm to suggest museum itineraries. AR studies have evaluated all kinds of recommendation techniques (e.g. context-aware, content-based, ontology-based, etc.), although most of them were applied for commercial and educational purposes. While there exists evidence of recommendation techniques for knowledge reuse in diagnosis applications [21,22,49–51], the authors could not find any that evaluated the potential benefits of AR technologies like context-awareness for enhancing knowledge re-use.

2.3. Research gaps

Latest research reviewed on AR for diagnosis applications makes use of its knowledge delivery abilities to support

human-related tasks like fault-finding or visual inspection with enhanced sensor-data visualisation. Although the authors found some academic literature on AR diagnosis applications for supporting knowledge capture tasks like reporting [43], it presented little evidence of the effects and benefits of knowledge re-use. However, AR knowledge re-use abilities have been widely applied in other applications like education or tourism in the form of recommendation methods [46]. Various recommender techniques (e.g. knowledge-based, collaborative filtering or context-aware) have been used mainly either to reduce user's field-of-view overload or to improve AR support of human-related tasks. Moreover, research on diagnosis have also analysed the benefits of knowledge re-use recommendation methods, although the authors could not find any that evaluated AR benefits like context-awareness.

Based on the discussions above, the authors identified several research gaps on AR research for diagnosis applications. First, there is lack of evidence on AR diagnosis applications for reporting tasks. Second, there is also little evidence of AR research that evaluates the effect of knowledge re-use in diagnosis applications. If both gaps were to be fulfilled, such research would be able to evaluate the effects of AR technologies for diagnosis reporting applications and also the benefits of context-aware knowledge re-use techniques. This will help to further improve the efficiency of diagnosis tasks and to ease the integration of expert knowledge as a data source for diagnosis systems or Digital Twins in the context of Industry 4.0. In order to contribute to the fulfilment of both research gaps, this paper proposes (1) an AR-based hybrid recommender method for improving diagnosis reporting tasks through knowledge re-use and (2) a method for dynamic reporting-content creation. The former takes advantages of both, knowledge-based and context-aware recommendations, through AR authoring and tracking techniques. This research aims to validate the proposal's ability to enhance diagnosis knowledge capture and re-use while sustaining the efficiency of fault-finding tasks. Thus, improving the integration and applicability of expert knowledge in the context of Industry 4.0.

3. Proposed methodology for hybrid recommendations and dynamic authoring methods

In [52], the authors proposed a framework for automatic authoring in AR knowledge transfer applications (PMAU). It included (1) a method to declare programmable content formats and (2) a real-time, ontology-based, pattern-matching algorithm to pair formats and maintenance datasets.

This proposal (Fig. 1-a) had several limitations for knowledge capture and re-use applications. First, it proposed static content allocation that resulted on occlusion and overload issues for assets bigger than 1.5 m in width. Second, it proposed adaptive content formats for knowledge transfer applications but did not consider formats for knowledge capture ones that ensure correctness of data inputs. Third, it did enable to adapt maintenance information to different content formats but did not consider adapting information itself to the maintenance context. For solving the issues above and fulfil the research gaps identified in Section 2, this paper proposes hybrid recommendation and dynamic authoring techniques (RPMAU – Fig. 1-b) to extend the previous proposal for knowledge capture and re-use applications. This research's contributions consist of the following:

1. A method to provide context-aware and ontology-based AR recommendations to improve efficiency of diagnosis reporting tasks through knowledge re-use.
2. A method to dynamically create and allocate content in augmented scenes to avoid their overload and improve efficiency and effectiveness of reporting tasks.

Fig. 2 presents an overview of the extended authoring method (RPMAU) including the contributions proposed in this research. Similarly to PMAU, RPMAU starts when the user 'selects an ontological class'¹ that represents the information that the user wishes to report. Then, it analyses user-configured environment and user features to discard content formats that are not suitable for the augmented scene according to real-world conditions like lack of light, noise, etc. After that, RPMAU also conducts semantic analysis of content formats' data facets and ontological asserted properties to match which content formats (e.g. models or text) are suitable to augment properties' information. It is at this point where RPMAU starts differing from PMAU. RPMAU proposes new content formats that include checking features to ensure correctness of data being reported. Matching of content formats and ontological properties is referred in Fig. 2 as 'assign fabrications' because fabrications are defined as pieces of augmented content that implement formats visualisation and interaction modes to augment a set of individual properties. Once fabrications are assigned, RPMAU also differs from PMAU as its 'applies recommendation facets'. At this point, RPMAU uses the proposed recommender technique to filter augmented information prior to augmentation. Recommendation rules are declared as facets so they can be implemented as an extension for programmable content formats.

Once recommendations are applied, fabrications are ready for augmentation in the scene in the form of elements. Elements are sets of fabrications that comprise all asserted properties of the ontological class being reported. Again, RPMAU differs from PMAU as so does the method to 'allocate a spot' (Fig. 2) for the element in the augmented scene. In RPMAU, the asset being maintained is analysed in terms of its size to determine what type of allocation is best to avoid content overload, either around the user or around the asset. Once the content is augmented in the scene, the user can interact with it for reporting purposes. When the user is finished, RPMAU, differently to PMAU, 'submits the resultant ontology individual' to the knowledge base and keeps 'trace' of it for analysing the maintenance context in further recommendations. This dynamic authoring combined with semantic-based filter facilitate augmentation of different amount of information to ensure the relevant augmented contents is scalable in a wide range of scenarios.

Fig. 2 details the overview presented in Fig. 1-b in more detail. It exemplifies how the RPMAU algorithm uses a class (`cl(prm, dcp, dop)` – (1)), formats with recommendation facets (`ft(uf, ef, dfn, rfl)` – (4)) and augmentation configuration (`ac(ace(seni), acu(dmj, dsk))` to automatically generate augmented content that provides context-aware, ontology-based recommendations of individuals (`in(prm, dcp, dop)`). The Figure displays classes (`cl`), individuals (`in`), formats (`ft`) and augmentation configuration (`ac`) as JSON objects to show their data. The figure also presents the different steps of the RPMAU algorithm, including their use of the abovementioned data along with the steps inherited from PMAU. The resultant exemplary element and its fabrications are displayed at the bottom of the figure. Each box containing data (`cl`, `in`, `ft`, `ac`), RPMAU steps and user selections have a different shape and colour as described in the legend.

The previous description of RPMAU in comparison with PMAU includes the contributions proposed by this research, which are explained in-depth in following subsections. First, Section 3.1 presents the dynamic authoring method as an extension for knowledge capture applications of the previous automatic authoring method for knowledge transfer applications [52]. This includes RPMAU's content formats and the context-aware content

¹ Words between quotation marks in this paragraph refer to Fig. 2.

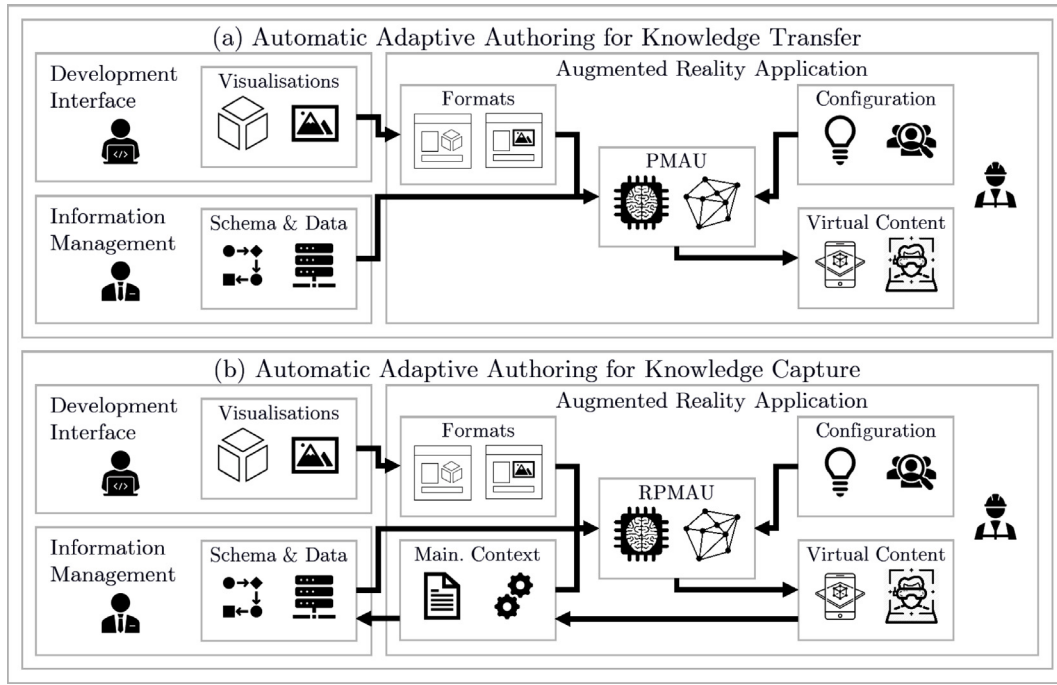


Fig. 1. Overview of automatic authoring for knowledge (a) transfer and (b) capture applications – It compares software modules (development interface, information management and AR application) in authoring approaches to identify new components needed for automatic authoring in knowledge capture.

allocation method and. Then, Section 3.2 describes the context-aware, ontology-based AR recommender technique that extends the dynamic authoring proposal to enable knowledge re-use, which comprises RPMAU's recommendation facets and their application. Finally, Subsection 3.3 presents these contributions implemented to diagnosis reporting tasks.

3.1. Dynamic authoring for augmented reality knowledge capture applications

The dynamic authoring proposal for AR knowledge capture applications can be considered an extension of a previous automatic authoring method for knowledge transfer applications presented in [52]. This extension can be considered a relevant contribution since it proposes two new algorithmic modifications to the previous method for enabling AR knowledge capture applications. These modifications are (1) content formats for knowledge capture with data correctness checking features and (2) the context-aware element allocation procedure. These are explained in the following subsections.

3.1.1. Content formats for structured and correct knowledge capture

PMAU's pattern-matching algorithm uses data, user and environment facets to assign and discard suitable content formats for augmenting a given ontology's individual for knowledge transfer. PMAU's proposal assumes that ontology individuals have properties (attributes and relationships) asserted and so, these can be assigned to formats for augmentation. Nevertheless, AR-based individual instantiation (knowledge capture) would require the AR user to make those assertions. In order for RPMAU to augment elements for individual instantiation, it would require some modifications to PMAU. These are the following:

1. It would require allowing the user to select for both individual visualisation and instantiation procedures.
2. It would require determining ontology classes' "exemplary" individuals to accept possible data facets' rules of value.

3. It would require submitting instantiated individuals to the ontology server and so, ensuring instantiation correctness.
4. It would require programmable content formats capable of capturing data correctly and efficiency through user interaction.

RPMAU's proposal extends PMAU with the modifications above to enable automatic creation of augmented content for knowledge capture applications. The first requirement is solved by embedding a procedure selection option as part of AR application's initialisation behaviour. Thus, users can choose whether use the AR application to visualise or instantiate ontologies' individuals, which will affect further AR application's behaviour (e.g. formats colour coding). The second requirement has been solved by automatically generating "ideal" individuals for each ontological class. A knowledge base call has been programmed so that it identifies an instantiation of each property (attribute or relationship) asserted to the class the user aims to instantiate. The subsequent http request retrieves the class example(*cl*) including asserted properties to the class (*pr*) and inferred "component" and "operation" distances (d_{cp} , d_{op}) utilised for content allocation, just as an individual in PMAU.

$$cl = (pr_1, pr_2, \dots, pr_m, d_{cp}, d_{op}) \quad (1)$$

$$pr_{in} = (res_{in,m,na}, res_{in,m,ra}, res_{in,m,ty}, res_{in,m,va}), \quad (2)$$

where m is the number of asserted properties to individual and $res_{in,m} \in URI \cup S \cup \mathbb{R} = uri, string \text{ or number that a property instantiates, and}$

$$d_{cl,c} : C_{in} \times C_c \rightarrow [0, \infty) \quad (3)$$

where, $c \in \{\text{component, operation}\} = \text{class to which individual's distance is measured to}$

If a property assertion is not found in the server's knowledge base, then the resultant json object will include no value for that property. Therefore, any "reporting" content format that has a

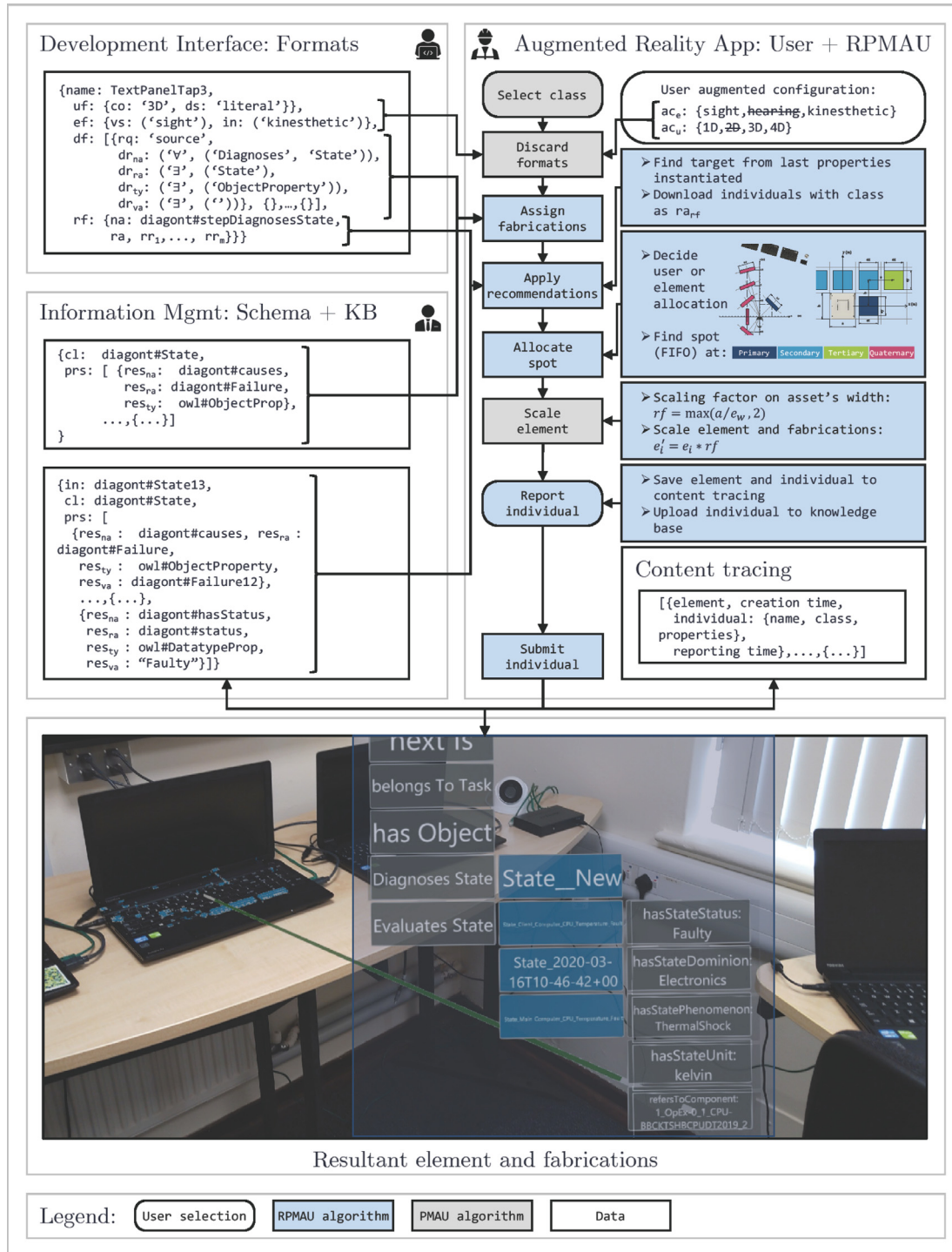


Fig. 2. Dynamic authoring algorithm including hybrid recommendations to enable knowledge capture and re-use.

non-null data value rule will not be pairable to that property. This behaviour allows RPMAU algorithm to treat content formats for knowledge transfer and capture similarly. Thus, identifying what file formats are required to record in case a content format allows to do so (e.g. pictures).

The third requirement involves additional server calls to ensure submitted individuals are correct instances of ontology classes including correct assertion in terms of domain and range of attributes and relationships instantiated. The reported individual is sent to the server in the form of a json object. A

http post request makes the necessary checks to ensure that the individual only includes properties whose domains are the instantiated class. It ensures that properties' ranges are those declared on the server's ontology schema. If a property fails to comply with these, the individual will not be added to the knowledge base. The final requirement involves creating programmable content formats whose user interaction gives as result data inputs. The research classified by visualisation modes and their programmed behaviours and facets can be seen at 10.17862/cranfield.rd.12382604. These formats aim to reduce reporting

Table 1
Description of content formats classified by visualisation and interaction.

Format	Visualise	Interact	Behaviour
Text dictation	Text	Dictate	Utilises MRTK text-to-speech module to capture strings
Text automatic	Text	None	Automatically records data value if property's range is of datetime type
Text keyboard	Text	Tap	Utilises HoloLens 2 default keyboard to input numerical values
Individual selection	Text	Tap	Utilises user's tap gesture to select individual from knowledge base. If new individual selected, then activates new individual instantiation.
Datatype selection	Text	Tap	Utilises user's tap to select a value from a datatype set (e.g. boolean).
Model selection	Model	Manipulate	Utilises user's manipulation to select an individual based on its referred model (obj file).
Recommend selection	Model + Text	Tap	Utilises individuals' properties to filter them according to recommendation algorithm and allow user to select from filtered selection.

time by easing input visualisation and filtering input information. They use a colour code to show interactive states: blue for selectable, and green for selected.

Different to PMAU's content formats, RPAU's content formats require extra features to ensure data inputted through user's interaction is correct according to formats data facets. For programming content formats in RPAU, developers need to implement data checking rules that comply with assigned properties instantiation rules for their ranges and types. For validation purposes, this research implements content formats that comply with expected properties types and ranges: individuals, numerical values, textual values, dataset values and components. There are other datatypes like images or audio files which could also be considered for creating new content formats. Future works can implement these to extend heterogenous datatypes that can be captured through RPAU.

Table 1 describes RPAU's content formats and their behaviours implemented in this paper for validation purposes. For example, 'Text keyboard' is meant for inputting numerical values. So, when the user introduces a non-numerical character, the format will return an error asking the user to input numerical values only. 'Individual selection' allows to select from all individuals of a certain class by name as well as to instantiate new individuals for continue with the reporting procedure. 'Model selection' allows to select an ontology individual based on model files asserted to it as a datatype property. Its aim is to reduce input time by using a different visualisation mode (model) than just referring to the individual by its name (text). 'Recommendation selection' uses hybrid recommendation methods to pre-filter selectable individuals. Thus, decreasing input time by reducing user's selection options. Moreover, this format also implements diverse visualisation modes (e.g. models) to simplify the selection process. The recommendation framework for improving AR knowledge re-use applications utilised in this content format is presented in Section 3.2.

3.1.2. Context-aware element allocation procedure

PMAU's content allocation assumed that augmented content needs to be overlaid around the equipment. So, maintainers have non-occlusive information visuals from any perspective while operating on the equipment. Nevertheless, this assumption has limitations when equipment's size is bigger than 1.5 m according to authors' testing. In this case, maintainers are required to highly deviate from their focus for consulting augmented information. This can become even more problematic when users are required to move around the asset for reporting purposes. To alleviate this issue, RPAU proposes to analyse equipment size and dynamically adapt content allocation accordingly. When the equipment is too big (>1.5 m), it maintains augmented content at specific distances from the user. Otherwise, it enables for augmented content to still be allocated around the equipment. Thus, enabling users to have good visuals of augmented content at any given situation.

The proposed dynamic content allocation method firstly analyses equipment's size measures using equipment's CAD model boundaries. If any of these are bigger than 1.5 m, then it enables the user-based allocation algorithm. Otherwise, it activates the asset-based allocation algorithm. Secondly, this method allocates newly created elements according to their assigned location (primary, secondary, tertiary and quaternary) calculated from ontological distances similarly to PMAU. In asset-based allocation, elements are placed at the first-free spot in its location (primary (1), secondary (4) or tertiary (6)). If there is not any free spot, then the allocation algorithm eliminates the earliest element allocated to that location and replaces it with the new one. In user-based allocation, newest element created is assigned to the primary location (1 spot). Any previous element in such location is moved to the quaternary (6 spots) location and placed in the first free spot available. If there is not any free spot, then the element replaces the earliest one moved to the quaternary location. Overall, both algorithms use a FIFO approach (First-In-First-Out) to allocate elements to spots at any location. Because elements are placed at different distances from the user, these need to be re-scaled in some cases. The re-scaling process occurs considering element's width as re-scaling factor. Other measures (length and height) are re-scaled accordingly. Element's width (e_w) is initially set at 0.5 m, which is sufficient for non-occlusive visualisation in user-allocation cases. Therefore, elements' re-scaling only occurs on asset-based allocation cases and it is dependent on equipment's width. If asset's width is bigger than 0.5 m, then element is re-scaled according to the maximum between 2 and the rate between asset's and element's width ($rf = \max(a/e_w, 2)$). Besides, elements are embedded additional behaviour for always pointing in user's direction. Thus, allowing for easier visualisation and avoiding occlusive scenarios.

For this method to work, AR requires asset tracking to be enabled. This research's implementation uses Vuforia's model and image target registration and tracking features to do so. Similarly to other methods in this proposal, it uses ontology-related files to load in real-time model or image target tracking and related CAD files ("obj") to allow AR users to select the asset or equipment to be maintained. This also enables other asset-based features of this proposal like component spatial proximity filtering for content-based recommendations.

3.2. Hybrid recommendations for augmented reality knowledge re-use applications

An important limitation in AR-based knowledge transfer and capture relates to the size that different visualisation formats take to augment maintenance information. AR user's field of view is limited and so is the space available to augment content. In AR knowledge transfer applications, this issue can be solved by accurate information management. For example, if there is a maintenance task that comprises various steps, this issue can be

solved by linking steps in order rather than connecting all to the main task. Nevertheless, it is necessary to use more advanced information filtering methods to solve this in AR knowledge capture applications. For example, if the AR application required to report the step that was done last, then it would not be necessary to augment those steps that were conducted previously. Besides, such filtering techniques can help to further improve efficiency of AR knowledge capture applications. For example, if the AR application was able to determine that the report was describing a repair task, then it could further filter “selectable” steps discarding those that are not repair-related. Overall, context-aware recommendations seem like reasonable methods to improve content adaptiveness in AR knowledge transfer and capture applications.

For realising automatic adaptive authoring in AR knowledge re-use applications, this paper proposes a recommendation framework to enable content formats to filter the information they augment. This framework includes recommendation algorithms as part of content formats in a generic manner. So, AR developers can create formats which can filter the information they augment. The framework aims to describe hybrid content-based recommendation methods: context-aware and ontology-based. So, content formats can apply them automatically without re-programming coded visualisation and interaction behaviours. Recommendation facets (*rf*) are the features declared by content formats that enable them to filter augmented information (selectable ontology individuals). These facets declare the instantiable property ($na_{pr.ft}$, $ra_{pr.ft}$) subject to recommendation and the rules (*rr*) to assess similarity between target and cases. The instantiable property subject to recommendation is declared by its name ($na_{pr.ft}$) and range ($ra_{pr.ft}$) for enabling the algorithm to identify recommendation targets. This instantiable property must be present in a source data facet of the format that declares the recommendation facet.

$$rf_{ft,n} = (na_{pr.ft}, ra_{pr.ft}, rr_{ft,n,1}, \dots, rr_{ft,n,l}) \quad (4)$$

t = property attributes of target case

l = number of rules that comprises the recommendation algorithm of a given format

Recommendation rules (*rr*) comprise the methods implemented in the recommendation facet to assess similarity between individual target and potential cases. These are defined by similarity method's type (*rt*) implemented, name ($na_{pr.rr}$) and range ($ra_{pr.rr}$) of property evaluated and the semantical weighted features (x_i, s_i) the implemented method utilises. Property name and range declare the property being assessed, which can be different to the property (attribute or relationship) being recommended. This research proposes four different similarity functions: binary, symmetric, discrete and spatial.

$$rr_{rf,l} = (na_{pr.rr}, ra_{pr.rr}, rt_{rr,l}, rs_{rr,l}) \quad (5)$$

$$rt_{rr,l} \in \{\text{binary}, \text{symmetric}, \text{discrete}, \text{spatial}\}$$

$$rs_{rr,l} = ((x_1, s_1), \dots, (x_m, s_m))$$

$$x_i \in \mathbb{R} = \text{realnumbers}; s_i \in S = \text{strings}$$

Binary similarity assessment ($rt = \text{binary}$) implements Kronecker's delta function to compare target and case. It applies to any datatype and object property assessed.

$$sim_{rt_{rr,i}=\text{binary}}(t, c) = \begin{cases} 1, & va_{pr_c} = va_{pr_t} \\ 0, & va_{pr_c} \neq va_{pr_t} \end{cases} \quad (6)$$

$$sim_{rt_{rr,i}=\text{binary}}(t, c) \in [0, 1]$$

Symmetric similarity assessment ($rt = \text{symmetric}$) implements a triangular function [53] to numerical-range properties (e.g. `xsd:double`).

$$sim_{rt_{rr,i}=\text{symmetric}}(t, c) = 1 - \frac{|va_{pr_t} - va_{pr_c}|}{\max(va_{pr_t}, va_{pr_c})} \quad (7)$$

$$sim_{rt_{rr,i}=\text{symmetric}}(t, c) \in [0, 1]$$

Discrete similarity assessment ($rt = \text{discrete}$) implements an analogous function to symmetric assessment but applicable to string-range and object type properties for semantic evaluation. To do so, it uses weighted semantical features declared by recommendation rules.

$$sim_{rt_{rr,i}=\text{discrete}}(t, c) = \begin{cases} x_i, & s_i \supset va_{pr_c} = va_{pr_t} \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

$$\sum_m x_i = 1$$

$$sim_{rt_{rr,i}=\text{discrete}}(t, c) \in [0, 1]$$

Spatial similarity assessment ($rt = \text{spatial}$) implements an analogous function to symmetric assessment that aims to evaluate the distance between real-world objects in the augmented space. To do so, it uses context-aware features from AR tracking and registration techniques to measure distances between objects.

$$sim_{rt_{rr,i}=\text{spatial}}(t, c) = \begin{cases} 1 - \frac{|\vec{tc}|}{|\vec{av}|}, & va_{pr_c} \in \text{Asset} \wedge va_{pr_t} \in \text{Asset} \\ 0, & \text{otherwise} \end{cases} \quad (3-9)$$

$$sim_{rt_{rr,i}=\text{spatial}}(t, c) \in [0, 1]$$

Fig. 3 presents the formulae declared to implement the spatial similarity assessment function. When the asset to maintain is selected through RPMAU, AR registration can identify its size, position and rotation in real-life. Besides, RPMAU does AR tracking by overlaying the asset's CAD model and scaling to match the registered real-world counterpart. Because the asset's CAD model includes its components, then these can be traced through their names in recommendation cases to measure the distances between them according to the formulae in Fig. 3. Fig. 3 also presents an example of a distance calculation between two components of this research's case study.

Fig. 3 shows how to calculate the distance between target and case components (tc) and the maximum distance (av) considered as the asset volume. Recommendation rules are assessed through a top-k nearest neighbour strategy ($similarity(t, c)$) by adding similarities calculated from each rule. Based on fabrication's size, number of similar neighbours was set to three ($k = 3$) to avoid augmented content overload in user-allocated scenes. Nevertheless, future research works should study the optimal number of similar neighbours according to recommendations' accuracy.

$$similarity(t, c) = \frac{\sum_{i=1..l} w_i * sim_{rt_{rr,i}}(t, c)}{\sum_{i=1..l} w_i}; w_i = 1/l \quad (10)$$

Recommendation facets aim to filter object type properties being augmented for instantiation ($na_{pr.rf}$, $ra_{pr.ft}$) and recommendation rules apply to target (t) and cases (c) individuals. Cases are therefore all individuals in knowledge bases whose asserted class coincides with the range ($ra_{pr.ft}$) of the recommendation facet. Target is the last individual instantiated within the AR knowledge capture application whose asserted class also coincides with the range ($ra_{pr.ft}$) of the recommendation facet. Recommendation facets download cases from knowledge bases and use the

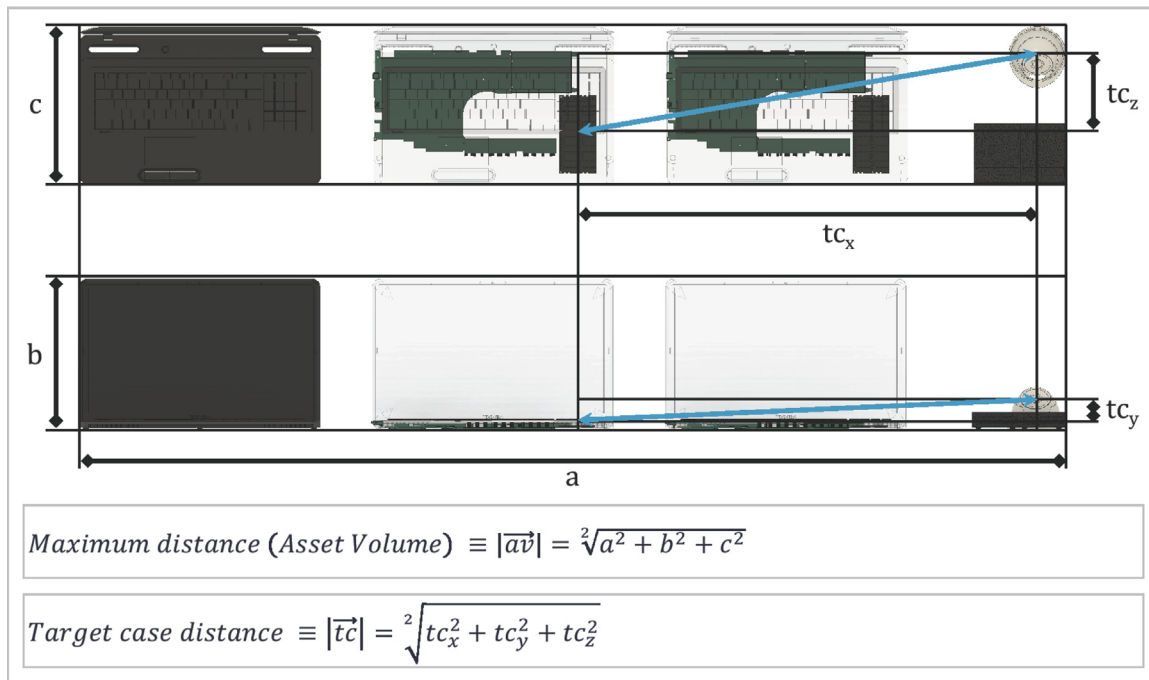


Fig. 3. Formulae for spatial similarity assessment and examples — An example is presented to describe formulae used in the spatial similarity assessment function (4–9).

proposed algorithm's content tracing feature to identify targets. In RPMAU, recommendation facets are applied once fabrications have been assigned (Fig. 2). So, they can identify if the properties assigned to their formats match their name and range.

Recommendation similarity strategy and methods aim to evaluate the impact of content-based recommendation techniques in AR knowledge re-use applications for reducing content overload and improving knowledge-capture efficiency. Moreover, semantic and spatial similarity assessment methods have been proposed to take full advantage of ontology-based AR context-awareness features. Nevertheless, this framework proposes a method to generically declare recommendation rules, so other approaches can be implemented to take advantage of other AR proposed features (e.g. content tracing, user performance, etc.).

4. Application, results and discussion

4.1. System implementation

The proposed solutions were implemented within a prototype system for experimentation. This prototype consists of two subsystems: (1) a cloud server for maintenance ontologies storage and (2) a HoloLens 2 AR application. Fig. 4 presents the languages and platforms utilised to code each subsystem. The cloud server storage uses the graphical database Neo4j [54] to store maintenance ontologies and Cypher [55] and neosemantics [56] to support data transfer through OWL and RDFS. Besides, the server incorporates a web-based application coded in EJS [57] for maintenance experts to input maintenance data, which has already been described in [52]. And also a service provider built in NodeJS [58] to transfer ontology data (e.g. classes, individuals, etc.) and related files (e.g. obj, png, etc.) using HTTP requests and JSON objects to the AR application. The HoloLens-based AR application has been coded and deployed using Unity Game Engine [59] and Visual Studio [60]. Content formats and recommendation facets and RPMAU's algorithm have been coded in C# [61]. HoloLens interaction has been enabled with Mixed Reality Toolkit [62]. Besides, Vuforia [63] has been used to enable

registration and tracking capabilities in the AR application and has been coded to use the same JSON-based API to transfer from the cloud server necessary ontology-related files like Vuforia's model targets.

4.2. Experiment design

This paper proposes recommendable content formats and dynamic authoring in for knowledge capture and re-use in AR maintenance applications. These aim to improve efficiency of maintenance reporting applications by (1) enhancing data input through instantiable augmented content and (2) data selection through hybrid (context-aware and ontology-based) recommender facets. These have been implemented for a case of study in diagnosis reporting. Hence, this research's validation aims to evaluate the impact of dynamically-authored, recommendable content in the efficiency of diagnosis reporting operations.

Since reporting operations are mostly human tasks, validation of human-computer interaction technologies should focus on their impact on human performance. However, the evaluation presented in this study was to primarily assess the impact of the proposed method on human productivity measure regarding efficiency (**time**), and effectiveness (**errors**) [64] rather than the comprehensive evaluation of human-system ergonomics. Besides, this paper proposes recommendations that can affect human performance in reporting operations and so, their **accuracy** should also be analysed [30,65,66]. This research aims to analyse the validity of this research's contributions:

1. Recommendation facets (Section 3.2) can improve descriptiveness (**recommendations accuracy**) of fault conditions in diagnosis scenarios for improving reporting efficiency (**time**).
2. Dynamic authoring and associated content formats (Section 3.1) can improve human performance (**time**, and **errors**) in diagnosis reporting operations.

Inspired by similar research [4,17,19,30], this research's validation proposes experimental and survey methods to evaluate the

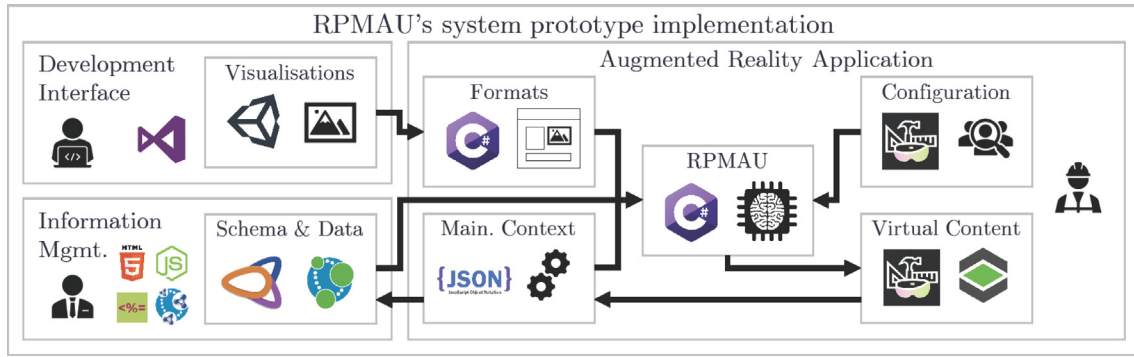


Fig. 4. Description of the proposal's implementation as a software system prototype — It replicates the structure of Fig. 1-b to present the logos of each tool/language utilised to develop the system's prototype.

Table 2
Overview of validation methods, criteria and objectives.

Method	Quantitative	Qualitative	Objective
Experiments	Accuracy		Evaluate the proposal's ability to produce recommendations for identifying faulty conditions in diagnosis scenarios
	Time		Evaluate the proposal's ability to reduce time for improving efficiency of diagnosis reporting operations
	Errors		Evaluate the proposal's ability to reduce errors for improving effectiveness of diagnosis reporting operations

abovementioned research contributions. Table 2 presents these methods, the criteria they aim to analyse and the objectives for doing so.

For these methods and criteria to appropriately evaluate diagnosis reporting effects, the following assumptions must hold true:

- Implemented recommendation facets select component conditions that can be considered faults of the contextual failure. So, accurate recommendations and their selection can affect diagnosis reporting performance through their ability to simplify reporting tasks.
- Reporting operations consist of data input steps. So, errors or incorrectly inputted values can be considered a measure of reporting effectiveness. If the above is true, then time and errors results should not be correlated.
- Reporting operations are human tasks. If errors can be considered a measure of reporting effectiveness, reporting time can be considered a measure of reporting efficiency.
- Users complete the task. 'completeness' measure is inherent and fulfilled by completion of all tasks and steps by the users over the experiment.

Validation experiments require additional reporting tools to which compare this research's proposal. These tools or **solutions** should have different attributes regarding this research's contributions. These attributes refer to recommendations (Section 3.2) and dynamic augmented content (Section 3.1). As part of this validation, the authors developed the following tools for comparison:

- AR + Recommendations (ARR): this paper's proposal includes both, augmented content and recommendations, for knowledge capture applications (Section 3).
- AR + Non-Recommendations (ARN): an alternative AR solution that includes augmented content for knowledge capture applications but not recommendations [52].
- Non-AR + Recommendations (TBR): an alternative non-AR solution that only includes recommendations but not augmented content [52]. This solution did not filter the recommendation facets of the context of augmented contents.
- Non-AR + Non-Recommendations (TBN): an alternative non-AR solution that neither includes augmented content nor

recommendations [52]. This was used as baseline for comparative analysis against the other three.

The following subsections describe these methods and their hypotheses as well as the case of study, experimental scenarios and the experimental testers sample.

4.2.1. Stopwatch time, errors and accuracy studies

Stopwatch **time**, **errors** and **accuracy** studies aim to analyse the effect of the proposed authoring **solution** (ARR) on reporting effectiveness and efficiency compared to alternative solutions (ARN, TBR, TBN) in different **failure** conditions (CNN and TEM). Stopwatch studies consist of testers performing diagnosis reporting procedures regarding different **failures**, which collect quantitative data regarding the abovementioned criteria. In order to validate this research's contributions (augmented content and recommendations), these studies evaluate the following hypotheses:

1. "Recommendations **accuracy** improves with AR-based recommendations compared to non-AR recommendations". It means that ARR should be better than TBR in terms of recommendations accuracy.
2. "Reporting **errors** reduce with the use of augmented content compared to non-AR reporting solutions". Thus, ARR and ARN should reduce reporting errors compared to TBR and TBN.
3. "Reporting **errors** do not vary with the use of recommendations compared to non-recommender reporting solutions". It means that ARR should be similar to ARN and TBR to TBN in terms of reporting errors.
4. "Reporting **time** decreases with the use of AR content compared to non-AR reporting solutions". Thus, ARR and ARN should reduce reporting time compared to TBR and TBN.
5. "Reporting **time** decreases with the use of recommender methods compared to non-recommender solutions". It means that ARR should reduce reporting time compared to ARN and TBR should do similarly compared to TBN.

Table 3 defines the quantitative variables relevant in these studies, while Table 4 declares factor variable levels to evaluate validity of these studies' hypotheses. These stopwatch studies employ the case studies from [52] because they are focused in

Table 3

Description of measured response and factor variables in reporting stopwatch studies.

Variable	Type	Definition
Time	Response	Number of seconds taken by a tester to complete a diagnosis reporting task of a given failure
Errors	Response	Number of mistakes when inputting failure data made by testers when conducting diagnosis reporting steps
Accuracy	Response	Number of times reporting fault condition is recommended and selected by a tester in a diagnosis reporting step
Failure	Factor	Incorrect asset behaviour that triggers a diagnosis reporting task
Solution	Factor	Reporting tool used by a tester to conduct a diagnosis reporting step
Completeness	Factor	Completion of each step and processes by all the users

Table 4

Description of relevant factors' levels in reporting stopwatch studies.

Factor	Level	Definition
Failure	CNN	Electrical failure to which the experimented asset is setup to for testers to conduct diagnosis reporting procedures
	TEM	Electronic failure to which the experimented asset is setup to for testers to conduct diagnosis reporting procedures
Solution	ARR	Proposed AR solution that includes content formats and recommendations for knowledge capture applications
	ARN	Alternative AR solution that includes content formats for knowledge capture applications but not recommendations
	TBR	Alternative non-AR solution that includes recommendations for knowledge capture applications
	TBN	Alternative non-AR solution that does not include recommendations for knowledge capture applications

Table 5

Overview of stopwatch experimental groups.

	ARR	ARN	TBR	TBN
CNN	A	C	D	B
TEM	B	D	C	A

diagnosis-related tasks. These case studies refer to the **failures** that testers need to report as part of the experiments and their reporting attributes are later described in Section 4.3. Finally, they also do not consider testers' IT expertise level as a relevant factor. The decision was related to the scope of the experiments. They aimed to evaluate the impact of content formats and recommendations in reporting (data input) tasks in terms of **time** and **errors**. As part of the experimental protocol (Section 4.4) testers were briefed on the reporting tasks to conduct (what data to input about the **failure** occurred). This was to avoid potential reporting errors due to lack of testers' knowledge that would have impacted the results. Thus, testers' expertise level seemed irrelevant as they were to have enough background on tasks to perform.

Stopwatch studies aimed to test the proposed reporting **solution** (ARR) against alternative **solutions** (ARN, TBR and TBN) in diagnosis reporting tasks about two different **failures**. The experimental procedure consists of three reporting tasks, each one comprising an ontology individual to instantiate ('Step', 'evaluatesState', and 'diagnosisState' –Fig. 5). Fig. 5 describes the experimental setup, including the hardware devices embedded with the reporting **solutions** and the case study asset (Section 4.3).

Each experiment consisted of a tester reporting one **failure**, using one reporting **solution**. Twenty-eight testers completed two experiments with different **solutions**, one for each **failure**. Testers were randomly allocated in one of four different groups according to experimental factors. Table 5 presents these groups according to their **failure** and **solution**. These **failures** (Section 4.3) and **solutions** can be considered sufficiently different in nature to not expect carry-over effects between experiments. So, the experimental design can be considered between-subjects for further statistical analysis.

4.3. Cases of study

This case of study comprises two diagnosis reporting tasks (electric and electronic) of no-fault-found scenarios in a complex-engineering asset. The reason to select no-fault-found scenarios was to reflect in validation experiments the complexity of reporting tasks. Fig. 6 presents a picture of the case study's asset. The

asset, named Helicopter Mission System (HMS), is a replica of an electronic system whose aim is to control the navigation mission of a helicopter. This replica was built with the same specifications as the original in order to enable laboratory experimentation. This system comprises three computers, one camera and an ethernet switch to connect them altogether. The first computer, called 'main mission computer', is used as controller for the rest of the elements and also controls the navigational parameters of the helicopter. The second computer, or 'client mission computer', is that used by helicopter pilots set the navigation mission. The third computer, which acts merely as a 'monitor', is that from which pilots control the 'client mission computer'. The 'camera' is there to provide pilots with a visual of the terrain while handling the helicopter. Finally, the 'ethernet switch' aims to connect the main mission computer to the client mission computer, the 'monitor' and the 'camera' for further control. The system comes with an integrated control monitoring system that evaluates electronic performance parameters from its different elements. Due to its criticality for piloting the helicopter, real-life maintainers are very careful when reporting diagnosis procedures on it. Besides, the control monitoring has some limitations regarding the electronic parameters it can control due to system's configuration.

These system's complexities make them suitable for this research's experimental methods as they make these systems prone to incur in no-fault-found scenarios, which are difficult to diagnose and report according to expert testers. Moreover, these allowed to identify two common no-fault-found scenarios to be used as experimental procedures. The following subsections explain these failures and their reporting procedures.

4.3.1. Electric failure reporting scenario (CNN)

Fig. 7 describes the most mentioned electronic failure in reporting tests regarding the HMS. The failure is caused by a fault in the cable that connects the "main computer" with the "ethernet switch". This failure provokes a no-fault-found condition because that cable cannot be monitored by the system's control module. The control module is managed by the "main computer" and one of its limitations is that it cannot evaluate its own connectivity to the "ethernet switch". When this disconnection occurs, the control module shows the rest of the system's components ("client computer", "monitor" and "camera") as disconnected even though the connectors are in good condition. As shown in Fig. 7, connectivity is measured through time, which is identified as zero by the control module when disconnected. The experimental reporting tasks are also shown in Fig. 7. Testers were asked to report the failure's root cause (blue), including a 'Step' and its 'evaluated' and 'diagnosed' 'States'.

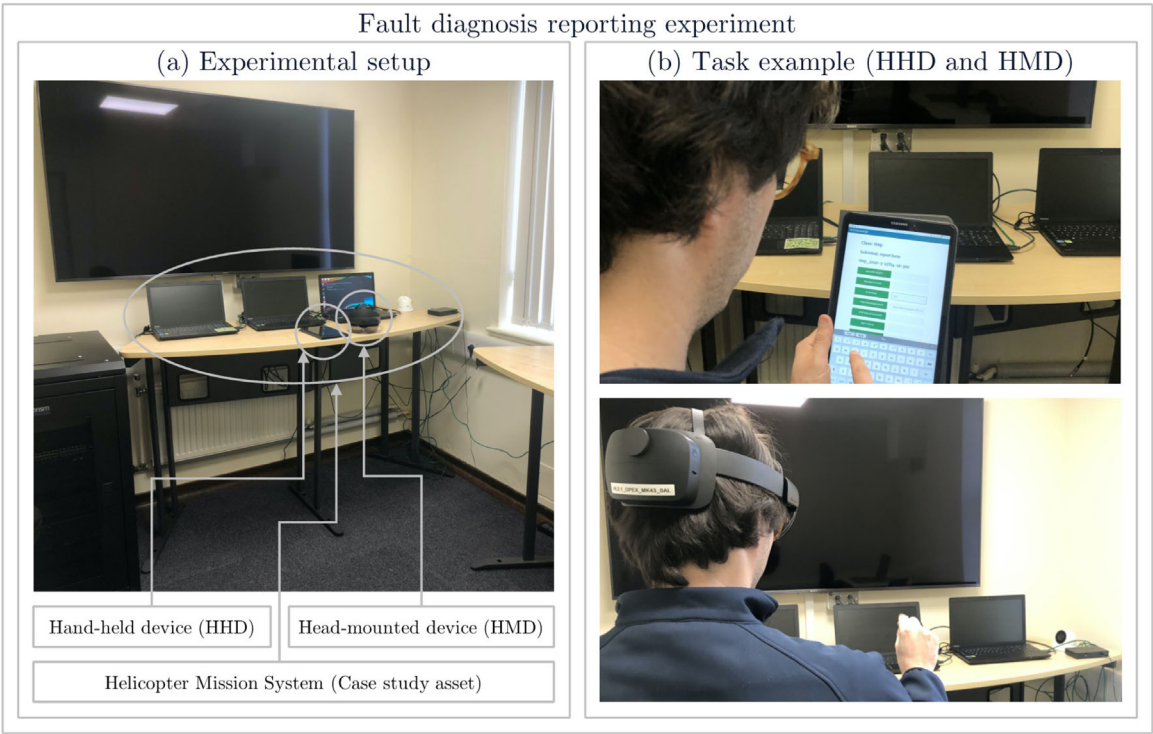


Fig. 5. Description of diagnosis reporting experiment: (a) experimental setup and (b) task examples with hand-held (HHD) and head-mounted (HMD) devices.

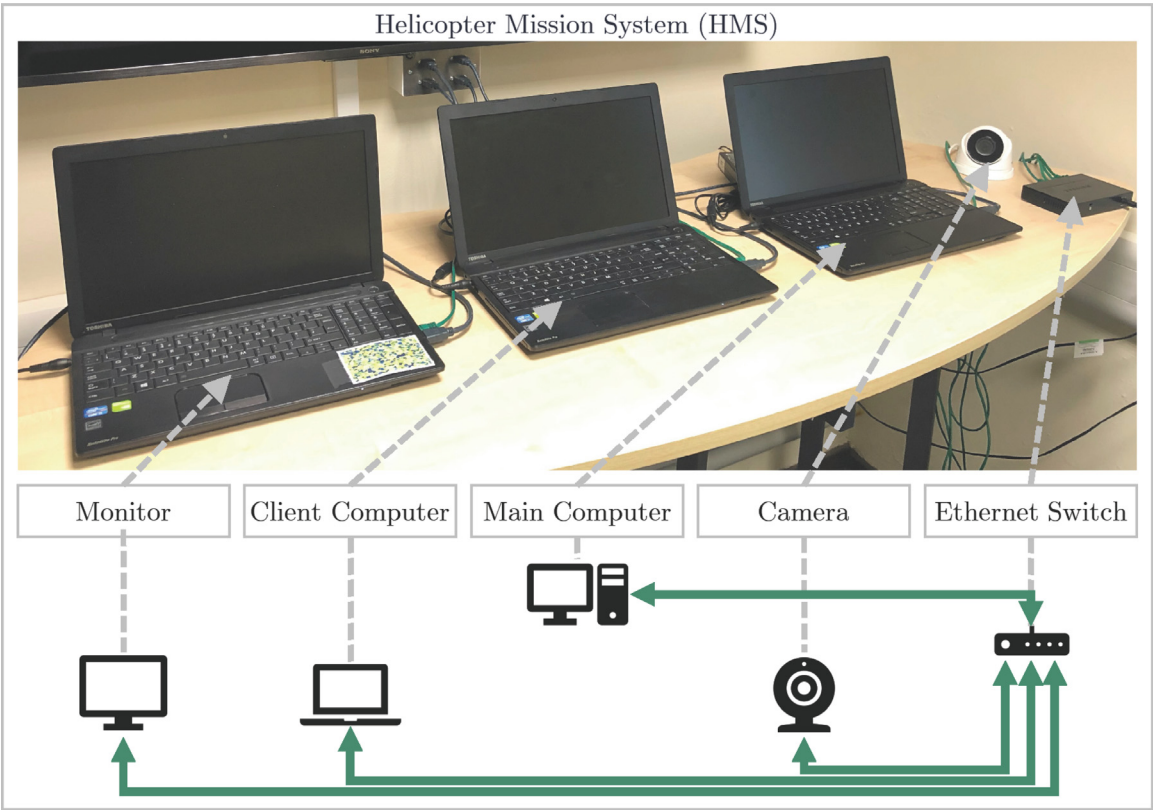


Fig. 6. Overview of diagnosis reporting case of study: Helicopter Mission System (HMS) – It displays a picture of the HMS prototype at the top along with a schema of its components at the bottom.

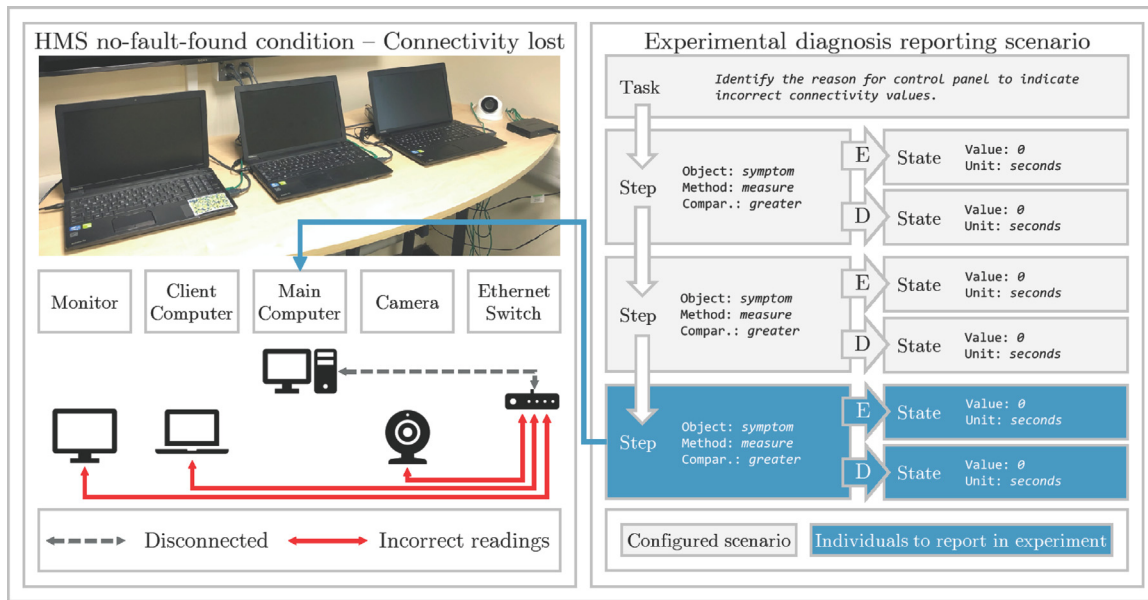


Fig. 7. A graphical representation of the system components (left-side) as well as diagont's classes and object and datatype properties assertions pre-configured (grey – used by the recommender method) and to be reported (blue – by the tester) within the experiment.

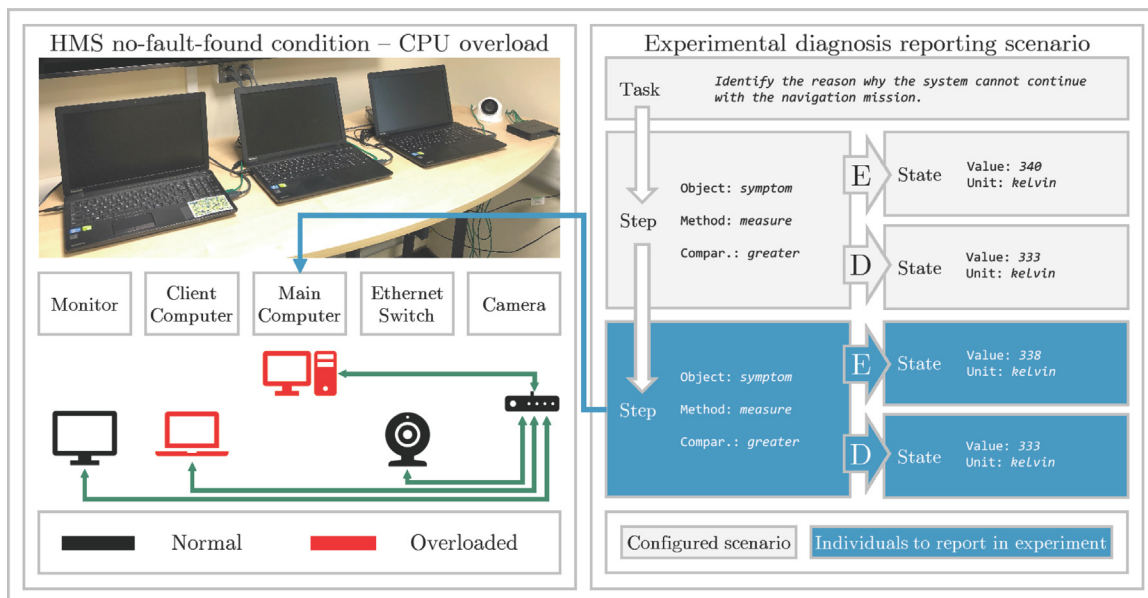


Fig. 8. A graphical representation of the system components (left-side) as well as diagont's classes and object and datatype properties assertions pre-configured (grey – used by the recommender method) and to be reported (blue – by the tester) within the experiment.

4.3.2. Electronic failure reporting scenario (TEM)

Fig. 8 describes the most mentioned electric failure in expert interviews regarding the HMS. It is another example of no-fault-found because the thresholds established by the system's control module are higher than the values that cause the failure. The failure consists of a hardware overload caused by too many software applications being run in the HMS simultaneously. When this occurs, both "main computer" and "client computer" reach CPU temperatures higher than 60° Celsius (~333 K). However, the system's control module cannot detect this issue because the CPU temperatures monitoring thresholds are set for each CPU independently at a temperature of 90° Celsius. Similar to previous failures, **Fig. 8** shows a simplified version of the failure report including the steps to be reported by experimental testers (blue).

Fig. 9 shows the example of reporting task and the diagnosis information to be reported on the Hololens. Some values for attributes and relationships need to be defined by either creating new or choosing from the recommended values. **Table 6** shows the value for each property in every step of the diagnosis task for both tasks that must be filled in by the participants and the number of options to choose from.

4.3.3. Experimental sample and protocol assumptions

A total of 28 MSc students (19 males and 9 females) participated as testers in laboratory experiments. Their ages range from 21 to 30 years and they are all enrolled in engineering-related MSc degrees. Although they have some basic knowledge in AR, ontology-based methods and maintenance due to their

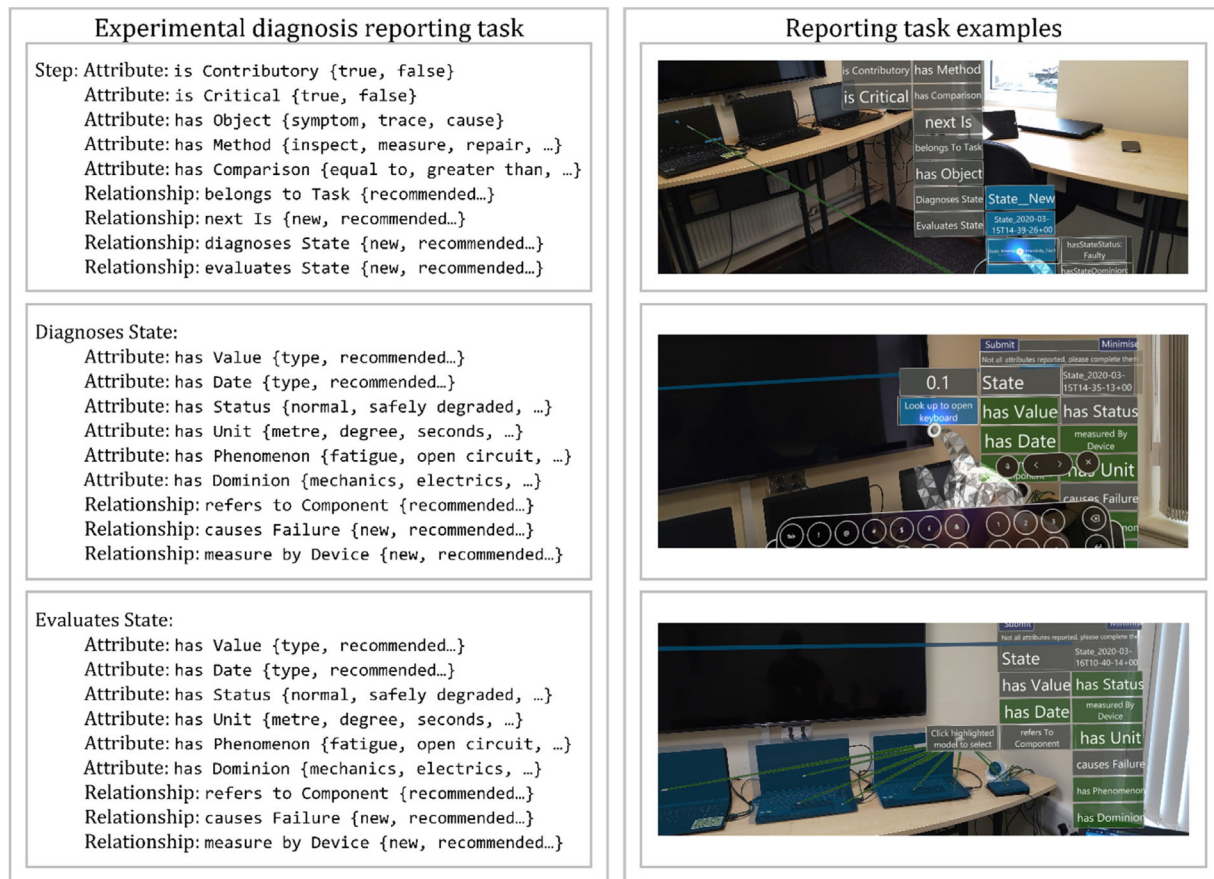


Fig. 9. Creating a new individual included fulfilling its relationships and attributes (e.g., 'Unit') by typing values or selecting one of the recommended options (e.g., 'Seconds').

Table 6

Properties' value needs to be filled by participants and the number of options to choose from.

Individual	Property	CNN correct result	TEM correct result	Number of options
Step	is Contributory	true	true	2
Step	is Critical	true	true	2
Step	has Object	cause	cause	3
Step	has Method	measure	measure	4
Step	has Comparison	equal to	greater than	6
Step	belongs to Task	Test	Test	10
Step	next Is	–	–	not needed for the test
Step	diagnoses State	State_Monitor_Fault	State_Monitor_Overheat	5
Step	evaluates State	New	New	5
Diagnosed State	has Value	0	333	typed by tester
Diagnosed State	has Date	–	–	not needed for the test
Diagnosed State	has Status	faulty	faulty	4
Diagnosed State	has Unit	seconds	kelvin	14
Diagnosed State	has Phenomenon	signal error	thermal shock	14
Diagnosed State	has Dominion	electrics	electronics	6
Diagnosed State	refers to Component	Monitor	Monitor	
Diagnosed State	causes Failure	–	–	not needed for the test
Diagnosed State	measured by Device	–	–	not needed for the test
Evaluated State	has Value	0	338	typed by tester
Evaluated State	has Date	–	–	not needed for the test
Evaluated State	has Status	faulty	faulty	4
Evaluated State	has Unit	seconds	kelvin	14
Evaluated State	has Phenomenon	signal error	thermal shock	14
Evaluated State	has Dominion	electrics	electronics	6
Evaluated State	refers to Component	Monitor	Monitor	
Evaluated State	causes Failure	–	–	not needed for the test
Evaluated State	measured by Device	–	–	not needed for the test

courses, they have no previous hands-on experience in any of them. So, they were given a short training on AR devices right before experimentation to avoid the presence of any learning

curves. Testers were randomly allocated to one of the four groups (A (7), B (7), C (7) or D (7) –Table 6) to avoid “carry-over” effects between failures while using two different reporting solutions.

This experimental protocol aims to validate this research's proposed methods against its expected contributions. For this validation to be coherent, there are few assumptions to consider:

- In order to keep consistency within experiments, these were conducted in a laboratory environment to maintain constant other factors (e.g. ergonomics or lighting conditions) that may affect the results. Hence, these factors were considered out of these experiments' scope.
- Experimental sample size for the abovementioned statistical tests can be estimated "a priori". Such estimation can be done using a F test for the most requiring analytical test (two-way ANOVA between-subjects). With 4 factor groups (failure and solution), a variance of 0.3 (partial eta squared), a type-I error of 0.1 (alpha) and a power of 0.9 (1 - beta), the resultant sample size is 31 people. That is quite close to the 28-sample size used in these experiments. These numbers are similar to those achieved by similar researches [4,18] (30-sample size).
- Testers are MSc students with none or very little experience in AR or maintenance. Although this ensures a baseline for measuring reporting effectiveness and efficiency, further experiments should be required to corroborate laboratory results in real-life working conditions with real maintainers to ensure validity of these hypotheses.

This protocol's results are discussed in the following section.

4.4. Results

This section aims to discuss experimental results regarding the validity of research hypotheses presented in Section 4.2. Its results are analysed and discussed in the following subsections to evaluate this research's hypothesis validity. The complete datasets and analyses can be found at [10.17862/cranfield.rd.12382604](https://doi.org/10.17862/cranfield.rd.12382604).

4.4.1. Recommendations accuracy study

The aim of this study was to validate the following hypothesis: "Recommendations accuracy improves with AR-based recommendations compared to non-AR recommendations". Recommendations accuracy is defined as the number of times recommender methods suggested failures' root causes and these were reported by testers. Stopwatch experiments consisted of testers completing diagnosis reporting procedures about two different failures (CNN and TEM – Section 4.3) using different reporting tools. Since stopwatch experiments only offered one recommendation per experiment, recommendation accuracy values can range between 0 and 1. Two of the experimented reporting tools were AR (ARR) and non-AR (TBR) recommender solutions. According to this study's aim, this paper's proposal (ARR) was hypothesised to provide higher accuracy than TBR.

Fig. 10 presents average recommendation accuracy results group by solution and failure factors. These results suggest a considerable difference in accuracy between ARR and TBR for both, electric (CNN) and electronic (TEM) failures. In electric failure experiments, recommendations accuracy was almost double for ARR (0.857) compared to TBR (0.429). In electronic failure experiments, accuracy was 2.4 times better for ARR (0.714) compared to TBR (0.286). These differences can be considered statistically significant ($p\text{-value} = 0.022$) with a confidence interval of 95% ($p\text{-value} < 0.05$) according to the results of t-test (between-subjects) presented in Table 7.

Overall, recommendations accuracy analyses indicate that the proposed AR-based content-aware recommender (ARR) has improved accuracy compared to a more conventional ontology-based (TBR) recommender. Thus, demonstrating the validity of

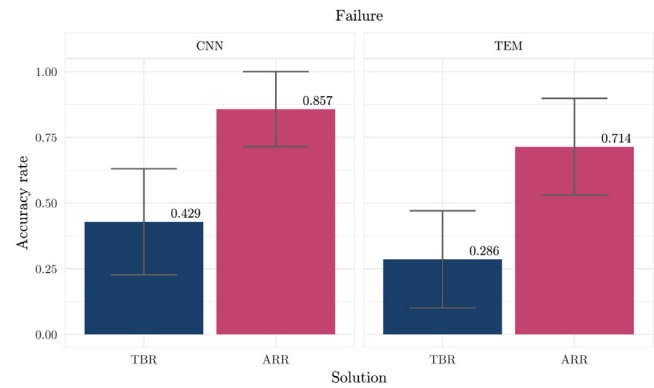


Fig. 10. Average recommendations accuracy for TBR and ARR solutions per experimental failure, with the standard deviation represented by error bars.

Table 7

Between-subjects t-test results on recommendations accuracy variance per solution.

Solution	Testers	Mean	Std. dev.	T	Df	p-value	Significant (95%)
ARR	14	0.786	0.426	2.449	25.39	0.0215	Yes
TBR	14	0.357	0.497				

this study's hypothesis. Based on the definition of recommendations accuracy, there are two complementary explanations for these results. First, AR-based hybrid recommendations (ARR) are more precise and provide correct suggestions more often than conventional ontology-based tools (TBR). Second, augmented content formats provide easier visualisation of recommended fault conditions allowing the tester to choose correctly more often. Future works can investigate the independent effect of each cause through experimentation in real-life conditions.

4.4.2. Errors effect study

The aim of the errors effect study was to validate the two following hypotheses:

- "Reporting errors reduce with the use of augmented content compared to non-AR reporting solutions".
- "Reporting errors do not vary with the use of recommendations compared to non-recommender solutions".

Stopwatch experiments also counted reporting errors, which aimed to measure reporting effectiveness as the number of testers' mistakes in data input tasks. The number of data input tasks varies with the ontology classes being instantiated at each experiment. Hence, errors rates are the percentage of errors by total number of data input tasks per reporting experiment. According to the hypotheses above, errors rates are expected to decrease with AR (ARR, ARN) compared to non-AR (TBR, TBN) reporting solutions but be similar among recommender and non-recommender (ARR vs ARN and TBR vs TBN) solutions.

Fig. 11 displays average errors rates per experimental solution and failure. These results indicate a considerable difference between AR (ARN, ARR) and non-AR (TBN, TBR) solutions but no relevant effect of recommendations (ARR vs ARN, and TBR vs TBN) in terms of reporting errors. For both experimental failures, average errors rates vary similarly for each solution. Non-AR reporting tools had errors rates ranging between 19%–21%, while AR-based reporting methods achieved smaller errors rates ranging between 10%–12%. Besides, recommender solutions (ARR, TBR) had slightly higher errors rates compared to their non-recommender counterparts (ARN, TBN). This can be caused due to

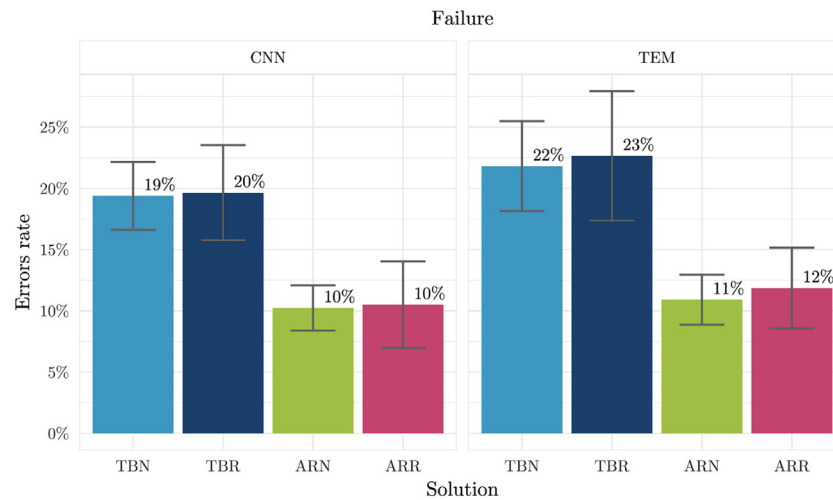


Fig. 11. Average reporting errors rates for TBN, TRB, ARN and ARR solutions per experimental failure, with the standard deviation represented by error bars.

Table 8

Two-way ANOVA between-subjects test results on errors rates for failure and solution factors.

Factor	Df	Sum Sq	Mean	F Value	Pr (>F)	Significant (95% ci)
Failure	1	0.0020	0.00015	0.010	0.9215	No
Solution	3	0.2439	0.08131	5.263	0.0019	Yes
Failure:Solution	3	0.0010	0.00035	0.022	0.9954	No
Residuals	122	1.8848	0.01545			

Table 9

Significance (p-value) results on post hoc comparisons (Tukey HSD) between solution factors in errors rates results.

	TBN	TBR	ARN	ARR
TBN	–	0.99996	0.00779	0.05976
TBR	0.99996	–	0.01886	0.09085
ARN	0.00779	0.01886	–	0.98113
ARR	0.05976	0.09085	0.98113	–

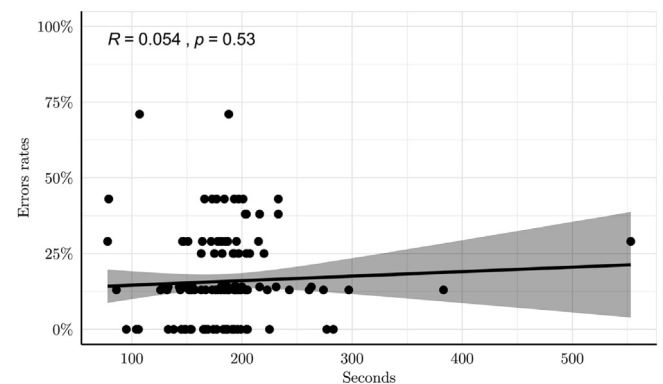


Fig. 12. Scatter plot displaying error rates versus completion time in seconds to show their correlation, with Pearson's correlation test results at the top left.

the impact of recommendations on the number data input tasks, which decrease with the use of recommendations.

Further analyses can identify the significance of the differences discussed above. Table 8 presents a two-way ANOVA between-subjects test conducted to analyse errors rates variance with failure and solution experimental factors. According to its results, it can be said with a confidence interval of 95% ($p\text{-value} < 0.05$), that 'solution' is a significant factor ($p\text{-value} = 0.0019$) while 'failure' and their interaction (solution-failure) are not. Moreover, post-hoc comparisons results from the Tukey HSD test (Table 9) can help to evaluate differences between solutions on errors rates results. These indicate that ARR and ARN errors rates are significantly different ($p\text{-value} < 0.05$) to TBR and TBN. They also show no significant differences between ARR and ARN, and TBR and TBN.

This studies' hypotheses assume reporting errors and time to measure respectively effectiveness and efficiency. For this assumption to be true, these variables should not be correlated. Fig. 12 presents Pearson's correlation test results. With a confidence interval of 95% ($p\text{-value} < 0.05$), it can be said that the correlation is not significant ($p\text{-value} = 0.53$). So, the assumption regarding errors as effectiveness measure and time as efficiency measure can be considered valid.

Overall, previous discussions support validity of the following considerations regarding the effect on completion errors of AR-based and recommendable reporting solutions:

- Correlation between reporting errors and times results cannot be considered significant.
- Errors can be considered a measure of reporting effectiveness and time a measure of reporting efficiency.
- Differences in errors rates among experimental failures cannot be considered statistically significant.
- Errors rates for AR reporting tools (ARR, ARN) are half of those from non-AR reporting solutions (TBR, TBN).
- Differences in errors rates between AR and non-AR reporting solutions can be considered statistically significant.
- Errors rates for recommender tools (ARR, TBR) are 10% higher than those from non-recommender reporting solutions (ARN, TBN).
- Differences in errors rates between recommender and non-recommender solutions cannot be considered statistically significant.
- Increase on errors rates for recommender tools can be caused due to the impact of recommendations in the number of data input tasks.

These results indicate the validity of this research's hypotheses regarding the effect of AR content and recommenders in reporting effectiveness. AR content formats for knowledge capture enabled data input methods with embedded correctness-checking features, which can have a positive effect on data input mistakes. Instead, recommenders reduce the number of data input tasks and so, their effect cannot be considered significant in reducing their errors.

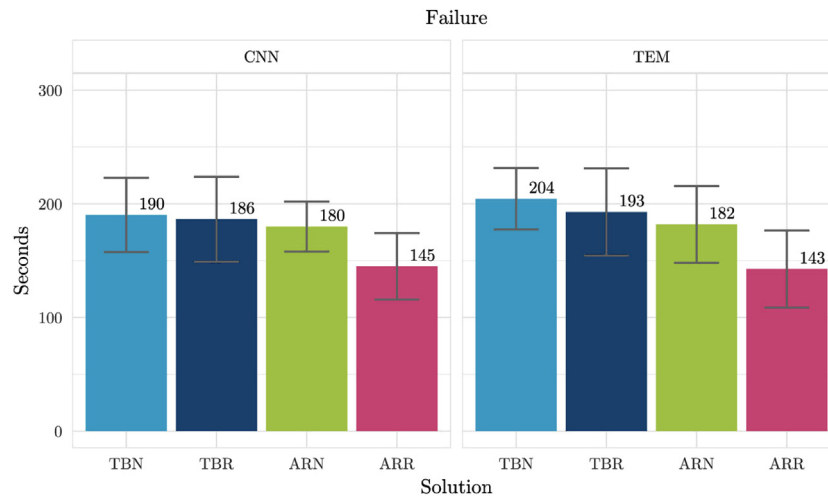


Fig. 13. Average reporting times in seconds for TBN, TBR, ARN and ARR solutions per experimental failure, with the standard deviation represented by error bars.

4.4.3. Time effect study

The aim of the time effect study was to validate the two following hypotheses:

- “Reporting time decreases with the use of AR content compared to non-AR reporting solutions”.
- “Reporting time decreases with the use of recommendations compared to non-recommender solutions”.

Stopwatch experiments measured reporting time to evaluate reporting efficiency. Time is defined as the number of seconds taken by a tester to complete reporting tasks regarding the diagnosis of a failure's root cause (CNN and TEM). Testers employed diverse reporting solutions (ARR, ARN, TBRN and TBN – Section 4.2) to accomplish these tasks. According to this study's hypothesis, this paper's proposal (ARR) is expected to obtain lowest reporting times, TBN the highest and TBR and ARN in between the other two.

Fig. 13 presents average reporting times per experimental solution and failure. These results show considerable differences between the proposed solution (ARR) and non-AR (TBR) and non-recommendable (ARN, TBN) alternatives in both failure experiments. In electric failure experiments (CNN), ARR testers were 45 s faster than TBN's, while ARN's and TBR's were 10 and 4 s faster than TBN testers. In electronic failure experiments (TEM), results also show similar differences. ARR testers are 61 s faster than TBN, while ARN's are 22 and TBR's are 11 s faster than TBN testers. These results indicate that both AR content and recommendations have a positive impact on reporting times, according to the comparison of ARN and TBR against TBN results. Moreover, these results suggest that the combined effect of both, AR content and recommendations, has a greater impact than each independently, according to the comparison of ARR against ARN and TBR results.

Further analyses can identify the statistical significance of differences discussed above. Table 10 presents the results of a two-way ANOVA between-subjects test conducted to analyse reporting time variances with failure and solution experimental factors. According to its results, the solution factor effect is statistically significant (p -value = $4.54e-07$) while the failure factor cannot be considered significant (p -value = 0.08) for a confidence interval of 95% (p -value < 0.05). Moreover, post hoc comparisons results from the Tukey HSD test (Table 11) can help to evaluate differences between solutions on reporting times per experimental failure. Comparison results indicate that ARR is significantly different to other alternatives in electric failure experiments,

Table 10

Two-way ANOVA between-subjects test results on times for failure and solution factors.

Factor	Df	Sum Sq	Mean	F Value	Pr (>F)	Significant (95% ci)
Failure	1	006072	06072	02.964	8.75e-02	Yes
Solution	3	074638	24879	12.145	4.54e-07	Yes
Failure:Solution	3	010500	03500	01.708	1.68e-01	No
Residuals	132	270415	02049			

and significantly different to non-AR alternative solutions (TBN and TBR) in electronic failure experiments. Besides, the non-recommender AR solution (ARN) is not significantly different on reporting times achieved to non-AR alternatives in any experiment. Hence, it can be said that AR achieves improved reporting times when implementing both, AR content and context-aware recommendations.

Overall, previous discussions support validity of the following considerations regarding the effect on reporting time of AR-based and recommendable reporting solutions:

- Differences in times between experimental failures cannot be considered significant.
- Differences among AR-recommender (ARR) solutions and alternative solutions (ARN, TBR, TBN) can be considered significant.
- Differences among AR-non-recommender (ARN) solutions and alternative solutions (TBR, TBN) cannot be considered significant.
- In electric failure experiments, average reporting time for ARR is 20%–23% faster than alternative solutions (ARN, TBR, TBN).
- In electronic failure experiments, average reporting time for ARR is 22%–38% faster than alternative solutions (ARN, TBR, TBN).

These considerations indicate the validity of this research's hypotheses regarding the effect of AR content and recommendations in reporting efficiency. AR content formats enabled significantly faster reporting times but only when implementing context-aware AR-based recommendations.

4.5. Discussion

Previous analytical results aimed to evaluate this research's hypothesis (Section 4.2) for demonstrating the validity of this research's contributions (Section 4). The first validation hypothesis

Table 11Significance (*p*-value) results on post hoc comparisons (Tukey HSD) between solution and failure factors groups in time results.

	CNN:ARN	CNN:ARR	CNN:TBN	CNN:TBR	TEM:ARN	TEM:ARR	TEM:TBN	TEM:TBR
CNN:ARN	–	0.33708	0.99571	0.99990	1.00000	0.25393	0.01221	0.99151
CNN:ARR	0.33708	–	0.08181	0.23972	0.26759	1.00000	0.00001	0.10554
CNN:TBN	0.99571	0.08181	–	1.00000	0.99891	0.05421	0.10068	1.00000
CNN:TBR	0.99990	0.23972	1.00000	–	0.99999	0.17890	0.11494	0.99995
TEM:ARN	1.00000	0.26759	0.99891	0.99999	–	0.19622	0.01922	0.99700
TEM:ARR	0.25393	1.00000	0.05421	0.17890	0.19622	–	0.00000	0.07378
TEM:TBN	0.01221	0.00001	0.10068	0.11494	0.01922	0.00000	–	0.27459
TEM:TBR	0.99151	0.10554	1.00000	0.99995	0.99700	0.07378	0.27459	–

stated that “recommendations accuracy improves with the use of AR-based hybrid recommendations (ARR) compared to conventional ontology-based recommendations (TBR)”. Accuracy effect study (Section 4.1) analysed the relation between recommendations proposed and chosen by testers and recommender methods in diagnosis reporting experiments. Its results showed that the proposed method’s accuracy (ARR) was 2.2 times significantly higher than conventional recommender’s (TBR) on average for both experimental failures. Therefore, this hypothesis can be considered valid within the context of the experiments conducted. This research work extend the existing knowledge on ontology-based diagnosis [21,22,49] and presents a context-aware and ontology-based AR recommendations to improve efficiency of diagnosis reporting tasks through knowledge re-use. These results imply that the proposed context-aware and ontology-based AR recommender method enhances knowledge re-use in diagnosis reporting tasks. The identified accuracy improvements can be the result of: (1) more precise recommendations and (2) more correct testers’ selections caused by contextualised visualisations. These capabilities appear to address the diagnosis efficiency problem in knowledge integration without overloading domain experts, especially when there are many regions to diagnose and report [23]. Furthermore, these capabilities also play an important part in minimising latent usability-related design flaws identified in medical equipment [41]. The knowledge-reuse from the database and the analysis of the maintenance data are the key components to help organisations identify whether a fault is usability-related. Hence, it enables an effective mitigation strategy to reduce the frequency of NFF reporting [41]. Future studies can investigate the independent effects of each cause in real-life experiments to quantitatively measure their independent impacts.

The second validation hypothesis assumed that “errors are a measure of reporting effectiveness and time is a measure of reporting efficiency”. Pearson’s correlation test’s results indicated that the correlation between these response variables could not be considered statistically significant. Moreover, Cohen’s interpretation of correlation test’s results suggested that even if correlation was significant, it was small. Hence, it can be said that within the context of this research’s experiments the assumption above can be considered valid.

The third validation hypothesis enounced that “errors reduce with AR reporting solutions compared to non-AR ones”. And the fourth one stated that “errors reduce with recommender reporting methods compared to non-recommender ones”. Errors study analysed the effect on errors rate per reporting task of diverse reporting tools (ARR, ARN, TBR, TBN) in failure diagnosis experiments. Errors rates for AR-based tools (10%) reduced 50% compared tablet-based solutions’ errors rates (20%) in both experimental failures. Besides, two-way ANOVA test results indicated statistical significance of these differences within the experiments’ context. Therefore, the second hypothesis can be considered valid but not the third one. These results also align with this research’s contributions that expected only the proposed dynamic authoring method to improve effectiveness (errors) of

diagnosis reporting tasks. Due to the nature of experiments, errors were measured per reporting task. A possible explanation for these conclusions is that AR methods allow to contextualise complex data input tasks, while recommender methods reduce their total number. These two features have been considered critical to enable accurate evaluation process as they increase the quality of information through individual aggregation and contextualisation which lead to reduced uncertainty in associating the recommendation facets [4]. Besides, the quality of information in a useable format is a critical issue for devising optimal maintenance strategies such as the use of fault reported data for reliability study. This approach can bridge the gap for a mechanism to collect high quality information for reliability analysis and to enable data-driven condition monitoring [39]. Moreover, the proposed fault-reporting approach that led to improved reporting errors could also minimise human failures takes place in safety-related system due to ineffective fault reporting system for instance at a nuclear power plant [42]. Furthermore, the decreased in reporting errors can potentially reduce the likelihood of reported symptoms to end up in NFF which is often caused by incorrect assessment made by inexperienced engineers on the diagnostic path [40]. Future works can extend recommendation facets to different datasets for studying recommenders’ effects in singular data-input tasks.

The fifth validation hypothesis stated that “time decreases with the use of AR reporting solutions compared to non-AR ones”. And the sixth one enounced that “time decreases with the use of recommender reporting tools compared to non-recommender ones”. Time study (Section 4.3) evaluated the effect on time of AR and recommender reporting and their counterparts in failure diagnosis experiments. In electric failure experiments (CNN), ARR testers were 45 s faster than TBN’s, while ARN’s and TBR’s were 10 and 4 s faster than TBN testers. In electronic failure experiments (TEM), results also show similar differences. ARR testers are 61 s faster than TBN, while ARN’s are 22 and TBR’s are 11 s faster than TBN testers. Besides, two-way ANOVA test results indicated a significant effect of the solution factor on experimental time results. Post hoc comparisons from Tukey’s HSD tests confirmed that resultant differences were mostly driven by the variances between the proposed reporting tool (ARR) and different alternatives. Overall, these results suggest the validity of both time-related hypothesis together as the proposed AR-recommender solution was found faster than its counterparts in diagnosis reporting experiments. These results imply that the proposed AR recommender and dynamic authoring methods improve efficiency of diagnosis reporting tasks. They also suggest the need to correlate AR content and recommendations to improve their efficiency. Thus, indicating that efficiency improvements in reporting tasks through AR are only achieved through knowledge capture and re-use combined. This also aligns with a knowledge-based Building Information Model (BIM) system that incorporate reasoning capability to search relevant information facets to achieve more efficient and effective knowledge capture [24]. Future studies can investigate this correlation more in-depth to quantitatively measure their independent effects. They

can also further corroborate this research's results applying the proposed methods to other AR-maintenance knowledge capture applications.

5. Conclusions and future works

This paper proposed (1) a method to provide context-aware and ontology-based AR recommendations to reduce extensive selection lists in diagnosis reporting tasks and (2) a method to dynamically create and allocate content in augmented scenes to avoid augmented content overload and ease data input tasks. Their aim was to prove that automatic recommendable authoring can improve efficiency and effectiveness of AR knowledge capture and re-use in diagnosis reporting applications. They were implemented in a cloud-based AR system prototype for validation with effectiveness and efficiency experiments in reporting diagnosis operations. Experimental results indicated that the proposed AR-recommender reporting method (ARR) reduces reporting errors (50%) and time (20%) compared to alternative non-AR (TBR) and non-recommender (TBN, ARN) solutions. These results also displayed that recommendations' accuracy doubles for the proposed AR-based hybrid techniques (ARR) compared to conventional ontology-based methods (TBR). Besides, surveys results suggested that testers perceived the proposed reporting solution as more effective and satisfactory than its non-AR and non-recommender counterparts. Thus, proving that the proposed methods can improve effectiveness and efficiency of diagnosis reporting applications.

The proposed methods for automatic recommender authoring contribute to fill an important research gap towards the integration of human operations in digital maintenance. Maintenance reporting operations are performance and mentally demanding and so, prone to errors in efficiency-challenging conditions. These often result in reports with decreased accuracy and unstructured knowledge difficult to re-use. The proposed hybrid recommender can increase the accuracy of recommendations, reducing the size of selection lists to decrease reporting time. The proposed dynamic authoring method can reduce augmented scene's overload to further reduce reporting time. Besides, it can automatically produce augmented content that checks correctness and eases identification of reported data to reduce reporting errors. Hence, this research's proposal can contextualise and structure diagnosis reporting tasks and increase the correctness of reported data. Thus, enhancing the digitalisation of diagnosis reporting operations and facilitating capture and re-use of human knowledge in digital maintenance.

Future works will explore further applications and enhancements of the proposed methods for pursuing human knowledge integration in digital maintenance. The following extends the future works described within this paper's discussions. **Dynamic content allocation:** (1) Investigate factors that can cause occlusion in AR maintenance applications and improve proposed allocation and scaling mechanisms to reduce it. (2) Study dynamics of AR knowledge capture applications and improve proposed allocation mechanisms to enhance content navigation. **Content formats:** (1) Develop advanced methods to determine input data correctness for further reducing reporting errors and improving effectiveness. (2) Develop adaptive content formats according to user and environmental conditions (e.g. performance or light) for further decreasing reporting time and improving efficiency. **Recommendation facets:** (1) Extend the proposed recommendation framework to different techniques (e.g. collaborative filtering) and implement automatic data collection methods (e.g. content-tracing or eye-tracking) for improving recommendations accuracy. (2) Develop new content formats and recommendation facets for different maintenance reporting operations

(e.g. service logs) to extend integration of human knowledge in digital maintenance.

AR technologies are information visualisation tools that can smooth knowledge transfer between humans and digital systems. Future works aim to find necessary research towards a framework for automatic recommender authoring for AR knowledge transfer and capture applications. The object of the future study should also address AR interfaces usability aspects (e.g. improving the aspect of ease of use, intuitiveness, etc.). Thus, envisioning a future where maintenance digital systems can integrate human knowledge to its full extent.

CRediT authorship contribution statement

Iñigo Fernández del Amo: Software, Validation, Formal analysis, Data curation, Writing – original draft. **John Ahmet Erkoyuncu:** Conceptualisation, Methodology, Writing – original draft. **Maryam Farsi:** Conceptualisation, Writing – original draft. **Dedy Ariansyah:** Conceptualisation, Validation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was funded by the EPSRC, UK as part of the “Digital Toolkit for optimisation of operators and technology in manufacturing partner-ships” project (DigiTOP; <https://digitop.ac.uk/>; EP/R032718/1). Data underlying this study can be accessed through the Cranfield University repository at <http://dx.doi.org/10.17862/cranfield.rd.17814002>.

References

- [1] C. Cimino, E. Negri, L. Fumagalli, Review of digital twin applications in manufacturing, *Comput. Ind.* 113 (2019) 103130, <http://dx.doi.org/10.1016/J.COMPIND.2019.103130>.
- [2] A. Angelopoulos, E.T. Michailidis, N. Nomikos, P. Trakadas, A. Hatziefremidis, S. Voliotis, et al., Tackling faults in the industry 4.0 era—a survey of machine-learning solutions and key aspects, *Sensors (Switzerland)* 20 (2020) 1–34, <http://dx.doi.org/10.3390/s20010109>.
- [3] S. Khan, T. Yairi, A review on the application of deep learning in system health management, *Mech. Syst. Signal Process.* 107 (2018) 241–265, <http://dx.doi.org/10.1016/j.ymssp.2017.11.024>.
- [4] F. Longo, L. Nicoletti, A. Padovano, Ubiquitous knowledge empowers the Smart Factory: The impacts of a Service-oriented Digital Twin on enterprises' performance, *Annu. Rev. Control* 47 (2019) 221–236, <http://dx.doi.org/10.1016/j.arcontrol.2019.01.001>.
- [5] K. Zhao, H. Jiang, K. Wang, Z. Pei, Joint distribution adaptation network with adversarial learning for rolling bearing fault diagnosis, *Knowl.-Based Syst.* 222 (2021) 106974, <http://dx.doi.org/10.1016/J.KNOSYS.2021.106974>.
- [6] IBM, Augmented and Virtual Reality Research at IBM Watson - IBM 2020, 2021, https://researcher.watson.ibm.com/researcher/view_group.php?id=8071 (accessed October 13, 2021).
- [7] S. Wan, D. Li, J. Gao, J. Li, A knowledge based machine tool maintenance planning system using case-based reasoning techniques, *Robot. Comput. Integr. Manuf.* 58 (2019) 80–96, <http://dx.doi.org/10.1016/j.rcim.2019.01.012>.
- [8] D. Li, Å. Fast-Berglund, D. Paulin, Current and future Industry 4.0 capabilities for information and knowledge sharing: Case of two Swedish SMEs, *Int. J. Adv. Manuf. Technol.* 105 (2019) 3951–3963, <http://dx.doi.org/10.1007/s00170-019-03942-5>.
- [9] M. del R. Pérez-Salazar, A.A. Aguilar-Lasserre, M.G. Cedillo-Campos, U. Juárez-Martínez, R. Posada-Gómez, Processes and measurement of knowledge management in supply chains: an integrative systematic literature review, *Int. J. Prod. Res.* 57 (2019) 2136–2159, <http://dx.doi.org/10.1080/00207543.2018.1521530>.

- [10] P. Pistofidis, C. Emmanouilidis, A. Papadopoulos, P.N. Botsaris, Management of linked knowledge in industrial maintenance, *Ind. Manag. Data Syst.* 116 (2016) 1741–1758, <http://dx.doi.org/10.1108/IMDS-10-2015-0409>.
- [11] B. Liu, Z. Ding, J. Wu, L. Yao, Ontology-based fault diagnosis: A decade in review, in: *ACM Int Conf Proceeding Ser.*, 2019, pp. 112–116, <http://dx.doi.org/10.1145/3307363.3307381>.
- [12] R. Palmarini, J.A. Erkoyuncu, R. Roy, H. Torabmostaedi, A systematic review of augmented reality applications in maintenance, *Robot. Comput. Integr. Manuf.* 49 (2018) 215–228, <http://dx.doi.org/10.1016/j.rcim.2017.06.002>.
- [13] E. Bottani, G. Vignali, Augmented reality technology in the manufacturing industry: A review of the last decade, *IJSE Trans.* 51 (2019) 284–310, <http://dx.doi.org/10.1080/24725854.2018.1493244>.
- [14] B. Bhattacharya, E.H. Winer, Augmented reality via expert demonstration authoring (AREDA), *Comput. Ind.* 105 (2019) 61–79, <http://dx.doi.org/10.1016/j.compind.2018.04.021>.
- [15] K. Ramirez-Amaro, M. Beetz, G. Cheng, Transferring skills to humanoid robots by extracting semantic representations from observations of human activities, *Artificial Intelligence* 247 (2017) 95–118, <http://dx.doi.org/10.1016/j.artint.2015.08.009>.
- [16] M. Gattullo, G.W. Scurati, M. Fiorentino, A.E. Uva, F. Ferrise, M. Bordegoni, Towards augmented reality manuals for industry 4.0: A methodology, *Robot. Comput. Integr. Manuf.* 56 (2019) 276–286, <http://dx.doi.org/10.1016/j.rcim.2018.10.001>.
- [17] X. Wang, S.K. Ong, A.Y.C. Nee, Multi-modal augmented-reality assembly guidance based on bare-hand interface, *Adv. Eng. Inform.* 30 (2016) 406–421, <http://dx.doi.org/10.1016/j.aei.2016.05.004>.
- [18] J. Gimeno, P. Morillo, J.M. Orduña, M. Fernández, A new AR authoring tool using depth maps for industrial procedures, *Comput. Ind.* 64 (2013) 1263–1271, <http://dx.doi.org/10.1016/j.compind.2013.06.012>.
- [19] H.L. Chi, Y.C. Chen, S.C. Kang, S.H. Hsieh, Development of user interface for tele-operated cranes, *Adv. Eng. Inform.* 26 (2012) 641–652, <http://dx.doi.org/10.1016/j.aei.2012.05.001>.
- [20] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee, A design science research methodology for information systems research, *J. Manage. Inf. Syst.* 24 (2008) 45–77, <http://dx.doi.org/10.2753/mis0742-1222240302>.
- [21] A. Zhou, D. Yu, W. Zhang, A research on intelligent fault diagnosis of wind turbines based on ontology and FMECA, *Adv. Eng. Inform.* 29 (2015) 115–125, <http://dx.doi.org/10.1016/j.aei.2014.10.001>.
- [22] G. Medina-Oliva, A. Voisin, M. Monnin, J.B. Leger, Predictive diagnosis based on a fleet-wide ontology approach, *Knowl.-Based Syst.* 68 (2014) 40–57, <http://dx.doi.org/10.1016/j.knsys.2013.12.020>.
- [23] V. Dimitrova, M.O. Mehmood, D. Thakker, B. Sage-Vallier, J. Valdes, A.G. Cohn, An ontological approach for pathology assessment and diagnosis of tunnels, *Eng. Appl. Artif. Intell.* 90 (2020) 103450, <http://dx.doi.org/10.1016/j.engappai.2019.103450>.
- [24] I. Motawa, A. Almarshad, A knowledge-based BIM system for building maintenance, *Autom. Constr.* 29 (2013) 173–182, <http://dx.doi.org/10.1016/j.autcon.2012.09.008>.
- [25] T. Wójcicki, Supporting the diagnostics and the maintenance of technical devices with augmented reality, *Diagnostyka* 15 (2014) 43–47.
- [26] G. Avallé, F. De Pace, C. Fornaro, F. Manurí, A. Sanna, An augmented reality system to support fault visualization in industrial robotic tasks, *IEEE Access* 7 (2019) 132343–132359, <http://dx.doi.org/10.1109/ACCESS.2019.2940887>.
- [27] R. Ghimire, K.R. Pattipati, P.B. Luh, Fault diagnosis and augmented reality-based troubleshooting of HVAC systems, in: *AUTOTESTCON (Proceedings)*, Vol. 2016-October, 2016, pp. 1–10, <http://dx.doi.org/10.1109/AUTEST.2016.7589590>.
- [28] M. Khalil, C. Bergs, T. Papadopoulos, R. Wuchner, K.U. Bletzinger, M. Heizmann, IIoT-based fatigue life indication using augmented reality, in: *IEEE Int Conf Ind Informatics*, Vol. 2019-July, 2019, pp. 746–751, <http://dx.doi.org/10.1109/INDIN41052.2019.8972114>.
- [29] M.H. Priya, S.K. Vasudevan, An innovative application for car engine disparity check - A novel attempt, in: *Proc IEEE Int Conf Signal Process Commun ICSPC 2017*, Vol. 2018-Janua, 2018, pp. 175–180, <http://dx.doi.org/10.1109/ICSPC.2017.8305833>.
- [30] S. Wang, R. Guo, H. Wang, Y. Ma, Z. Zong, Manufacture assembly fault detection method based on deep learning and mixed reality, in: *Proc IEEE Int Conf Inf Autom ICIA 2018*, 2018, pp. 808–813, <http://dx.doi.org/10.1109/ICInfA.2018.8812577>.
- [31] M.P. Das, Z. Dong, S. Scherer, Joint point cloud and image based localization for efficient inspection in mixed reality, in: *IEEE Int Conf Intell Robot Syst.*, 2018, pp. 6357–6363, <http://dx.doi.org/10.1109/IROS.2018.8594318>.
- [32] R. Masoni, F. Ferrise, M. Bordegoni, M. Gattullo, A.E. Uva, M. Fiorentino, et al., Supporting remote maintenance in industry 4.0 through augmented reality, *Procedia Manuf.* (2017) <http://dx.doi.org/10.1016/j.promfg.2017.07.257>.
- [33] J. Rambach, A. Pagani, M. Schneider, O. Artemenko, D. Stricker, 6DoF Object tracking based on 3D scans for augmented reality remote live support, *Computers* 7 (2018) 6, <http://dx.doi.org/10.3390/computers7010006>.
- [34] N. Zenati-Henda, A. Bellarbi, S. Benbelkacem, M. Belhocine, Augmented reality system based on hand gestures for remote maintenance, in: *Int. Conf. Multimed. Comput. Syst. Marrakech, Morocco*, 2014, <http://dx.doi.org/10.1109/ICMCS.2014.6911258>.
- [35] J. Oyekan, V. Prabhu, A. Tiwari, V. Baskaran, M. Burgess, R. McNally, Remote real-time collaboration through synchronous exchange of digitised human-workpiece interactions, *Future Gener. Comput. Syst.* 67 (2017) 83–93, <http://dx.doi.org/10.1016/j.future.2016.08.012>.
- [36] E. Hadar, J. Shtok, B. Cohen, Y. Tzur, L. Karlinsky, Hybrid remote expert - An emerging pattern of industrial remote support, in: *CEUR Workshop Proc.* 1848, CEUR-WS, 2017, pp. 33–40.
- [37] D. Mourtzis, A. Vlachou, V. Zogopoulos, Cloud-based augmented reality remote maintenance through shop-floor monitoring: A product-service system approach, *J. Manuf. Sci. Eng. Trans. ASME* 139 (2017) 1–11, <http://dx.doi.org/10.1115/1.4035721>.
- [38] I. Fernández del Amo, J. Erkoyuncu, R. Vrabčič, R. Frayssinet, C. Vazquez Reynel, R. Roy, Structured authoring for AR-based communication to enhance efficiency in remote diagnosis for complex equipment, *Adv. Eng. Inform.* 45 (2020) 101096, <http://dx.doi.org/10.1016/j.aei.2020.101096>.
- [39] K. Leahy, C. Gallagher, P. O'Donovan, D.T.J. O'Sullivan, Issues with data quality for wind turbine condition monitoring and reliability analyses, *Energies* 12 (2019) 201, <http://dx.doi.org/10.3390/EN12020201>.
- [40] C. Hockley, L. Lacey, A research study of no fault found (NFF) in the royal air force, *Procedia CIRP* 59 (2017) 263–267, <http://dx.doi.org/10.1016/j.procir.2016.09.034>.
- [41] C.J. Flewelling, A.C. Easty, K.J. Vicente, J.A. Cafazzo, The use of fault reporting of medical equipment to identify latent design flaws, *J. Biomed. Inform.* 51 (2014) 80–85, <http://dx.doi.org/10.1016/j.jbi.2014.04.009>.
- [42] P. Pyy, An analysis of maintenance failures at a nuclear power plant, *Reliab. Eng. Syst. Saf.* 72 (2001) 293–302, [http://dx.doi.org/10.1016/S0951-8320\(01\)00026-6](http://dx.doi.org/10.1016/S0951-8320(01)00026-6).
- [43] I. Fernández del Amo, J.A. Erkoyuncu, R. Roy, R. Palmarini, D. Onoufriou, A systematic review of augmented reality content-related techniques for knowledge transfer in maintenance applications, *Comput. Ind.* 103 (2018) 47–71, <http://dx.doi.org/10.1016/j.compind.2018.08.007>.
- [44] M.L. Markus, Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success, *J. Manage. Inf. Syst.* 18 (2001) 57–93, <http://dx.doi.org/10.1080/07421222.2001.11045671>.
- [45] C.C. Aggarwal, Knowledge-based recommender systems, in: C.C. Aggarwal (Ed.), *Recomm. Syst. Textb.*, first ed., Springer, New York, NY, USA, 2016, pp. 167–197, http://dx.doi.org/10.1007/978-3-319-29659-3_5.
- [46] S.E. Schaeffer, S.E. Garza, J.C. Espinosa, S.C. Urbina, P. Nurmi, L. Cruz-Reyes, A framework for informing consumers on the ecological impact of products at point of sale, *Behav. Inf. Technol.* 37 (2018) 607–621, <http://dx.doi.org/10.1080/0144929X.2018.1470254>.
- [47] P.H. Lin, S.Y. Chen, Design and evaluation of a deep learning recommendation based augmented reality system for teaching programming and computational thinking, *IEEE Access* 8 (2020) 45689–45699, <http://dx.doi.org/10.1109/ACCESS.2020.2977679>.
- [48] M. Torres-Ruiz, F. Mata, R. Zagal, G. Guzmán, R. Quintero, M. Moreno-Ibarra, A recommender system to generate museum itineraries applying augmented reality and social-sensor mining techniques, *Virtual Real.* 24 (2020) 175–189, <http://dx.doi.org/10.1007/s10055-018-0366-z>.
- [49] N. Dendani, M. Khadir, S. Guessoum, Hybrid approach for fault diagnosis based on CBR and ontology: Using jCOLIBRI framework, in: *Proc. 2012 Int. Conf. Complex Syst. ICCS 2012*, 2012, <http://dx.doi.org/10.1109/ICCS.2012.6458509>.
- [50] D. Wang, W.H. Tang, Q.H. Wu, Ontology-based fault diagnosis for power transformers, in: *IEEE PES Gen. Meet. PES 2010*, 2010, pp. 1–8, <http://dx.doi.org/10.1109/PES.2010.5589575>.
- [51] R. Renu, D. Visotsky, S. Knackstedt, G. Mocko, J.D. Summers, J. Schulte, A knowledge based FMEA to support identification and management of vehicle flexible component issues, *Procedia CIRP* 44 (2016) 157–162, <http://dx.doi.org/10.1016/j.procir.2016.02.112>.
- [52] I.F. del Amo, Ontology-based augmented reality content related techniques and their impact in knowledge capture and re-use within maintenance diagnosis, 2020.
- [53] B. Smyth, Case-based recommendation, in: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *Adapt. Web Methods Strateg. Web Pers.*, first ed., Springer, Berlin, Heidelberg, 2007, pp. 342–376, http://dx.doi.org/10.1007/978-3-540-72079-9_11.
- [54] Z. Zhu, X. Zhou, K. Shao, A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem, *Comput. Ind. Eng.* (2019) <http://dx.doi.org/10.1016/j.cie.2019.03.022>.
- [55] O. Panzarino, *Learning Cypher*, first ed., Packt Publishing, 2014.
- [56] J. Barrasa, *Neosemantics*, 2019.
- [57] M. Eernisse, *EJS*, 2015.

- [58] L.M. Surhone, M.T. Tennoe, S.F. Henssonow, Node.js., Betascript Publishing, Beau Bassin, MUS, 2010.
- [59] Unity Technologies. Unity Game Engine. <https://unity.com> 2019.
- [60] Microsoft Corporation. Visual Studio. <https://visualstudio.microsoft.com> 2019.
- [61] A. Hejlsberg, The C# Programming Language, Addison-Wesley, 2011.
- [62] Microsoft Corporation, Mixed Reality Toolkit Documentation, Microsoft Corp, 2020.
- [63] PTC Corporation. Vuforia SDK. PTC Corp 2020.
- [64] S. Tangen, Demystifying productivity and performance, Int. J. Product. Perform. Manag. 54 (2005) 34–46, <http://dx.doi.org/10.1108/17410400510571437>.
- [65] A. Booth, D. Papaioannou, A. Sutton, Systematic Approaches to a Successful Literature Review, Sage Publications, Thousand Oaks, CA, USA, 2012.
- [66] A. Gunawardana, G. Shani, Evaluating recommender systems, in: F. Ricci, L. Rokach, B. Shapira (Eds.), Recomm. Syst. Handb., second ed., Springer, Boston, MA, USA, 2015, pp. 265–308, http://dx.doi.org/10.1007/978-1-4899-7637-6_8.



DSpace

<https://dspace.org/>

Hybrid recommendations and dynamic authoring for AR knowledge capture and re-use in diagnosis applications

Fernández del Amo, Iñigo; Erkoyuncu, John Ahmet; Farsi, Maryam; Ariansyah, Dedy

2021-12-18

Elsevier

<https://doi.org/10.1016/j.knosys.2021.107954>

Fernández del Amo I, Erkoyuncu JA, Farsi M, Ariansyah D. (2022) Hybrid recommendations and dynamic authoring for AR knowledge capture and re-use in diagnosis applications, Knowledge-Based Systems, Volume 239, March 2022, Article number 107954