Applied Artificial Intelligence
An International Journal

ISSN: 0883-9514 (Print) 1087-6545 (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

A Survey on the Evolution of the Notion of Context-Awareness

J. Augusto, A. Aztiria, D. Kramer & U. Alegre

To cite this article: J. Augusto, A. Aztiria, D. Kramer & U. Alegre (2017) A Survey on the Evolution of the Notion of Context-Awareness, Applied Artificial Intelligence, 31:7-8, 613-642, DOI: 10.1080/08839514.2018.1428490

To link to this article: https://doi.org/10.1080/08839514.2018.1428490

Published online: 20 Feb 2018.

APPLIED ARTIFICIAL INTELLIGENCE
2017, VOL. 31, NOS. 7–8, 613–642
https://doi.org/10.1080/08839514.2018.1428490

A Survey on the Evolution of the Notion of Context-Awareness

J. Augusto[a], A. Aztiria[b], D. Kramer[a], and U. Alegre[a]

[a]Research Group on Development of Intelligent Environments, Department of Computer Science, Middlesex University, London, UK; [b]Faculty of Engineering, Mondragon Unibertsitatea, Spain

ABSTRACT

The notion of *context* has been considered for a long time in different areas of Computer Science. This article considers the use of context-based reasoning from the earlier perspective of artificial intelligence as well as the newer developments in ubiquitous computing. Both communities have been somehow interested in the potential of context-reasoning to support real-time meaningful reactions from systems. We explain how the concept evolved in each of these different approaches. We found initially that each of them considered this topic quite independently and separated from each other; however, latest developments have started to show signs of cross-fertilization amongst these areas. The aim of our survey is to provide an understanding on the way context and context-reasoning were approached, to show that work in each area is complementary, and to highlight there are positive synergies arising amongst them. The overarching goal of this article is to encourage further and longer term synergies between those interested in further understanding and using context-based reasoning.

Introduction

The notion of *context* has been considered from different perspectives within Computer Science. Initially, it sparked interest within traditional artificial intelligence (AI), especially in the 1980s–1990s. Most of those efforts concentrated on discussing notations which can distinguish amongst different contexts and a way to tell the system that in different contexts, it should react accordingly; that is, the decisions of a system should be moderated and adjusted when dealing with the same decision in different contexts.

Recently, new areas have emerged with Computer Science: first pervasive computing and communications (Percomm) and ubiquitous computing (Ubicomp) (Weiser, 1991), then internet of things (IoT) (Atzori, Iera, and Morabito, 2010), ambient intelligence (AmI) (Aarts and Roovers, 2003) and intelligent environments (IE) (Augusto et al., 2013a). These

CONTACT J. Augusto j.augusto@mdx.ac.uk Research Group on Development of Intelligent Environments, Department of Computer Science, Middlesex University, NW4 2SH, London, UK.

Color versions of one or more of the figures in the article can be found online at www.tandfonline.com//UAAI areas follow more of a bottom-up approach, systems in these areas are more service oriented (see e.g., Cook, Augusto, and Jakkula, 2009). Some examples of applications driving development in those areas are domotics in smart homes, safety in ambient-assisted living (AAL), efficiency in smart offices, pedagogical support in smart classrooms, improved user experience and sales in smart shopping, improving health of those in an intensive care unit of a hospital and so forth. Once the target services are identified, an infrastructure (sensors, actuators, network, interfaces and intelligent software) is created which is capable of delivering those services. The system has to be not only reactive but also anticipatory and there are all sorts of subtleties to consider which can affect the satisfaction of the user with the system. A missed opportunity to help can be fatal in a health-care environment, too much insistence or a reminder in the middle of an important meeting may not be welcomed. The more knowledge the system has of the user and the subtler the understanding of the contexts as well, as the dos and don'ts associated with those different contexts, the most effective the system can be. Clearly, there are interesting tensions between knowledge of the user and privacy but that will not be the focus in this article. The focus will be instead on how these systems know which contexts are important for specific applications, how the system can recognize that it has reached one of contexts of interest and how to react appropriately in each of those. From here onwards, we will group all those areas mentioned at the beginning of this paragraph under the umbrella term "intelligent environments." Not that we think all those areas are the same. Nor do we suggest that IEs are the best representative of the work conducted in all of them. Our choice is purely pragmatic to facilitate reference to those within this article and also because at the intersection of all of them is, precisely, context-awareness.

This article considers the different perspectives of analysis of context and context-awareness both from the AI and the IE communities. These communities have approached the concept from different directions. AI has been traditionally more concerned with ways of representing concepts and their role in commonsense reasoning and in doing so has often interacted with areas like Philosophy, Logic, Linguistics, Psychology and Mathematics. AI systems can be designed following any possible strategy; however, given the interest in the area for capturing how humans solve problems (in some branches, even the focus is mimicking how humans behave) meant that overall AI systems tend to arrive to their concepts, including context, more in a top-down fashion. IEs (and similar areas) have followed quite a different path. Whilst AI has been mostly motivated to larger extent by a philosophical enquiry on human problem solving, IE on the other hand has been more driven by technological developments. Of course, technological advances have also influenced AI and, on the other hand, philosophical concerns have also guided some researchers in IE; however, in the previous sentences, we are referring to a significant difference in emphasis, the dominant concerns in each area.

A group of researchers has been advocating for a decade on the benefits of increasing the use of more AI within the IE-related areas and firmly believes that it is mutually beneficial to increase the understanding these two communities have of each other's work. This advocacy for interaction has been reflected through publications (see e.g., Pollack, 2005; Augusto and Nugent, 2006; Augusto, 2007; Ramos, Augusto and Shapiro, 2008), workshops[1] and tutorials[2,3] at mainstream conferences. This article provides a more focused analysis on one of the many topics which highlights the importance of AI for IE. We provide a state of the art in context-awareness from the specific perspective which allows us to compare how this topic has been explored in both AI and IE and will highlight the synergies and opportunities between these two communities. First, we describe a scenario which represents the typical daily life challenges where context-reasoning provides a valuable support for an IE system trying to provide a service. Then, we will provide surveys on how the notion of context has been addressed both within the more traditional realms of AI and also within the newer areas mentioned at the beginning of this section. We will complement this with a survey on the interaction between the communities of machine learning and context learning.

Contextual scenario

IEs designed to support people on their daily activities are typically referred as AAL systems Augusto et al. (2012). Their focus is on supporting people with specific needs and some of the most popular applications are those to help people with cognitive or physical impairments (e.g., people with symptoms of Alzheimer's or Parkinson's, or people with Down's syndrome or some form of autism) to stay active and live independently. An essential step to achieve these goals is to help the main intended beneficiary of the services to reach places where they can learn, develop a profession, improve their health or socialize (see Figure 1).

Imagine such a person living at home and preparing to go out to the city in the morning to go to work. They encounter challenges in all places, remembering and taking decisions. At home, they need to prepare adequately

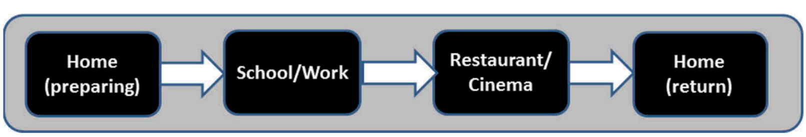

Figure 1. Essential independence support to vulnerable users.

and then they face the challenge of navigating safely through a busy city. The following scenario describes some of the situations where a system can find opportunities to help:

> *Michael is a 60-year-old man who lives alone and enjoys an assistance system that makes his daily life easier. On weekdays, Michael's alarm goes off a few minutes after 08:00 a.m.; approximately 10–15 minutes later, he usually steps into the bathroom. At that moment, the lights are turned on automatically. On Tuesdays, Thursdays and Fridays, he usually takes a shower; Michael prefers the temperature of the water to be around 24–26 degrees Celsius in the winter and around 21–23 degrees Celsius in the summer. Before he leaves the bathroom he turns off the fan and the lights. When he goes into the kitchen the radio turns on so that he can listen to the news while he prepares his breakfast. Before he leaves the house, the system suggests to him that he should wear appropriate clothing because it is raining, and he usually prefers walking even in this situation. He leaves the house 15–20 minutes after having breakfast.*
>
> *Once outside Michael walks towards the bus station where he expects to take a bus at 9AM. Sometimes Michael stays longer in bed, takes longer in the shower or to take breakfast and as a result he arrives at the bus stop after 9AM. The system issues reminders for these activities but Michael sometimes ignores them. Depending on the time when Michael is arriving at the bus stop the system may recommend to wait for the next bus or to take a taxi. If Michael agrees to the suggestion offered then the system provides further guidance otherwise it contacts the carer to help resolve the situation in a safer manner.*

This scenario will be used as a running example showcasing different approaches in the remainder of the article.

Context in KR and reasoning

We start with an analysis of how *context* evolved within knowledge representation (KR) and reasoning. First, we look at how classical AI approached this topic and then how IEs did. We end up highlighting differences, similarities and complementarity between them.

Context in traditional AI

Discussions on "context" can be traced a long time back within philosophy, linguistics and logic; see for example, Frege (1892) and Frege and Gedanke (1918). The notion of context has been present since the initial stages of computing when the study of formal languages (e.g., Chomsky's hierarchy; Chomsky, 1959) was influential in the formation of a theory of Computer Science. These discussions have survived, adapted and specialized to different interests in areas like natural language processing (sentences within a context) and AI (context dependent KR and reasoning). Here, we focus mostly on the debates sparked within this second community of KR and reasoning.

Within the AI community, we can say that the topic was brought into a lively debate by John McCarthy in a series of articles, the most specific of which were in the 1980s and 1990s. Although those discussions considered language-related and problem-solving-related notions of contexts, it is the later one we will focus in this article.

Toward the end of his Turing Award Lecture McCarthy (1987), McCarthy started by motivating the notion of context as

> Whenever we write an axiom, a critic can say that the axiom is true only in a certain context. With a little ingenuity the critic can usually devise a more general context in which the precise form of the axiom does not hold.

This highlights the problem that in areas like IEs where the aim is to provide services to humans in daily life situations, there are few inferences which are always valid. Clearly, we can say that all humans will have to be somehow conceived to exist and they will eventually die. We can state permanent truths in mathematics but daily life is more dynamic, unpredictable and loaded with exceptions. It is difficult to program systems which can take every possibility into account and have a pre-planned specific reaction but we can at least highlight which are some of the situations of interest we can aim to react to appropriately.

The next few articles then pursued an initial formalization of the idea of contexts. Through a sequence of articles (see e.g., McCarthy, 1993; McCarthy and Buvac, 1998), some of them revisions of previous versions, a formal framework is discussed, starting with the notation $ist(c,p)$ to express that a proposition p is true in context c. Then, $ist(c',ist(c,p))$ will mean that *in context* c'*, it is known that "*$ist(c,p)$*" is true*. McCarthy highlights the complexity of describing contexts in general:

> Contexts are abstract objects. We don't offer a definition, but we will offer some examples.… For example, the context associated with a conversation is rich; we cannot list all the common assumptions of the participants. Thus we don't purport to describe such contexts completely; we only say something about them. On the other hand, the contexts associated with certain micro-theories are poor and can be completely described.

The theory includes ways of associating values to contexts, for example, time, and of expressing the relationships amongst contexts to indicate for example that some contents are contained within other contents (specializations). Then, McCarthy explains why in his opinion, $ist(c,p)$ should not be confused with $c \supset p$ in a natural deduction system.

…contexts contain linguistic assumptions as well as declarative and a context may correspond to an infinite and only partially known collection of assumptions.

Other concepts discussed are those of *entering* and *exiting* contexts, *lifting axioms* on contexts (i.e., the process of inferring what is true in one context based on what is true in another context), transferring statements from one context to another and of de-contextualization. These concepts are illustrated with examples of databases integration and plan integration.

Meanwhile, one of McCarthy's students wrote a thesis on the subject (Guha, 1991). Guha's thesis offers a quantified theory of context and revisits many of the concepts listed above when we described McCarthy's related work. Guha was related to the well-known project CYC where the concept of contexts was used in the form of micro-theories. The work by Guha and the work by McCarthy and other colleagues at that time coexisted temporally and informed each other. As a follow up on these developments, Buvac (1996) provided a quantified version (predicate calculus extension). This extension enabled expressing arbitrary first-order properties of contexts as well as expressing that an arbitrary predicate calculus formula is true in a context. Most recently, Bouchard (2017) has revisited McCarthy's and Buvaˇc's ideas through a concept called *epistemic contexts*, supported by a natural deduction inference system, in a system which enables classical reasoning among contexts governed by different concepts of knowledge.

Giunchiglia, a visiting fellow at Stanford, explored another view of context together with other colleagues in Trento. Giunchiglia (1993) proposed a formalization of contexts as multi-view epistemological theory which will form the basis for another branch of analysis on contexts as it was developed by various scholars in Trento. Context is taken as "a subset of the complete state of an individual that is used for reasoning about a given goal." This is contrasted in the paper with the notion of a situation which is taken as "the complete state of the universe at an instant of time." Each context is represented as a logical theory $\langle L, A, R \rangle$, where L is the language of the context, A is a set of axioms and R is a set of inference rules defined over L. This gives place to multiple coexisting first-order theories and overall to a system called *multilanguage system* and the influence of a context c_i over another context c_j is represented through *bridge (inference) rules* which are of the form:

$$\frac{< A_i, c_i >}{< A_j, c_j >} \ (A_i,\ A_j \text{ formulas in contexts } c_i,\ c_j \text{ resp.})$$

Work on contexts as explored by Giunchiglia had in common the tools and logic; however the objectives were slightly different. The vision there was that contexts allow to represent localized reasoning and that common sense reasoning is conducted in such a way only small parts of our knowledge are used for specific inferences and other knowledge allows us to connect these isolated partial inferences. Each context is assumed to have its own associated language and inference engine forming a self-contained logical theory. The novel work is then at the level of treating contexts as complex objects and on the process of connecting the outcomes of those different mini-theories, this is achieved through *bridge (inference) rules*, all together forming *multi-context systems* (MCS).

Giunchiglia and Bouquet addressed the relation between the multi-view approach to contexts and previous work at the time in Giunchiglia and Bouquet (1996). The authors considered two main uses of context within AI up to that time. The one they refer to as pragmatic context considers context *as part of the structure of the world* whilst the one they refer to as cognitive context considers context *as part of the structure of an individual's representation of the world*. The core of the paper revolves around assessing to what extent contexts are needed for modeling reasoning. Their conclusion is that what they called pragmatic contexts can be either subsumed in the notion of cognitive context or does not play any role at all. On the other hand, they argue that cognitive context is needed: it is not true that "any context dependent sentence can be transformed into a sentence whose semantic value is independent of context" given that (a) on one hand, "there are dependencies that cannot be accessed by an agent;" (b) on the other hand, "context-dependence can be so complex and deep that no finite agent can in general have a full knowledge of it." Their theory is summarized through the proposition that *contextuality* = *locality* + *compatibility*. Locality and compatibility are taken here relative to the formalization of contexts given in Giunchiglia (1993); that is, a logical theory $\langle L, A, R \rangle$ as explained before. *Locality* refers to the assumption that each context has its own logic and this allows distinct languages in each of them, so expressivity is local. *Compatibility* means that despite contexts having their own language, they can still allow that the truth of a sentence (or set of sentences) in one of them entails the truth of some other sentences in the second. They refer to knowledge which is perceived to be the same but expressed or referred to in different ways within different contexts.

By the mid-1990s, workshops and conferences started to be created to discuss context specifically related topics. Other developments started to appear in connections with different areas, for example, ontologies. The two surveys from Brézillon (1999a,b), a pioneer in this area, provided an overview of the different views, problems and applications.

Continuing with the Trento line of research, Benerecetti, Bouquet and Ghidini (2000) classify contextual reasoning into three general forms: "localized reasoning," "push and pop" and "shifting." They associate these three to notions to what they believe are three fundamental ways for context-dependent representations: partiality (the portion of the world considered), approximation (the level of detail at which the portion of the world is considered) and perspective (the point of view from which the world is observed). The authors distil two general principles of a logic of contextual reasoning which regulate the relation between models and contexts in the theory.

Let us consider the practical scenario we introduced early on to relate these concepts to practical IE situations. "Localized reasoning" is related to McCarthy's and Giunchiglia's previous developments and refers to the reasoning which is specific to the context being considered; for example, if Michael gets up on Friday, the system knows he should have breakfast in no more than 20 min so a reminder may be useful if he is exceeding that time, and also as he usually takes the bus at 9 AM, again that offers an opportunity to help if Michael is unaware he is getting late. The two mechanisms "push" and "pop" allow a system to take a context as part of the reasoning assumptions, so for example, a smart home should not need to bother explicitly reasoning on whether he is or is not inside the home if the system knows he is inside the bedroom when he is sleeping, this is implied by the structure of the house and by previous information confirming that he indeed is at home (say the house identified when he arrived). So "push" can be metaphorically understood as pushing a concept inside a conceptual box, once there it is accepted and assumed a given context so there is no need to mention it, it is not questioned through reasoning whether that is the case or not, it is accepted as a fact. The "pop" mechanism allows the system to revert that process, say Michael says in loud voice he is going out and the house does not have a way to understand whether he meant he was going to the garden (back door) or to the supermarket (front door). In one case, he may still be considered being "at home" (although not inside the building called house) whilst in the second one, he is not in the house and he is not at home so it is justified for the system to be able to deliberate about this concepts as the context of being home is not trivially obvious any more. "Shifting" refers to changing contextual parameters and reinterpreting a piece of knowledge accordingly, so Michael getting up on Friday can be classified by the system as "getting up on a working day" whilst Michael getting up on Saturday can be classified by the system as "getting up on a weekend day." This will then connect with other areas of knowledge in the system and the system will give priority to different issues which depend on Michael being on a working day or not.

Context is a concept which different areas use and understand in different ways and this problem of a lack of general consensus on what context is has been a long-standing issue within Computer Science and branches from other disciplines closely interacting with Computer Science. After two decades of wrestling with this issue, various researchers have pointed out this problem and Bazire and Brézillon (2005) offer a survey of the various definitions considered, in an attempt to extract from previous literature, different lessons which can help the field to move forward based on a more solid basis.

Brezillion introduced (Brézillon, 2005) a context-based representation formalism for modeling task accomplishment by users by means of so-called contextual graphs:

> A contextual graph is a context-based representation of a task execution. Contextual graphs are oriented without circuits, with exactly one input and one output, and a general structure of spindle. A path (from the input to the output of the graph) represents a practice (or a procedure), a type of execution of the task with the application of selected methods. There are as many paths as practices Different solutions can be associated with the unique output, ….

> Contextual graphs are a formalism of representation allowing the description of decision making in which context influences the line of reasoning (*e.g.*, choice of a method for accomplishing a task).

Brezillion argued that contextual graphs are useful to facilitate the tasks of incremental acquisition, learning and explanation of contexts. This concept has been expanded in several directions; see for example, Brézillon (2017).

The paper by Brewka, Roelofsen and Serafini (2007) provides a multi-context variant of Reiter's default logic in the form of a logic they call contextual default logic. This work was motivated by the observation of consistency problems naturally occurring when more than one observer (e.g., sensor) collect partial information on the same part of reality being monitored. The system includes the use of paraconsistent reasoning to tackle some problems observed on previous systems facing the same challenge. The problems addressed in Brewka, Roelofsen and Serafini (2007) naturally lead to consider that there is a need for better tools within systems of this characteristics to handle consistency. One significant attempt to address this came from what we can call the Leipzig–Vienna line of work, through the so-called MCS (see e.g., Brewka and Eiter, 2007 and Brewka et al., 2011a) by allowing heterogeneous logical formalisms exchange information in a potentially non-monotonic fashion. This framework was generalized later on by Brewka et al. (2011b) into what they called "managed MCS (mMCS)" to allow more flexibility of operations between contexts than those originally allowed by "bridge rules" which only allowed to add information to contexts. In the new system, generalization allows arbitrary operations on context knowledge bases to be freely defined, for example, deletion or revision, operators which can be useful to address the consistency problems mentioned earlier so that instead of just adding new knowledge to the existing one, the new incoming knowledge leads to a revision of the previous one to avoid inconsistencies. The addition of the new operations is encapsulated on the "context manager." A revised version of the mMCS system (Brewka, 2013; Brewka, Ellmauthaler, and Pührer, 2015) focuses on reactive systems and "runs" or streams of data which are continuously flowing. This paper extends previous seminal work by Brewka on mMCSs and incorporates "observations" of streams of data which allows the system to become reactive by continuously matching sensor input to the existing belief sets in each context. The main contribution of the paper to previous reasoning on streams is that it combines a solution to both knowledge integration and online reasoning. In this reactive reasoning framework, the system keeps two different types of bridge rules. One type of bridge rules work in the traditional sense, with knowledge internal to the system, and other bridge rules operate with external data (e.g., coming from sensors). So in Michael's case, there will be bridge rules which reflect Michael's activities in getting up from bed, another group of rules handling the context of having a shower, another set on the context of having breakfast and a separate set of rules which takes input from sensors, for example, to regulate water temperature in the shower or to monitor whether is getting close to 9 AM and warn he may be getting late. Bridge rules can also relate contexts of emergency with time awareness in the system so that if a suspected emergency is detected, then the system can adapt the length of the time window to be considered for reasoning in an analogous fashion to how it is used in stream-based languages like C-SPARQL.[4] Other recent work which spans over several areas is (Halpin, Hayes, and Thompson, 2015) exploring the confluence of context, ontologies and reasoning in the semantic web.

IE approaches

Computational systems are not only becoming smaller but more available for the general public. Tiny electronic devices can be interconnected to work together as part of bigger and more complex systems located in diverse environments that do not necessarily have to be in the classical desktop. Devices can identify or measure a physical input from the world as well as influence physical changes that are tangible to the users. There are different means of sensing and actuating on different physical properties, as summarized in Tables 1 and 2. These provide newcomers to the area with some

Table 1. Sensors transform real world stimuli into digital information.

Real-world stimuli	Sensors
Light (luminosity)	Photoelectrical sensors (LDR)
Light (image)	Cameras
Sound	Microphones
Motion and acceleration	Accelerometers, infrared (active and passive), cameras, radio based, sound based, magnetic
Touch and pressure	Electromagnetic sensors, piezoelectric sensors, piezoresisitive sensors, potentiometric sensors
Location and distance	GPS, cameras, proximity sensors (e.g., RFID, NFC), sonars, radars, infrared thermal sensors, magnetic sensors, electrical sensors
Temperature and humidity	Mechanical sensors (e.g., thermometers) and electrical sensors
Biometrical	Microphones, cameras, fingerprint sensors, eye recognizers (retina, iris), face recognizers
Size	Cameras

Table 2. Actuators transform digital information into real-world stimuli.

Real-world stimuli	Actuators
Movement	Motors and servos (electric, hydraulic, pneumatic, thermal, mechanical)
Visualization	Displays and printers
Sound	Speakers
Electronic	Switches and circuits

examples of the number and diversity of tools available for us to collect data supporting context-aware reasoning.

The access of users to many small different devices with different sensing and actuating capabilities opens up new opportunities of interaction. Weiser (1991) envisioned a future in which devices are anywhere and everywhere, ubiquitously interconnected to offer a seamless experience to the users. His vision materialized in what came to be "Ubicomp" and was largely influential in the later development of areas such as Percomm, IoT, AmI and IE. All these approaches need information of the situation, in order to adapt their services accordingly. Inspired by this demand, Schilit, Adams and Want (1994) first introduced the notion of context-aware computing applications as software that examines and reacts to an individual's changing context. The most acknowledged definition of context which was related to the early Ubicomp area was created few years later by Dey and Abowd (1999), who considered it as "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

In terms of representation, one of the most well-known general purpose context-related ontologies has been SOUPA (Chen et al., 2004). SOUPA was built using a collection of reference ontology vocabularies including FOAF, DAML-Time and the Entry Sub-ontology of Time, OpenCyc, Regional Connection Calculus, COBRA-ONT, MoGATU BDI ontology and the Rei policy ontology. This ontology is broken down into two distinct ontologies; SOUPA Core for generic pervasive applications and SOUPA Extension for specific pervasive domains. The standard in the area is the use of Protégé and Description Logics for reasoning.

Context Modelling Language (CML) (Henricksen and Indulska, 2006) was formulated using concepts from object-role modeling providing a relational database query-based type of framework with a closed world assumption. The representation of tuples has an associated semantics of a three-valued logic (true/possibly true/false). A concept of situations (e.g., when a person is occupied) is created out of lower level contextual information. Situations are handled through a first-order logic with restricted quantification. This concept is supplemented with a system of preferences, triggering situations in an event-condition-action rule fashion (upon-when-do). CML has limitations on the capacity to structure knowledge or to reason with different categories as all contexts are at the same level.

There have been some attempts to define different logic-based systems, for example, the Calculus of Context-Aware (Siewe, Zedan, and Cau, 2011), a logical language for expressing context properties using context expressions. Context expressions can be composed to form more complex expressions and formulas using first-order operators. Many of these have been mostly theoretical explorations which have not gained popularity and are not applied in the construction of practical systems.

Chahuara, Portet and Vacher (2013) present a formal logical model for taking decisions based on the context, which handles the uncertainty of inferring facts from sensor information. They present an approach to represent knowledge based on ontologies and a set of logical rules. For supporting uncertainty, they use a Markov logic network, which makes probabilistic inferences from a model based on weighted logic rules. The authors apply this system to a voice-controlled smart home system.

Providing intelligibility for context-aware applications, allowing for better system understanding by users is nontrivial yet helps improve user trust (Lim and Dey, 2010). An architecture for generating explanations from rules, decision trees, naïve Bayes and hidden Markov models was given. The Intelligibility Toolkit proposed extended the Enactor framework of the Context Toolkit. These added components included a querier, explainer, reducer and presenter. Using either of the four decision models supported by the Intelligibility Toolkit, explanations are generated into disjunctive normal form. These explanations can then be used at runtime to answer questions from the user including *why, why not, what, what if, how to*.

A literature survey of context modeling and reasoning techniques was carried out by Bettini et al. (2010). In that work, discussion on the requirements that context modeling and reasoning should contain was given. For context modeling, it was proposed that these models should consider *heterogeneity and mobility, relationships and dependencies, reasoning* and *usability of modeling formalisms*. By considering heterogeneity contexts can differ in a number of ways including rate of change, the method of data collection, and the type of data they collect. Relationships and dependencies are crucial for allowing different contexts to create higher forms of context also known as compound contexts based on lower level, atomic context data. Reasoning allows the system to determine when a change has taken place allowing for higher level contexts to redetermine their state and determining if a system adaptation is required. By considering usability in modeling, the developer can more easily translate real-world concepts into modeling constructs. High-level context abstractions and uncertainty of context information were two highlighted issues that should be addressed in any modeling framework. Lastly, it was proposed that hybrid context models, those that integrate different models and reasoning, be used.

Other more recent surveys on the notion of context but from slightly different perspectives than those we are considering in this article are Perera et al. (2014) and Alegre, Augusto and Clark (2016).

These are examples of systems which try to capture in their systems a wider complexity of concepts. Other examples of research in the IE area aiming at creating a bottom-up approach which allows for the representation of layers in the system growing in complexity and ambition are based on the definition of more complex contexts based on previously defined ones. Some attempts at addressing this are presented in Gero and Smith (2009) and Ye, Dobson and McKeever (2012) where they borrow the term *situations*, not quite with the same semantics in the well-known situation calculus as used in Lifschitz, V. (Ed.) 1990. Instead, "situations" is more of a catch phrase for contexts which are defined in a hierarchy of increasing complexity.

Contexts can be categorized in several different ways and the categorization selected provides an important link between the reasoning process and the specific application as it interacts with the real world. For example, a categorization which was adopted in the POSEIDON project (Augusto et al., 2013), and suggested as a template for AAL systems, is the categorization of contexts into three broad categories: *user* (e.g., mood, weight and allergies), *environment* (e.g., location and weather) and *system* (e.g., device connectivity level and device battery level). This categorization of contexts is then reflected in an ontology which supports the implementation of the context-reasoning. Contexts in each of these categories can be then classified in *primitive*, that is, they cannot be decomposed, or *composite*, that is, they are made up of other primitive and/or complex contexts, (some authors will also call them in different ways, e.g., "primary" and "secondary," or "simple" and "complex"). The primitive and composite contexts we just mentioned were related to any of the three categories mentioned above (user/environment/system), *intra-category context*. There can still be composite events which are based on more than one contextual category, *inter-category context*. We offer some examples in Table 3, where we assume Michael is waiting for the bus and his mobile phone can give us an image of his face through the camera, the GPS location, the time and the weather forecast through internet services, whilst a wristband sensor provides his heartbeats ratio. Other

Table 3. Examples of different categories of contexts and their combination for reasoning.

Ontological category	Intra-category contexts	Inter-category contexts
User	Primitive: normal heart beats ratio, relaxed face Composite: calm	Safe public transport waiting
Environment	Primitive: sunny, dry Composite: good weather	
System	Primitive: clock time, GPS location Composite: timely at bus stop	

classifications of contexts are used in different systems with different aims; see for example, Lindgren and Nilsson (2013).

AI and IE views compared

Historically, the motivations in both areas were different. In AI, it started more as a philosophical enquiry which had connections to Linguistics, Computer Science and Cognitive Science. One of the important differences is that developments around the notion of context in IE were driven by technology. IE stems from the idea of making machines fit the human environment instead of forcing humans to enter theirs. In order to make systems "disappear" from our daily lives, end users need a more natural interaction with computational systems. This notion of context stems from the need of a more comfortable interaction with technology. If machines would not have the need of being explicitly told what the user wants, but instead would be able to easily get that information from the context, the users could enjoy a richer and more natural human–computer interaction experience. When creating IE systems, developers are mainly interested in the contexts the system can "perceive" and what that can enable a system to do. The system can gather meaningful information data which can contribute to the notion of contexts in a variety of forms: user input (e.g., preferences of the user indicated through an interface), data collected online (e.g., current weather), time (from the machine clock) and learning from past (e.g., that a specific day is special for a given user). However, the recent availability of sensors which can capture a wide range of physical phenomena in real time has triggered the curiosity of developers exploring what type of services they can create based on sensing (and actuation).

On the other hand, AI stems from the idea of creating computational systems that are able to exhibit intelligence, through models that describe the human process of thinking. In this approach, the efforts concern the representation of contexts nested with knowledge Brézillon (1999a). The initial AI analysis on how some deductions are context-dependent, the assumptions which supported a deduction, a statement P is true in the context of assuming the context of $p_1, \ldots, p_n$. Developments in IE use the notion of context as in context-awareness, a system is notified of different facts and the system makes inferences taking those as departing assumptions.

Potential synergies between both approaches

McCarthy's approach and the more recent use of context-awareness as in IE-related areas are different in style. In the initial classic AI approach, the aim was to create a self-contained formalism which included the notion of context within it and the consistency came from using an elegant formal theory with nice meta-theoretical properties.

In the IE approach, systems are designed by programmers with less affinity with the theorem proving background AI researchers would have been familiar in the 1990s. Systems are programmed mainly as a combination of database query languages, Java and AI modules. The AI component may consist of learning components and reasoning components using a variety of techniques the developers are familiar with or borrow as they were produced by others, for example, from repositories like WEKA Hall et al. (2009). In these more recent approaches, systems are too heterogeneous to rest the responsibility of consistency within the system, instead there is a combination of probably internally consistent parts of the system and the programmer holds the responsibility to assess coherence and consistency amongst this collection of modules, that is, the programmer instills the “lifting axioms” using some “gluing language,” most probably Java. There is a wide range of attitudes toward this issue which go from some developing teams using formal model checking tools to increase the possibilities to assess consistency in core parts of the system to those who do not know the meaning of the word consistency.

Say, we expect our IE system to warn us whether it is worth to be better prepared for weather. In today’s approach, it will know the person is at home given the in-house presence sensors, will check the time to leave home from the calendar and the weather from the web. Whilst in McCarthy’s approach, it will require temporal reasoning combined with weather reasoning, facts and inference system about place of the user and a theory about contexts handling, all self-contained.

One of the motivations for the exercise conducted in this article is the hope that in getting the approaches of these two communities better known to each other a middle ground will be reached were “formalists” and “pragmatists” can put together the best of each experiences. With few exceptions, the collaboration landscape can be metaphorically recreated as in Figure 2. Hopefully, after some trial and error with different levels of contribution from each side, the best mixed approaches will survive to support the next generation of systems in this area.

As Euzenat, Pierson and Ramparany (2008) pointed out, the approach originally explored by McCarthy, Guha, Buvac and others at Stanford as well as the one investigated by Giunchiglia, Bouquet, Serafini and others at Trento are of interest to modern sensor-based context handling in the following sense. In the original Stanford-led explorations, each context was considered independent theories which can be related by lifting. Whilst in the Trento-led analysis contexts were partial or approximate views of the same theory. The independent theories can represent each of the independent sensor streams, each stream of data with its own theory to interpret that data. The partial

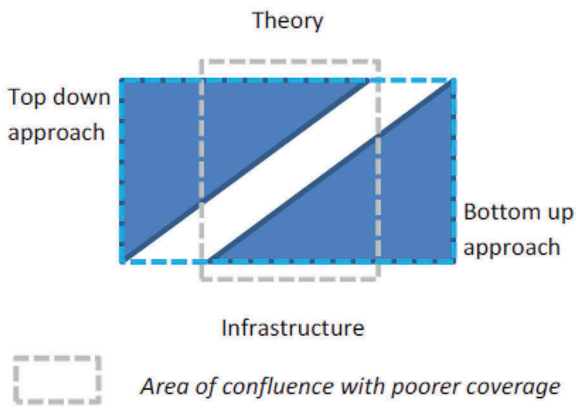

Figure 2. Gap between different approaches to context.

views approach is meaningful to the process of data fusion where some of the sensors provide complementary interpretations (sometimes incomplete, sometimes contradictory with other views) of the same phenomenon being sensed. More recent work on MCS and mMCS led by Brewka is a good example of converging work originated in AI.

Learning for context-awareness

As stated previously, it is already assumed that IEs have to be transparent to the user in all senses. Thus, techniques that allow to extract and learn new knowledge from data have become necessary. Let us consider the scenario showed in Section 'Contextual Scenario' that illustrates an IE that makes the life of the users easier and safer.

One of the hidden and most important assumptions in IEs is that they propose a transition from techno-centered systems to human-centered systems. IEs suppose a change of roles in the relationships between human and technology. Unlike current computing systems where the user has to learn how to use the technology, an IE adapts its behavior to the user, even anticipating his/her needs, preferences or habits.

For that shift to take place, an environment should learn how to react to the actions and needs of the user, and this goal should be achieved in an unobtrusive and transparent way. Due to the complexity of IEs (hardware, software and networks must cooperate in an efficient and effective way to provide a suitable result to the user), initial developments have been focused upon the needs associated with hardware and network as supporting infrastructure. This focus has resulted in a simple automation that implements a reactive environment, which does not take into account the personalized and adaptive features of IEs. There exist sensing systems that are wrongly considered to be intelligent because they act over the user using manually predefined patterns of behavior. In order to provide personalized and adapted services, context awareness is essential, and to create a context awareness which is relevant to people knowing their habits is useful. Thus, the ability to learn patterns of behavior, including the context, becomes an essential aspect for the successful implementation of IEs, because knowing such patterns allows the environment to act intelligently and proactively when it matters. In IEs, learning is mainly focused on supporting the environment to gain knowledge about the preferences, needs and habits of the user, along with context information, in order to better assist the user (Galushka, Patterson, and Rooney, 2006; Leake, Maguitman, and Reichherzer, 2006; Kyriazakos et al., 2016).

The area of learning for context awareness has been acknowledged as an important area (Brézillon, 1999a). It has already attracted a significant number of researchers, and some applications are already being deployed with different degrees of success.

A brief analysis of initial applications developed by different groups shows that current applications are very specific with focused goals, where the context plays a key role in all them. In addition to analyzing the knowledge learned in each application, strong and weak aspects of each ML technique used in the applications are analyzed.

Artificial neural networks

Mozer et al. (1995) and Chan et al. (1995) were amongst the first reports on applications for IEs in which user patterns and context were considered. The aim of the system developed by Mozer et al. and installed in the adaptive house was to design an adaptive control system that considers the lifestyle and energy consumption of the inhabitants. Such an environment was provided with different types of sensors (temperature, light status, illumination and so on) that reported the state of the environmental context. Moreover, the system had the ability to control the status of the lights, the water heater and the gas furnace. Based on this context and using a feed-forward neural network, they developed two applications. The first application, an occupancy predictor, predicted the expected amount of time spent in the home by the inhabitants in the next 30, 60 or 90 min. The second one, a "zone anticipator," predicted whether a particular zone was going to be occupied in the coming 2 s so that the lights were turned on prior to a zone being entered. Chan et al. developed an application in order to assess whether a situation was normal or abnormal. For this application, they assumed an elderly person had fairly repetitive and identifiable habits. Training Artificial Neural Networks (ANNs) with these regular habits, they were able to detect discrepancies to his/her usual behavior. After validating this application in an institution for elderly and disabled people, they claimed that the system had 90% chance of providing correct predictions. Boisvert and Rubio (1999) also used ANNs to develop an intelligent thermostat. Learning about the behavior of the occupants, the objective of this application was to reduce the number of interactions with the user and eliminate the need for users to learn how to program the device. Additionally, the thermostat reduced energy consumption by turning off whenever occupants were absent. Thus, people who have fairly foreseeable behavioral patterns significantly reduced (9–16%) their energy consumption by using a prototype of this thermostat. Campo et al. [Cam06] developed a system that calculated the probability of each area of the home being occupied at a given moment based on continuous observation of the users' habits. See Begg and Hassan (2006) for a survey focused on ANNs for smart homes.

Most of the authors who have used ANNs for the learning process highlight their ability to generalize as well as their robustness when faced with complex data (e.g., noisy or missing values). In order to clarify the strengths and weaknesses of ANNs, Michael's scenarios will be used as an example. Due to the capacity of ANNs to manage complex data and create complex models, a system based on ANNs will provide correct responses in situations such as turning on the lights when Michael goes into the bathroom or getting the shower ready on Tuesdays, Thursdays and Fridays. There are already systems (see applications mentioned above) that use ANNs to predict the presence of the user or the occurrence of an action. In that sense, ANNs are one of the techniques that better accommodate the complexity (type of data, data inconsistency etc.) of IEs. However, ANNs have an important limitation related to their black box nature; their internal structure is not human readable. Thus, the system would be able to turn on the light, but it would not be able to explain, in a comprehensible way, how it inferred such an output. If understanding users frequent behaviors is considered as essential, ANNs face an insuperable difficulty.

Classification techniques

The group that works on the environment named 'SmartOffice' (Le Gal et al., 2001) was the first to identify the use of rules in order to recognize working conditions contexts and act proactively. SmartOffice was composed of 50 context-related sensors (cameras and microphones) and 3 context-related actuators (a video projector and 2 speakers). Given these sensors and actuators, the researchers used a set of predefined rules to integrate different components into a coherent application. One of the main reasons rules were used in this application was because they allowed the addition, deletion or modification of rules without influencing other rules. Thus, they guaranteed scalability of the system. The SmartOffice group continued to use classification techniques in IEs. In order to justify the use of classification techniques, they pointed out that "a user is only willing to accept an intelligent environment offering services implicitly if he understands and foresees its decisions" (Brdiczka, Reignier, and Crowley, 2005) and the context plays a key role in this sense. Taking as a starting point a pre-defined context model, they identified situations where examples indicated different reactions for such situations. Thus, it was necessary to define under what conditions a reaction would or would not take place. With the knowledge that decision trees were able to perform classifications, they experimented with FIND-S, Candidate Elimination and ID3 methods, finding the last to be the best.

Stankovski and Trnkoczy (2006) also analyzed the possibility of using decision trees in smart homes. The application they proposed was the detection of abnormal situations by means of decision trees. Based on the assumption that events that usually happened in a smart home may be considered normal events, they induced a decision tree. Then, each new situation was analyzed and the decision tree determined whether it was abnormal or not.

One of the main advantages of these classification techniques for IEs is the way they represent knowledge. Due to their human-readable representation, extracted knowledge can be used by a third party to understand a user's behavior, as well as to explain to the user the decisions made by the system, where the decisions made in different nodes are related to context. As mentioned in one of the applications, classification techniques can be very useful for discovering conditions where certain actions follow other specific actions. For example, in Michael's case, the environment would realize that sometimes he has a shower and sometimes he does not. Using classification techniques, the environment would be able to discover what days he does and when he does not. The advantages of representing a user's behavior by means of rules are clear. Even so, a single rule does not give any sense of sequence to the actions, so something else is required to discover and represent a user's behaviors by means of sequences.

Fuzzy logic

Researchers at Essex's iDorm lab focused on the problem of learning and were one of the most active groups in this area (Hagras et al., 2004; Doctor, Hagras, and Callaghan, 2005). Their objective was to develop learning and adaptation techniques for embedded agents. To that end, they developed a test bed, iDorm (later on iDorm2, iSpace and iSpace2), where 7 input sensors were monitoring the activities of daily living context (e.g., internal/external light level or bed pressure) and 10 output actuators were controlled (e.g., desk and bed side lamps or window blinds).

Their initial efforts were focused on developing an unsupervised approach for extracting fuzzy rules and membership functions from data to develop a fuzzy controller that would model the user's behaviors based on previous actions and contextual information. The data were collected by monitoring the user in the environment over a period of time. The learned controller provided an inference mechanism that produced output control responses based on the current state of the inputs. They defined a five phases approach to create a fuzzy controller:

- Monitoring the user and the context, capturing input/output data.
- Extraction of the fuzzy membership functions from the data. To achieve this extraction, they used a double-clustering approach (Castellano, Fanelli, and Mencar, 2002), combining fuzzy-C-means and hierarchical clustering.
- Extraction of fuzzy rules from the recorded data. The extraction approach used was based on an enhanced version of the Mendel–Wang method (Wang and Mendel, 1992) developed by Wang (2003).
- Control of the environment by the agent controller environment on behalf of the human according to his/her desires.
- Adaptation mechanism. Whenever the user was dissatisfied with the agent's actions, he/she could always override the agent's control responses by simply altering the manual control of the system. When this occurred, the agent adapted its rules online or added new rules based on the new user preferences.

Vainio, Valtonen and Vanhala (2008) also used fuzzy rules to represent habits of a user. In contrast to the approach followed in the iDorm project, these authors manually constructed the membership functions and used reinforcement learning to replace old rules in order to prevent single overriding events from having too large an impact.

The nature of rules generated in this way will be similar to those rules obtained using the classification techniques described in the previous section. They are considered more robust when dealing with context data of a continuous nature (e.g., temperature, humidity and time). In Michael's case, for those actions performed when the global situation was similar (e.g., by taking a shower on Tuesdays, Thursdays and Fridays), the controller would provide a correct output. Due to the multiplicity of sensors and the number of different situations that can be generated when combining sensors, it seems clear that relating actions only to global conditions (without relating actions to other actions) will result in an excessive number of generated rules with very little meaning. In Michael's case, it is clear that the action of turning on the lights in the bathroom is typically associated with the action of going into the bathroom. Thus, it is essential to discover frequent relations between actions.

Association techniques

The group working on the MavHome and Casas projects is one of the most active groups in this field of research (Sprint, Cook, and Schmitter-Edgecombe, 2016). The first applications developed by this group were focused on building universal models, represented by Markov models, to predict future locations or activities (Cook and Das, 2007). The researchers made notable improvements by developing applications to discover daily and weekly patterns (Heierman and Cook, 2003). Additionally, they constructed an application with the ability to infer abstract tasks automatically and identify corresponding activities that were likely to be part of the same task (Rao and Cook, 2004).

However, the major contributions of this research group have been their research on discovering frequent relations between events which inform the recognition of human behavior (Jakkula, Crandall, and Cook, 2007). After collecting context data, they first identified temporal relations that occurred among events, and they then applied association rule mining techniques to focus on the event sequences and temporal relations that frequently occurred. They used the temporal relations between events as a basis for reasoning to perform anomaly detection and prediction of events. In order to define temporal relations, they used Allen's temporal logic (Allen, 1984), which produced fairly intuitive sequences of actions.

Once their new approach was developed, they tested it using a dataset collected from the MavLab smart workplace (Youngblood, Cook, and Holder, 2005), which contained 2 months of data. Additionally, they generated a synthetic dataset containing about 4000 events representing 2 months of activities.

The knowledge discovered by associating actions and activities can easily be represented in a comprehensible way. Moreover, relating such events temporally provides a sequential representation that also facilitates including context data. In Michael's case, the system would be able to detect that he first gets up, then goes into the bathroom and then turns on the light. As stated previously, this representation produces intuitive sequences of actions, allowing the system to detect anomalies as well as to predict future events. Although this is one of the most promising approaches, a few aspects that need improvement can be noted. First, this system does not determine that a group of activities is part of the same sequence but rather detects relations separately. Second, this system only considers Allen's temporal logic relations (which define relations qualitatively), thereby ruling out quantitative relations. Thus, the term "after" means that Michael goes into the bathroom and then he turns on the lights; however, the likely delay between one action and the next cannot be measured. Defining relations by means of quantitative values allows the system to automate actions, which is impossible with purely qualitative values (e.g., the system knows that turning the lights on comes after a given event, but it does not know if the time delay is 2 s, 5 min or 2 h after the first event).

Instance-based learning

The MyCampus group at Carnegie Mellon University (Sadeh, Gandon, and Kwon, 2006) developed some interesting applications for IEs using case-based reasoning (CBR). Their main objective was to provide a set of services to enhance everyday campus life. Thus, applications for recommending services (e.g., where to eat or public transportation) or for reminding users about tasks were developed. One of the most interesting services was a message filtering service, which allowed a user to specify preferences as to when he/she wanted to see different types of messages based on the nature of the message (i.e., subject and sender) and context. In addition, users could provide feedback to help the system refine the preferences they originally entered.

In the first iteration, users had to specify their message filtering preferences (a priori preference) for different categories of messages. Seeing the poor results obtained by using a priori preferences, the group implemented a CBR module, which attempted to learn preferences for individual users based on their feedback.

Apart from the MyCampus project, some other researchers have also used CBR to acquire knowledge about users. Kushwaha et al. (2004) proposed an intelligent agent for Ubicomp environments (UT-AGENT), which had the objective of determining users' information requirements and helping them by providing a task of interest. They stored the user's behavior as cases, and new queries were classified according to its similarity with previous recorded queries. In this case, context information was used for measuring the similarity.

Considering the use of instance-based learning (IBL) techniques in Michael's scenario, their strengths and weaknesses will be clarified. Given a situation similar to one stored previously, the system would act properly because IBL techniques provide similar solutions to similar problems/situations without any initial model. Thus, when the system detects it is raining, and considering previous similar situations, it suggests Michael to wear appropriate clothing.

However, the use of IBL techniques has some limitations. As this process infers a solution for each specific situation, it does not create a model that represents patterns. Therefore, it would not be possible to extract a general pattern indicating the behavior of Michael to turn on the lights after going into the bathroom. Further, as each situation can be represented by means of a large number of parameters, the matching process could be very difficult because there are no clues regarding the importance of each parameter in each situation. Considering Michael's habit of having a shower, if we consider the parameter day of the week, it seems clear when he takes a shower and when he does not. However, other parameters (e.g., light level or temperature) that would shape the pattern differently could also be considered, making the process of matching difficult.

Reinforcement learning

As seen previously, Mozer et al. developed a system that predicted whether a zone in the house would be occupied. In addition to this system, these researchers developed other methodologies, using the Q learning algorithm (Watkins and Dayan, 1992) for lighting regulation. The system controlled the status of the lights (on/off) and their intensity. Starting with the assumption that the inhabitant had no preferences for the device setting, the system tried to minimize energy consumption as long as the inhabitant did not express discomfort. Once the system received feedback from the user, it tried to balance user's preferences with energy consumption.

The SmartOffice group has also used reinforcement learning in their research work (Zaidenberg, Reignier, and Crowley, 2009). Their main objective was to construct automatically a context model by applying reinforcement techniques, where the user gave rewards by expressing his/her satisfaction with the system actions.

In Michael's example, if we consider that the system already has a model (either defined manually or learned by means of previously mentioned techniques), reinforcement learning techniques can be used in order to adapt such patterns. Let us hypothesize that learned patterns define that the shower must be ready every weekday. Every time Michael does not have a shower would be a penalty for the system; that is, it would be considered as negative feedback. After collecting feedback, reinforcement learning would change the pattern and adapt it to Michael's new preferences, that is, to have the shower ready only on Tuesdays, Thursdays and Fridays.

Still, the use of this technique demands a set of initial patterns that ideally should be learned automatically instead of from pre-defined models (which could annoy users and even make difficult the process of learning habits without any bias). Although other techniques have the same limitation, the inherent difficulty in reinforcement learning is interpreting user's feedbacks; this is particularly important for reinforcement learning because this system is based mainly on the interpretation of this feedback.

Technique combination and holistic approaches

As seen in above, context-related information is used in all cases, sometimes in order to predict the status of the context itself and some other times in order to help predicting user's actions. Each technique has its own strengths and weaknesses, but it is difficult to design a holistic learning system using only one technique. Thus, many researchers combined several techniques, machine learning techniques among them as well as with other techniques, in order to develop holistic approach for context awareness.

Brdiczka, Reignier and Crowley (2005) combined 3D video tracking system together with head set microphones. The 3D tracking system created and tracked people in the scene, and the role of each person is derived from the extracted properties of the 3D tracker. The speech activity detector analyzes audio streams and determines for each person whether the person speaks or not. Then, using Hidden Markov Models, different situations are learned and detected in order to analyze human behaviors and further detection of these patterns.

Aztiria et al. (2013) also combined different learning techniques in order to learn frequent behavioral patterns of the users. Association, classification and clustering techniques were used in the learning process.

Classification and clustering techniques were combined by Li et al. (2013) in order to improve user experience by mining user preferences from the user's past context. To cope with the high dimensionality and heterogeneity of context data, they used a subspace clustering approach that is able to find user preferences identified by different feature sets.

Gjoreski (2015) developed a domain-independent approach consisting of three steps: context extraction, context modeling and context aggregation. Processing comes after partition of data: it uses the understanding of the nature of the training datasets to select more meaningful perspectives of the data each model is related to. This is complemented with an aggregation process of the context models to obtain a more robust generalization. The main difference in this approach is that multiple reasoning models are created using different contexts (Gjoreski, Gams, and Lustrek, 2014). Each classifier is trained on a subset of the training set that is more homogeneous than the whole set and used in the context of this subset. For example, the model constructed for the activity sitting uses only the data instances that contain that activity.

Conclusions

We have explained how the concept of context and context-reasoning has been evolving in two different communities, namely AI and IEs. Each of these communities approached the concept differently because their agendas have different priorities so they explored and emphasized different aspects of contextual reasoning.

On the use of context for KR and reasoning, we found that AI tried to create a theory of context within the robust KR available at the time.

However, the IE community was more strongly led by experiments on how to get systems running on resource scarce computing devices to appropriately react to specific real-world-related conditions. On the machine learning approach, context learning was more uniform in the sense that it consisted of applying more or less the same tools to different problems. Some approaches in IE try to learn the contexts where context-reasoning is worth applying and also to learn how in different contexts, different learning approaches can be more advantageously applied. Figure 3 provides a graphical view of some of the main developments and landmarks in these areas.

Although we found initially that AI and IE considered this topic quite independently and separated from each other, latest developments have started to show signs of interaction amongst these areas. Some AI work addressing more formal theories using sensor data started to emerge; however, they are not yet massively adopted. This will require time, dissemination for the IE community to understand and put to the test to see how it works at a practical level. On the other hand, IE has a wealth of experience on deployed systems in a variety of domains but has not yet converged on a systematic approach and probably can benefit from adapting more often existing extensively researched AI techniques rather than inventing new ones.

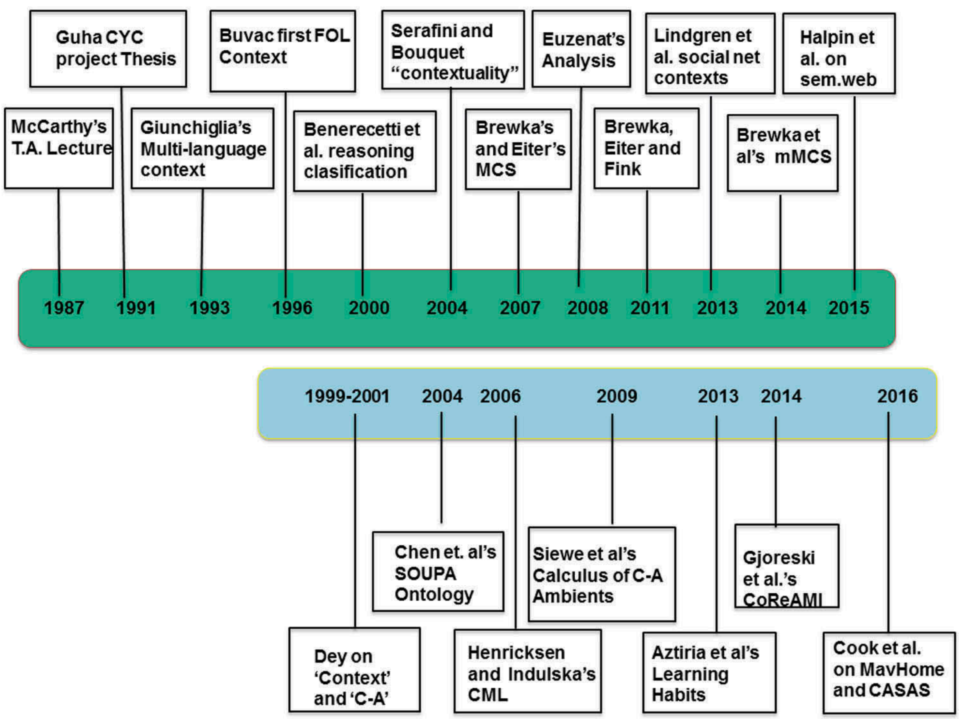

Figure 3. Timeline including some landmarks in context awareness both from AI and IE perspectives.

The aim of our survey was to provide an understanding of the way context and context-reasoning were complementary approached in each of those areas and to highlight the scope for positive interactions arising amongst them. We hope this article informs and encourages further and longer term synergies between those interested in further understanding and using context-based reasoning.

Notes

1. http://aitami2015.mondragon.edu/aitami15.
2. www.eis.mdx.ac.uk/staffpages/juanaugusto/tutorial_program.pdf.
3. www.ijcai-07.org/tutorialdesc.php#t18.
4. https://www.w3.org/community/rsp/wiki/RDF_Stream_Processors_Implementation.

References

Aarts, E. H. L., and R. Roovers. 2003. IC design challenges for ambient intelligence. 2003 Design, Automation and Test in Europe Conference and Exposition (DATE 2003), Munich, Germany, 3-7 March 2003, 10002–07.

Alegre, U., J. C. Augusto, and T. Clark. 2016. Engineering context-aware systems and applications: A survey. *Journal of Systems and Software* 117:55–83.

Allen, J. F. 1984. Towards a general theory of action and time. *Artificial Intelligence* 23 (2):123–54.

Atzori, L., A. Iera, and G. Morabito. 2010. The internet of things: A survey. *Computation Network* 54 (15, October), pp. 2787–2805.

Augusto, J. C. 2007. Ambient intelligence: The confluence of ubiquitous/pervasive computing and artificial intelligence. In *Intelligent computing everywhere*, ed. A. Schuster, 213–34. Springer Verlag.

Augusto, J. C., V. Callaghan, D. Cook, A. Kameas, and I. Satoh. 2013a. Intelligent environments: A manifesto. *Human-Centric Computing and Information Sciences* 3:12.

Augusto, J. C., T. Grimstad, R. Wichert, E. Schulze, A. Braun, G. M. Rodevand, and V. Ridley. 2013b. Personalized smart environments to increase inclusion of people with down's syndrome. Proceedings of 4th Int. Joint Conf. on Ambient Intelligence, Dublin, Eire, 223–28.

Augusto, J. C., M. Huch, A. Kameas, J. Maitland, P. McCullagh, J. Roberts, A. Sixsmith, and R. Wichert. 2012. *Handbook on ambient assisted living*. Springer, Berlin: IOS Press.

Augusto, J. C., and C. D. Nugent. 2006. Smart homes can be smarter. In *Designing smart homes. The role of artificial intelligence*, 1–15, Springer, Berlin.

Aztiria, A., J. C. Augusto, R. Basagoiti, A. Izaguirre, and D. Cook. 2013. Learning frequent behaviours of the user in intelligent environments. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 43 (6):1265–78.

Bazire, M., and P. Brézillon. 2005. Understanding context before using it. Modeling and Using Context, 5th International and Interdisciplinary Conference, CONTEXT 2005, Paris, France, July 5-8, 2005, Proceedings, 29–40.

Begg, R., and R. Hassan. 2006. Artificial neural networks in smart homes. In *Designing smart homes*, 146–64. Springer, Berlin.

Benerecetti, M., P. Bouquet, and C. Ghidini. 2000. Contextual reasoning distilled. *Journal of Experimental & Theoretical Artificial Intelligence* 12 (3):279–305.

Bettini, C., O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, and D. Riboni. 2010, April. A survey of context modelling and reasoning techniques. *Pervasive and Mobile Computing* 6 (2), 161–80.

Boisvert, A., and R. G. Rubio. 1999. Architecture for intelligent thermostats that learn from occupants' behavior. *ASHRAE Transactions* 105:124–30.

Bouchard, Y. 2017. Reasoning in epistemic contexts. Modeling and Using Context - 10th International and Interdisciplinary Conference, CONTEXT 2017, Paris, France, June 20-23, 2017, Proceedings, 99–112.

Brdiczka, O., P. Reignier, and J. L. Crowley. 2005. Supervised learning of an abstract context model for an intelligent environment. Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: Innovative context-aware services: Usages and technologies, ACM, 259–64.

Brewka, G. 2013. Towards reactive multi-context systems. Logic Programming and Nonmonotonic Reasoning - 12th International Conference, LPNMR 2013, Corunna, Spain 1–10.

Brewka, G., and T. Eiter. 2007. Equilibria in heterogeneous nonmonotonic multi-context systems. Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, July 22-26, 2007, 385–90.

Brewka, G., T. Eiter, and M. Fink. 2011a. Nonmonotonic multi-context systems: A flexible approach for integrating heterogeneous knowledge sources. Logic Programming, Knowledge Representation, and Nonmonotonic Reasoning - Essays Dedicated to Michael Gelfond on the Occasion of His 65th Birthday, eds. M. Balduccini, and T. Son, 233–58, Springer, Berlin.

Brewka, G., T. Eiter, M. Fink, and A. Weinzierl. 2011b. Managed multi-context systems. In Proceedings of the 22nd IJCAI, Barcelona, Spain, 786–91.

Brewka, G., S. Ellmauthaler, and J. Pührer. 2015, May. Multi-context systems for reactive reasoning in dynamic environments. Proceedings of the Int. Workshop on Reactive Concepts in Knowledge Representation (ReactKnow 2014), Prague, Czech Republic, 23–29.

Brewka, G., F. Roelofsen, and L. Serafini. 2007. Contextual default reasoning. Proceedings of IJCAI, Hyderabad, India, 268–73.

Brézillon, P. 1999a. Context in artificial intelligence: I. A survey of the literature. *Computers and Artificial Intelligence*, Morgan Kaufmann, San Francisco, USA, 18 (4):321–40.

Brézillon, P. 1999b. Context in artificial intelligence: II. Key elements of contexts. *Computers and Artificial Intelligence* 18 (5):425–46.

Brézillon, P. 2005. *Task-realization models in contextual graphs*, In Proceedings of 5th International and Interdisciplinary Conference on Modeling and Using Context, CONTEXT 2005, Paris, France 55–68.

Brézillon, P. (2017). Contextual modeling of group activity. In Modeling and Using Context - 10th International and Interdisciplinary Conference, CONTEXT 2017, Paris, France, June 20-23, 2017, Proceedings, 113–26.

Buvac, S. 1996. Quantificational logic of context. *AAAI/IAAI 1*, 600–06.

Castellano, G., A. M. Fanelli, and C. Mencar. 2002. Generation of interpretable fuzzy granules by a double-clustering technique. *Archives of Control Science* 12 (4):397–410.

Chahuara, P., F. Portet, and M. Vacher. 2013. Making context aware decision from uncertain information in a smart home: A markov logic network approach. In *Ambient intelligence*, 78–93. Dublin, Eire: Springer.

Chan, M., C. Hariton, P. Ringeard, and E. Campo. 1995. Smart house automation system for the elderly and the disabled. In Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on, Volume 2, IEEE, 1586–89.

Chen, H., F. Perich, T. Finin, and A. Joshi. 2004, August. SOUPA: Standard ontology for ubiquitous and pervasive applications. In Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004. The First Annual International Conference on, Cambridge, MA, USA, 258–67.

Chomsky, N. 1959. On certain formal properties of grammars. *Information and Control* 9:137–67.

Cook, D. J., J. C. Augusto, and V. R. Jakkula. 2009. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* 5 (4):277–98.

Cook, D. J., and S. K. Das. 2007. How smart are our environments? An updated look at the state of the art. *Pervasive and Mobile Computing* 3 (2):53–73.

Dey, A. K., and G. D. Abowd. 1999. Towards a better understanding of context and context-awareness. *Computing Systems* 40 (3):304–07.

Doctor, F., H. Hagras, and V. Callaghan. 2005. A fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions On* 35 (1):55–65.

Euzenat, J., J. Pierson, and F. Ramparany. 2008. Dynamic context management for pervasive applications. *The Knowledge Engineering Review* 23 (01):21–49.

Frege, G. 1892. On sense and meaning. In *Philosophy of language*, 1985, Oxford University Press.

Frege, G. 1918. Der Gedanke: Eine Logische Untersuchung. Eine Logische Untersuchung. In *Beitrage zur Philosophic des Deutschen Idealismus*, 1.

Galushka, M., D. Patterson, and N. Rooney. 2006. Temporal data mining for smart homes. In *Designing smart homes. The role of artificial intelligence*, 85–108. Springer, Berlin.

Gero, J. S., and G. J. Smith. 2009. Context, situations, and design agents. *Knowl.-Based Systems* 22 (8):600–09.

Giunchiglia, F. 1993. Contextual reasoning. *Epistemologia, Special Issue on I Linguaggi E Le Macchine*, vol. XVI,345–64.

Giunchiglia, F., and P. Bouquet. 1996. Intoduction to contexual reasoning: An artificial intelligence perspective. *Perspectives on Cognitive Science* 3:138–59.

Gjoreski, H. 2015. An approach for context-based reasoning in ambient intelligence. *Informatica (Slovenia)* 39 (1):100–101.

Gjoreski, H., M. Gams, and M. Lustrek. 2014. Context-based fall detection and activity recognition using inertial and location sensors. *Jaise* 6 (4):419–33.

Guha, R. V. 1991. Contexts: A formalization and some applications. Ph. D. thesis, Stanford University.

Hagras, H., V. Callaghan, M. Colley, G. Clarke, A. Pounds-Cornish, and H. Duman. 2004. Creating an ambient-intelligence environment using embedded agents. *Intelligent Systems, IEEE* 19 (6):12–20.

Hall, M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. 2009, November. The weka data mining software: An update. *SIGKDD Explor Newsletter* 11 (1), 10–18.

Halpin, H., P. J. Hayes, and H. S. Thompson (2015). When owl: Sameas isn't the same redux: Towards a theory of identity, context, and inference on the semantic web. Modeling and Using Context - 9th International and Interdisciplinary Conference, CONTEXT 2015, Lanarca, Cyprus, November 2-6, 2015. Proceedings, 47–60.

Heierman, E. O., III, and D. J. Cook. 2003. Improving home automation by discovering regularly occurring device usage patterns. Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, IEEE, Florida, USA, 537–40.

Henricksen, K., and J. Indulska. 2006, February. Developing context-aware pervasive computing applications: Models and approach. *Pervasive and Mobile Computation* 2 (1), 37–64.

Jakkula, V. R., A. S. Crandall, and D. J. Cook. 2007. Knowledge discovery in entity based smart environment resident data using temporal relation based data mining. Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on, IEEE, 625–30.

Kushwaha, N., M. Kim, D. Y. Kim, and W.-D. Cho. 2004. An intelligent agent for ubiquitous computing environments: Smart home UT-AGENT. Software Technologies for Future Embedded and Ubiquitous Systems, 2004. Proceedings. Second IEEE Workshop on, IEEE, 157–59.

Kyriazakos, S., M. Mihaylov, B. Anggorojati, A. Mihovska, R. Craciunescu, O. Fratu, and R. Prasad. 2016. eWALL: An intelligent caring home environment offering personalized context-aware applications based on advanced sensing. *Wireless Personal Communications* 87:1093–111.

Le Gal, C., J. Martin, A. Lux, and J. L. Crowley. 2001. Smart office: Design of an intelligent environment. *IEEE Intelligent Systems* 16 (4):60–66.

Leake, D., A. Maguitman, and T. Reichherzer. 2006. Cases, context, and comfort: Opportunities for case-based reasoning in smart homes. In *Designing smart homes. The role of artificial intelligence*, 109–31. Springer, Berlin.

Li, F., K. Rasch, S. Sehic, S. Dustdar, and R. Ayani. 2013. Unsupervised context-aware user preference mining. Workshop on Activity Context-aware Systems at Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13), Bellevue, Washington, USA.

Lifschitz, V., Ed. 1990. *Formalizing common sense: Papers by John McCarthy*. Ablex Publishing Corporation, New York, USA.

Lim, B., and A. Dey. 2010. Toolkit to support intelligibility in context-aware applications. Proceedings of the 12th ACM Int. Conf. on Ubiquitous Computing, Copenhagen, Denmark, 13–22.

Lindgren, H., and I. Nilsson. 2013. Towards user-authored agent dialogues for assessment in personalised ambient assisted living. *International Journal Web Engineering Technological* 8 (2):154–76.

McCarthy, J. 1987. Generality in artificial intelligence. *Communications of the ACM* 30 (12):1030–35.

McCarthy, J. 1993. Notes on formalizing context. Proceedings of the Thirteenth IJCAI, Chambéry, France.

McCarthy, J., and S. Buvac. 1998, October. Formalizing context (Expanded notes). Technical report. Stanford University, Stanford, USA.

Mozer, M. C., R. H. Dodier, M. Anderson, L. Vidmar, R. F. Cruickshank III, and D. Miller. 1995. The neural network house: An overview. In Current trends in connectionism, eds. L. Niklasson, and M. Boden. pp. 371–380. Hillsdale, NJ: Erlbaum.

Perera, C., A. Zaslavsky, P. Christen, and D. Georgakopoulos. 2014. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys and Tutorials* 16 (1):414–54.

Pollack, M. E. 2005. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI Magazine* 26 (2):9–24.

Ramos, C., J. C. Augusto, and D. Shapiro. 2008. Ambient intelligence: The next step for artificial intelligence. *IEEE Intelligent Systems* 23:15–18.

Rao, S. P., and D. J. Cook. 2004. Predicting inhabitant action using action and task models with application to smart homes. *International Journal on Artificial Intelligence Tools* 13 (01):81–99.

Sadeh, N., F. Gandon, and O. B. Kwon. 2006. *Ambient intelligence: The mycampus experience*.

Schilit, B. N., N. Adams, and R. Want. 1994. Context-aware computing applications. Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA 85–90.

Siewe, F., H. Zedan, and A. Cau. 2011, July. The calculus of context-aware ambients. *Journal of Computer and System Sciences* 77 (4), 597–620.

Sprint, G., D. J. Cook, and M. Schmitter-Edgecombe. 2016. Unsupervised detection and analysis of changes in everyday physical activity data. *Journal of Biomedical Informatics* 63:54–65.

Stankovski, V., and J. Trnkoczy. 2006. Application of decision trees to smart homes. In *Designing smart homes. The role of artificial intelligence*, 132–45. Springer, Berlin.

Vainio, A.-M., M. Valtonen, and J. Vanhala. 2008. Proactive fuzzy control and adaptation methods for smart homes. *Intelligent Systems, IEEE* 23 (2):42–49.

Wang, L.-X. 2003. The WM method completed: A flexible fuzzy system approach to data mining. *Fuzzy Systems, IEEE Transactions On* 11 (6):768–82.

Wang, L.-X., and J. M. Mendel. 1992. Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, IEEE Transactions On* 22 (6):1414–27.

Watkins, C. J. C. H., and P. Dayan. 1992. Q-learning. *Machine Learning* 8 (3–4):279–92.

Weiser, M. 1991. The computer for the 21st century. *Scientific American* 265:94–104.

Ye, J., S. Dobson, and S. McKeever. 2012, February. Situation identification techniques in pervasive computing: A review. *Pervasive and Mobile Computing* 8 (1), 36–66.

Youngblood, G. M., D. J. Cook, and L. B. Holder. 2005. Managing adaptive versatile environments. *Pervasive and Mobile Computing* 1 (4):373–403.

Zaidenberg, S., P. Reignier, and J. L. Crowley. 2009. Reinforcement learning of context models for a ubiquitous personal assistant. 3rd Symposium of Ubiquitous Computing and Ambient Intelligence 2008, Springer, Salamanca, Spain, 254–64.