

Article

# Cognitive Assistants—An Analysis and Future Trends Based on Speculative Default Reasoning

João Ramos <sup>1,\*</sup> , Tiago Oliveira <sup>2</sup>, Ken Satoh <sup>2</sup>, José Neves <sup>1</sup> and Paulo Novais <sup>1</sup>

<sup>1</sup> Algoritmi Centre, Department of Informatics, University of Minho, 4710 Braga, Portugal; jneves@di.uminho.pt (J.N.); pjon@di.uminho.pt (P.N.)

<sup>2</sup> National Institute of Informatics, Hitotsubashi, Chyoda-ku, Tokyo 101-8430, Japan; toliveira@nii.ac.jp (T.O.); ksato@nii.ac.jp (K.S.)

\* Correspondence: jramos@di.uminho.pt; Tel.: +351-253604430

Received: 28 February 2018; Accepted: 4 May 2018; Published: 8 May 2018



**Abstract:** Once a person is diagnosed (or a caregiver suspects that the person may have) cognitive disabilities he may lose the state of being autonomous, which may range from partial to total loss of independence, according to the level of incidence. Smart houses may be used as a tentative solution to overcome this situation. However, when one goes outside their premises, this alternative may become unusable. Indeed, due to the decreased orientation ability, caregivers may prevent these people from going out, as they may get lost. Therefore, we are developing a system for people with mild or moderate cognitive disabilities that guides the user through an augmented reality interface and provides a localization tool for caregivers. The orientation method implements a speculative computation module, thus the system may calculate and anticipate possible user mistakes and issue alerts before he takes the wrong path. Through a trajectory mining module, the path is also adjusted to user preferences. These two modules enable the system to adapt to the user.

**Keywords:** ambient intelligence; cognitive disabilities; mobile communication; orientation; person tracking; trajectory mining

## 1. Introduction

Cognitive disability is, according to the Diagnostic and Statistical Manual for Mental Disorders (DSM-IV) [1], a medical condition associated to an individual who has more difficulties in one or more types of mental tasks. These tasks include self-care, communication, orientation, use of community resources, functional academic skills, work, leisure, health, and safety, among others. The diagnostic of a cognitive disability is not an easy task since it may be present in different forms, like stroke, Alzheimers, traumatic brain injury, and even with different levels of incidence (varying from mild to extreme). The main criteria or the most frequently observed is the intellectual function, which is compromised in the presence of cognitive disabilities. A person with this disability has more difficulties in accomplishing one or more types of mental tasks [2].

When a person is submitted for a diagnosis, physicians may provide two different types of diagnosis, the functional and/or the clinical. The clinical diagnosis is related to the technical name, e.g., Down's syndrome, cerebral palsy or autism, and usually coexists with the functional one. The functional diagnostic is more concerned with the deficits that the person may face, namely memorization deficit, attention deficit, and mathematical comprehension deficit [3]. The categorization of the disabilities into a functional or clinical perspective may create a distance from the medical point of view. Indeed, the functional diagnostic may ignore the causes of the disease and focus on the resulting challenges due to the capacity decreasing.

Cognitive disabilities are usually divided into four stages, namely mild, moderate, severe and extreme. The majority of diagnostics are for mild cognitive disabilities (about 80%), while about 14% are related to moderate to severe ones [4]. According to the Diagnostic and Statistical Manual of Mental Disorders 5 (DSM-V) [5] the degree of incidence is less dependent on the Intelligence Quotient (IQ) scores and has more emphasis on the amount and type of intervention needed. Despite considering the IQ scores for assessing the level of disability, mental health professionals must consider the ability or impairment of the person in three areas, namely conceptual, social, and practical life skill.

An adopted solution by physicians may consist of the use of drugs with the goal of slowing the progression of the disease, and of giving the person a better quality of life. Brain stimuli (e.g., in Alzheimers) attenuate the progression of the disease, which results in an improvement in the quality of life. Thus attaining a better interaction and communication with the society in which he is embedded. On the other hand, according to the trials, this intervention is not appropriate for people in the later stages of dementia [6]. When the degree of incidence is moderate or severe, the loss of autonomy may be inevitable. The presence of a caregiver may be needed or the person may be reallocated to a nursing home or to a relative's house.

Despite the possible disability that a person may face, it is of extreme importance to remember that each person must have equal recognition before the law, must not be submitted to inhumane treatment, has the right to respect and has the right to live independently and to be included in the community. These are some of the rights expressed in the United Nations Convention on the Rights of Person's with Disabilities, namely articles 12, 15, 17, and 19 [7]. Indeed, despite the possible existence of a disability, people must be treated equally and have the same rights. Thus, a disability should not in any case impose a barrier to ordinary living. However, this is sometimes forgotten by modern societies where the absence of, for example, ramps for wheelchairs in crosswalks can be found.

Another possibility is to apply the concept of smart house [8], in which several embedded devices in the environment may allow for the monitoring of the person with disabilities, enabling remote access to the collected data by health professionals or caregivers. There are projects under the Active and Assisted Living (AAL) initiatives in Europe like Mylife [9] designed for older persons with reduced cognitive disabilities where the goal is to provide software as a service, giving access to simple and intuitive services that are adapted to the individual needs and wishes of each person. The aim of this project is to increase the independence and wellbeing of the user in home environments, and to decrease his social isolation by letting the user easily contact other people, among others.

Previous identified alternatives do not guarantee safety when the monitored person goes outside. Indeed, the lack of orientation is one of the causes for the loss of autonomy by people with mild or moderate cognitive disabilities. To create applications taking into consideration the end-user, it is necessary to consider features in its design in order to maximize its usability and accessibility. The cognitive and/or physical limitations of the user can be minimized since the cognitive processing is reduced to a minimum. This reduction can be achieved through alternatives to written text, such as pictures, animations and sounds [10]. The development of such applications must also consider the ability of the user to understand the instructions and to follow them. Thus, despite the level of the disabilities, the user must be able to use the application in an independent way.

To minimize the loss of independence we are engaged in developing a system that through an interface based on augmented reality (in order to diminish the effort needed to understand the displayed information), guides the user when traveling outside its premises. This system has a Speculative Computation unit which may anticipate possible user mistakes and alert him before taking the wrong turn. The path is also adapted to user preferences through a trajectory mining module. Our system also has a localization feature in order to keep caregivers aware of where the person with disabilities is. The developed system is aimed for people with mild or moderate cognitive disabilities with orientation deficit considering acquired disabilities such as the ones resulting from a traumatic brain injury, or those that are life long, such as Down's Syndrome, which do not evolve to higher

degrees of incidence over time. However, it is assumed that the user is able to use and remembers how to use the mobile device during the guidance process.

The main contributions of this work are: (1) a brief description of orientation systems for people with cognitive disabilities and the identification of their main features; (2) the development of a system that guides the user, taking into consideration his preferences; (3) the development of a system that enables caregivers to remotely access the location of the person with disabilities; (4) the description of the module that adapts the path to the user preferences; (5) the description of the module that anticipates possible user mistakes and alerts him before he takes the wrong turn. The developed system has a double goal: (1) guide the user through an adapted trajectory, considering his preferences; (2) provide a localization feature for caregivers to enable them to know, in real-time, the current location of the person with disabilities. Using CogHelper, both users may see his independency increased since the person with disabilities may travel alone without the feeling of getting lost, and caregivers may develop another activity without neglecting the provided care due to the ability of remotely knowing the location of the user. Another topic that should be highlighted is the previously cited articles of the United Nations Convention on the Rights of Person's with Disabilities [7]. With this project there is the social goal of providing simple tools to enhance normal living.

This document is organized as follows: Section 1 introduces the scope of the document. Section 2 presents the main features on which orientation systems are grounded and the current state of the art with respect of orientation methods. Section 3 describes the proposed system. Section 4 explains the use of Speculative Computation in the system. Section 5 details the process of adapting the user path to his preferences through a trajectory mining process. Finally, Section 6 presents conclusions and future work directions.

## 2. Orientation Systems

The reduction in the quality of life caused by the lack of spatial orientation was studied in [11,12], where it is stated that there is a low applicability of orientation systems from a user perspective, i.e., people with diminished cognitive or physical capacities may not use these systems.

In traditional methods there is the presence of an instructor dedicated to the person which, through a lengthy and labor intensive training, was able to teach the person with disabilities to use a public transportation system, the bus, in an independent way [13]. This process that, on average, took one year considers the ability of the person to learn one route. Newer routes could take between one to eight weeks. However, if the person stopped performing the trained route, it could easily be forgotten and all intensive and exclusive work would be rapidly wasted [13].

Orientation methods may epitomize an important progress in assistive technology. These are very rigorous methods that may not tolerate faults and correctly respond to user stimuli. As such, existing systems place an extreme importance on the interfaces. They must be, simultaneously, simple and complete, i.e., they have to show/provide available features/functionalities in an easy way, so the person with disabilities may use the application without feeling confused or losing his confidence in these systems.

One major factor for the acceptance and use of an application is, according to the study of Dawe [4,14], the user interface. In her work, both caregivers and people with disabilities were integrated in the development of the application and, according to the results, this integration led to a better adoption of the system. It was also confirmed that the system was developed according to the user preferences and that they were not assumed by the programmer.

In Table 1, a perspective is given of the system features considered in different works. Here, the orientation method is related to the user interface, while prompts consider the type of alerts given to the user. The localization feature (useful for caregivers) is expressed as real time monitoring, i.e., on the fly. As depicted in the table, there are few proposals for guiding the user with cognitive disabilities, and, even in those, the information is only given through audio and visual (textual) prompts. In Table 1, with a tick (✓), one has the features that were implemented in each

project. The last two columns of the table present works in which the main goal is not the user orientation, but his remote monitoring and next movement detection. These tasks are accomplished under a context aware setting.

**Table 1.** System features covered by existing projects.

Features	Projects	Carmien et al. [13]	Liu et al. [15]	Ramos et al. [16,17]	Fraunhofer Portugal [18]	Patterson et al. [19]	Sadilek & Kautz [20]
Orientation Method	Images		✓				
	Compass				✓		
	Augmented Reality			✓			
Prompts	Audio	✓	✓	✓	✓		
	Visual	✓	✓	✓	✓		
Context Aware						✓	✓
User Frequent Behaviour				✓		✓	
Real Time Monitoring				✓	✓	✓	✓

Although these orientation methods have a significant impact on the quality of life, and use up-to-date technologies and different types of hardware (like accelerometers, magnetometers, and GPS), they lack reasoning methods. These methods allied to the most recent technology create extremely promising systems in assistive technology, since they conciliate the fast technological progress with embedded artificial intelligence features. These systems allow for a “humanization” of the technology, enabling it to take into account user necessities. In a robotic scenario, the goal is to let an agent (or robot) to serve the purpose of moving an object from one place to another, taking the context into consideration. On the one hand, these decisions must be taken on-the-fly, and also in situations where the sensory data may not always be available. Indeed, the robot moves under an incomplete (data/information/knowledge) scenario. On the other hand, new data obtained from the sensors is continuously streamed to update the belief state of the robot (or agent) [21,22]. Under these frameworks and in order to develop an autonomous agent-based system, there are multiple ways to treat default data, information or knowledge, which are key factors in the process of enabling an agent to learn and to plan its actions. Table 2 provides an overview of the most prominent reasoning methods used in localization systems. It is outside of the scope of the paper to provide a detailed description of each one. The goal is to highlight the ones considered more suitable to be applied to the problem under analysis.

**Table 2.** Comparison between different reasoning methods.

Reasoning Methods	Complexity	Fault Tolerance	Learning Mechanism	Reward/Feedback	Input
Decision Tree Learning	NP-hard	✗	Supervised		✓
Artificial Neural Networks	High processing	✓	Unsupervised		✓
Support Vector Machines		✓	Supervised		✓
Bayesian Networks		✓			✓
Markov Decision Process		✓	Unsupervised	Reward function	✗
Case Based Reasoning		✓	Supervised	User Feedback	Knowledge Base
Speculative Computation		✓	Unsupervised	✗	Default Values

Decision Tree Learning is a method with predictive abilities through knowledge extraction and learning [23]. However, the Decision Tree has to be fed with a set of input values [24]. Thus, an initial supervised learning phase is necessary in order to enable this reasoning method to autonomously classify future inputs.

Inspired on biological systems, Artificial Neural Networks (ANNs) are able to model complex problems [25,26]. Through massive parallelism, these networks are able to learn and adapt to future

cases. ANNs are also fault tolerant, i.e., when there is some missing information the ANNs are able to keep the computation.

Data classification may also be automated by Support Vector Machines [27,28]. After a training phase, the Support Vector Machine may handle new data and label it. Although this method has a predictive feature, it has to be previously trained.

Based on the Bayes Theorem, a Bayesian Network is able to reason under incomplete information scenarios [29]. A Bayesian Network establishes causal relations between its nodes through a probability distribution. Thus, given an input value this reasoning method calculates possible outcomes with the respective probability. Applying this method to the guiding context has its advantages since it could predict the next movement of the user. However, there is still the need for previous user information to be fed into the network in order to train it.

The Partially Observed Markov Decision Process (POMDP) has proven its advantages in fields like robotic navigation and planning under uncertainty. This reasoning method has been successfully applied in orientation systems for people with cognitive disabilities [30]. A POMDP is able to select the next action and, based on a reward function, evaluate if that action produced a positive or negative impact on the user. However, in an initial phase this method starts with a naive approach, in which it is assumed that the options will be correctly followed every time. With the usage, the POMDP evolves and better adapts to future cases.

Case-based Reasoning is a method that resembles the human thinking method, i.e., when facing a problem the user recalls similar situations and adapts the previously used ones to the new one [31]. Past experiences may be used through some adaptation to the new situation or to know what should not be done in order to avoid negative outcomes. This reasoning method needs a knowledge base from which a previous case is retrieved and adapted to solve a new problem. Once the reuse phase is over, there is the need to evaluate the acquired solution before saving it back to the knowledge base. This evaluation may be dependent on user feedback.

In this work, it is assumed that Speculative Computation [32,33] has a more adjusted knowledge representation and reasoning method, as it may anticipate the next movement of the user and provide alerts. This feature may not be found in statistical learning, nor is a pattern recognition method, and may not be found in reinforcement learning (since in the reinforcement learning method the reward is returned after the action is taken). Speculative Computation has the ability to deal with incomplete information, i.e., when the missing data is received the information is processed, the computation is revised and the system keeps executing the orientation guidelines. Indeed, the system is previously loaded with a set of default values, with information about the travel paths (e.g., which directions should the user take, what are the turns that should be done), and information about possible missing turns (e.g., locations in which the user may take the wrong direction and get lost). One is faced with a scenario that uses default values and keeps the execution of the program. Thus, the system does not go under an idle state and when the information is received, the execution is revised. Another aspect that must be referred to is related with a process of movement anticipation, i.e., at any moment the system is able to predict how the user acts, and therefore prevents mistakes. This reasoning method has never been used, as far as we know, in a system intended to guide a user with cognitive disabilities. The speculative computation framework uses a set of default values which are obtained from a trajectory mining module (described in detail in Section 5). This method may not be considered a substitute to the previously described reasoning methods, but a complementary one. Indeed, the approach used to obtain the default values is independent from the execution of the speculative computation framework. In this work a trajectory data mining is applied, but it may be replaced by another machine learning method. The Speculative Computation method offers a way to deal with incomplete information and to update it in real-time.

### 3. CogHelper—An Orientation System for People with Cognitive Disabilities

Nowadays, it is possible to have small devices with huge processing power (e.g., smartphones). This kind of device brought more portability than existing solutions (like laptops), since the user may install and execute several applications according to his needs. Indeed, although the old solutions help the user in his tasks, some of these applications may be too complex to be used by people with disabilities, especially by those with cognitive ones.

Being able to set the user needs, the system may adapt itself. For instance, if the system detects that the user is confused or lost, it may put an emphasis on the orientation process, minimizing the risks of a wrong move. It uses data gathered by a context aware framework, which is fed to a system that works in a speculative way, i.e., the system may predict the user movements and adjust its alerts accordingly.

The name of our system is CogHelper and it helps people with cognitive disabilities and their caregivers [17]. This system is intended to work outdoors, and its structure (Figure 1) consists of internal services like the application for people with cognitive disabilities (*Cognitive Helper Mobile Solution*, Section 3.1), the application for caregivers (*Caregiver Applications*, Section 3.2), and the server, as well as external services that let the system use other facilities, like an external reminder. The framework of Speculative Computation and the trajectory mining module are described in detail in Sections 4 and 5, respectively. These features are considered internal services of the application *Cognitive Helper Mobile Solution*.

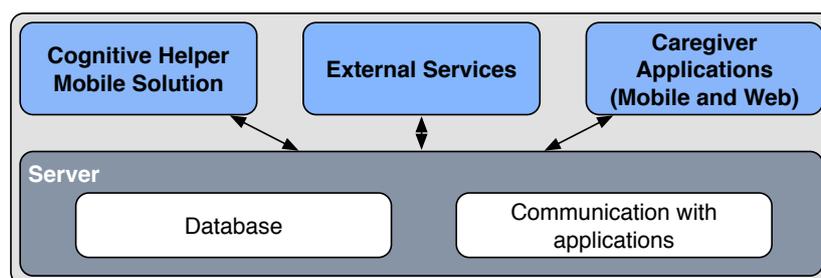


Figure 1. Simplified Framework of CogHelper.

The server is the core of the system, since all the other services are connected to it, namely in terms of a database (to store all the data needed to its correct operation, like usernames, locations, points of interest), and a communication module to connect to other internal services, as well as to external ones.

#### 3.1. Application for People with Cognitive Disabilities

With the advent of high processing powered mobile devices, it was possible to overcome a major limitation of smart houses. Instead of having multiple fixed sensors at a home environment, it is possible to develop a portable monitoring bubble using mobile devices and sensors (e.g., over the body or clothes), thus leading to a new concept or abstraction, the Body Area Network (BAN) [34–36].

The system developed so far operates under the Android Operative System, targeting human beings with cognitive disabilities. It is based on the BAN concept. It fulfills two main objectives: to guide the user by adapting the user path and anticipating possible user mistakes, and to provide his localization to caregivers. Not only does it enable the user to travel between two locations without getting lost, but it also provides a localization system, so caregivers may know in real time the current location of the user.

A detailed framework of the *Cognitive Helper Mobile Solution* is presented in Figure 2 and consists of a Localization Layer, with methods to obtain the user location, necessary to let the Navigation system settle the route. The user location may be achieved through the GPS module of the mobile device,

or by a coarse location from the network. This layer also includes a Navigation Algorithm, which is based on a Speculative Computation Framework, described in detail in Section 4, and a Trajectory Mining Module (responsible for obtaining the default values for the speculative computation module), described in Section 5. With the information from the user location and the Speculative framework, the system may learn from the frequent paths taken by the user.

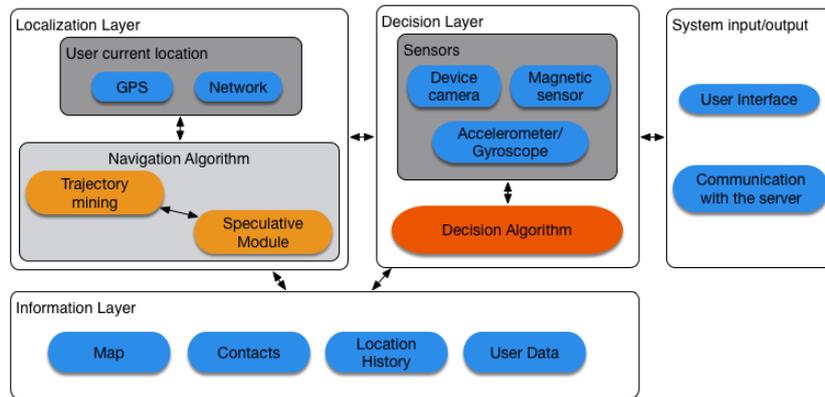


Figure 2. Detailed framework of the application for people with cognitive disabilities.

Once the destination is selected, using tools like cameras, accelerometers, or magnetic sensors, the route is presented to the user through augmented reality (Figure 3). In other words, one may say that the computational system is able not only to set the direction to where the user must point, but also to present it to him. If the user is pointing to the right direction, a green arrow will appear on the screen, signaling that the chosen track is the correct one. On the other hand, if the user is pointing to the wrong direction, a red cross is displayed. The decision process (at the center in Figure 2) ensures that the user is moving on the right track, and some warnings/prompts are brought to him, whenever necessary. These prompts may be seen as a stimulus to the user to keep himself on the right track, or to alert him to take the right decisions. To keep caregivers aware of user movements, an email or short message may also be sent to them, i.e., when the user reaches the destination point, passes a control one, or gets lost. Indeed, through an interface (right part in Figure 2) the user may communicate with the computational system and to set the destination points, namely in terms of starred destinations (e.g., home, school, or office), or generic ones (e.g., the mall). To communicate with the server in order to obtain and send the needed information, the computational system is provided with an Input/Output layer. The information exchanged between the system and the server may include the update of user destinations, the update of his current location, or information about the external services that are available.



Figure 3. Orientation system using augmented reality.

### 3.2. Applications for Caregivers

To look at and follow the user throughout his lifetime, caregivers have access to two different platforms, namely a mobile application for the Android Operative System (Figure 4a), and a Web application (Figure 4b). The former gives extra degrees of freedom to caregivers, since it may be used everywhere with an Internet connection. The detailed framework of these platforms is shown in Figure 5, and consists of three components, being the extra module exclusive for the Web application, which is pointed out as “Web”. On the left it is possible to see the Input/Output layer, which enables the interaction between the caregiver and the computational system. There is also a Communication Module for the communication between the caregiver applications and the server, in order to send and receive requests, like getting the current position of the user.



Figure 4. Localization system for caregivers. (a) Android application; (b) Web application.

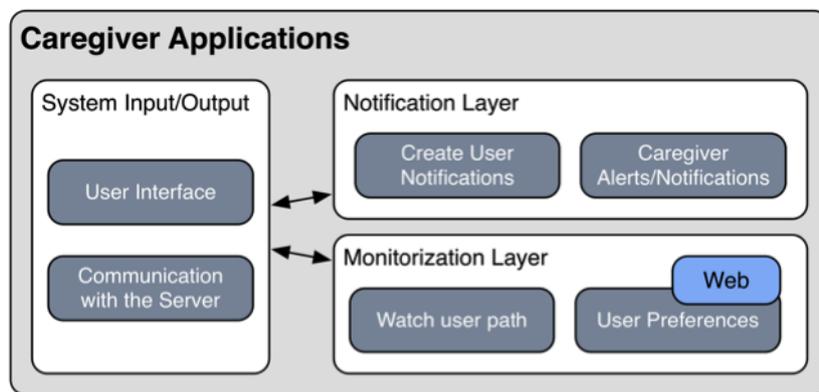


Figure 5. Detailed framework of the caregiver’s application.

The Notification Layer is in charge of handling the alerts that come from the user application. These alerts may inform the caregiver that the user is lost or that they have reached their destination. The Create User Notification Module enables the caregiver to create notifications to the user, which are no more than small text messages that may include questions with a Yes/No answer (e.g., the user may be asked if he needs assistance). Finally, the framework has a Monitoring Layer, which enables the caregivers to be aware of user movements, like knowing his current location or the traveling path. The Web application is the only one that has a User Preferences Module, through which the caregiver may edit some preferences of the person with disabilities. This feature includes, for example, the creation of destination points, which may be done by a direct selection on the map or by searching an address. One is also able to state if the destination point is general, or favorite (i.e., more frequently used).

### 3.3. Ethical Issues

The development of an application which needs or collects sensitive information about the user must be handled with care. In CogHelper's particular case, the main users are considered to be specially vulnerable, people with disabilities, and sensitive information is stored. Indeed, despite storing all information in a ciphered form, regarding user position, user location, among other data, by using the system it is assumed that the user has given his consent to provide all necessary information for the correct operation of the system. The collected information is used to adapt the system to each individual user and, if requested, this is provided to the caregiver.

According to [37,38] and taking into consideration research purposes, privacy is a priority when dealing with vulnerable populations and the published results will not compromise or reveal sensitive information about the user. With this system there is the intention of maximizing the benefits, adapting the trajectory to the user preferences and anticipating possible user mistakes, and no information except what is absolutely necessary will be collected or demanded of the user.

## 4. Applying Speculative Computation to an Orientation System

Satoh [32] extended the procedure of Kakas et al. [33], giving rise to the theory of Speculative Computation and Abduction, which is a step forward in handling incomplete information. Instead of having the computational process in an idle state (i.e., waiting for incoming information to continue the computation), it moves ahead by replacing the unknown information by default information, therefore obtaining a tentative solution to the problem. Whenever the missing data is acquired, the computation process is re-examined. The execution of the speculative framework is based on two phases, *Process Reduction Phase* and *Fact Arrival Phase*. The former one stands for the normal execution of the computation process, while the latter denotes an interruption stage. Before starting the execution of the Speculative Computation framework, it is necessary to have the following information:

1. All the possible paths between the current user location and the intended destination as facts in the knowledge base;
2. The transitions between points usually performed by the user as default values;
3. Information regarding the inclusion of a point in the current recommended traveling path as default values; and
4. A set of rules that structure the execution of the computation regarding the most likely path that the user will follow and the issuing of alerts if a (potential) mistake happens.

### 4.1. The Speculative Computation Framework

The Speculative Computation Framework in the Orientation Method ( $SF_{OM}$ ) for people with cognitive disabilities is defined in terms of the tuple  $\langle \Sigma, \mathcal{E}, \Delta, \mathcal{A}, \mathcal{P}, \mathcal{I} \rangle$  [32], where:

- $\Sigma$  stands for a finite set of constants (an element of  $\Sigma$  is called a system module);
- $\mathcal{E}$  denotes a set of functions called *external predicates*. When  $Q$  is a literal belonging to an external predicate and  $S$  is the identifier of the information source,  $Q@S$  is called an *askable literal*. We define  $\sim(Q@S)$  as  $(\sim Q)@S$ ;
- $\Delta$  is the *default answer* set, which is a set of ground askable literals that satisfy the condition:  $\Delta$  does not contain both  $p(t_1, \dots, t_n)@S$  and  $\sim p(t_1, \dots, t_n)@S$  at once;
- $\mathcal{A}$  is a set of predicates called *abducible predicates*.  $Q$  is called *abducible* when it is a literal with an *abducible predicate*;
- $\mathcal{P}$  is a Logic Program (LP) and contains a set of rules in the form:
  - ▷  $p \leftarrow p_1, p_2, \dots, p_n$  where  $p$  is a positive ordinary literal, and each of  $p_1, \dots, p_n$  is an ordinary literal, an askable literal or an abducible; and
  - ▷  $p$  is the head of rule  $R$  and is named as *head(P)* (always non-empty), in which  $R$  is a rule of the form  $p \leftarrow p_1, \dots, p_n$ ; where  $p_1, \dots, p_n$  is the body of the rule denoted as *body(P)*, that in some situations is replaced by the boolean value *true*.

- $\mathcal{I}$  is a set of integrity constraints in the form:
  - ▷  $?(p_1, p_2, \dots, p_n)$ , where the symbol “?” denotes “falsity”, the  $p_1, p_2, \dots, p_n$  are ordinary literals or *askable literals* or *abducibles*. At least one of  $p_1, p_2, \dots, p_n$  is an *askable literal* or an *abducible*.

An *askable literal* may have different meanings, namely:

1. An *askable literal*  $Q@S$  in a rule  $\mathcal{P}$  stands for a question put to a system module  $S$ ; and
2. An *askable literal* in  $\Delta$  denotes a default truth value, either *true* or *false*, i.e.,  $p(t_1, \dots, t_n)@S \in \Delta$ ,  $p(t_1, \dots, t_n)@S$  is usually *true* for a question to a system module  $S$ , and  $\sim p(t_1, \dots, t_n)@S \in \Delta$ ,  $p(t_1, \dots, t_n)@S$  is generally *false* for a question to a system module  $S$ .

In the logic program given below  $path(a,b)$  denotes that it is possible to travel between locations  $a$  and  $b$ ;  $show\_next\_point$  states that the system must show the next location (which may be intermediate or final) to the user;  $show\_user\_warning$  indicates that the system must alert the user, given that he is going in the wrong direction; and the default values for the travel path of the user are defined in  $\Delta$ .  $user\_travel(a,b)$ , says that the user will travel from location  $a$  to location  $b$ ;  $included(a)$  is evidence that location  $a$  is part the route.

To ensure that the user is traveling on the correct track and that he is alerted whenever he is off of it, a supporting structure was specified based on Speculative Computation, in terms of the logic programming suite (LPS):

- ▷  $\Sigma = \{gps\_sensor, recognizer\}$
- ▷  $\mathcal{E} = \{user\_travel, included\}$
- ▷  $\Delta = \{user\_travel(1,2)@gps\_sensor,$   
 $user\_travel(2,3)@gps\_sensor,$   
 $user\_travel(2,4)@gps\_sensor,$   
 $user\_travel(4,5)@gps\_sensor,$   
 $user\_travel(5,6)@gps\_sensor,$   
 $user\_travel(6,3)@gps\_sensor,$   
 $included(1)@recognizer,$   
 $included(2)@recognizer,$   
 $included(3)@recognizer,$   
 $\sim included(4)@recognizer$   
 $included(5)@recognizer,$   
 $included(6)@recognizer\}$
- ▷  $\mathcal{A} = \{show\_next\_point, show\_user\_warning\}$
- ▷  $\mathcal{P}$  is the following set of rules:
  - $guide(A, A) \leftarrow .$
  - $guide(A, B) \leftarrow$   
 $path(A, F),$   
 $show\_next\_point(F),$   
 $user\_travel(A, F)@gps\_sensor,$   
 $guide(F, B).$
  - $guide(A, B) \leftarrow$   
 $path(A, F),$   
 $user\_travel(A, F)@gps\_sensor,$   
 $show\_user\_warning(F),$   
 $guide(F, B).$
  - $path(1,2) \leftarrow .$
  - $path(2,3) \leftarrow .$
  - $path(2,4) \leftarrow .$

$path(4,5) \leftarrow .$   
 $path(5,6) \leftarrow .$   
 $path(6,3) \leftarrow .$

▷  $\mathcal{I}$  denotes the following set of integrity constraints or invariants:

$? (show\_next\_point(F),$   
 $\sim included(F)@recognizer).$   
 $? (show\_user\_warning(F),$   
 $included(F)@recognizer).$

To ensure program integrity, two invariants were added that state that the system may not show the next route point to the user if it is not part of it, or that the system may not alert the user if he is moving on the right track.

As an example, it is assumed that the user will travel between locations 1 and 3 through intermediate location 2. An elucidation of the possible paths that the user may use are presented in Figure 6.

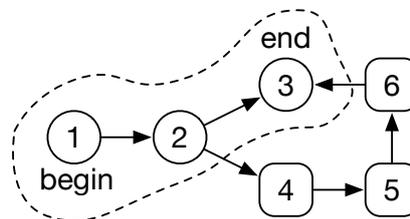


Figure 6. Possible ways to move among locations 1 and 3.

#### 4.2. Preliminary Definitions

There are some aspects of the formal process set above that must be defined in order to translate it into a proof procedure.

**Definition 1.** An extended literal is either a literal or an expression of the form  $fail(\{l_1, \dots, l_n\})$  where  $l_i$  is a literal.  $fail(\{l_1, \dots, l_n\})$  is used to prove that there is no proof for  $l_i$  [39].

**Definition 2.** A process is the tuple  $\langle GS, OD, IA, ANS \rangle$  in which  $GS$  is a set of extended literals and called a Goal Set (GS), expressing the current status of an alternative computation;  $OD$  is a set of askable literals called Outside Defaults (OD), which denote a set of assumed information about the outside world during a computation;  $IA$  is a set of negative literals or abducibles called Inside Assumptions (IA) that stand for the values assumed during a computation; and  $ANS$  is a set of instantiations of variables in the initial inquiry (named Answers - ANS).

**Definition 3.**  $PS$  is a set of processes. A set of Already Asked Questions  $AAQ$  is a set of askable literals. A Current Belief State  $CBS$  is a set of askable literals.

The set of processes  $PS$  expresses all the alternative computations that were considered. The  $AAQ$  set is used to avoid asking redundant questions to the sensors.  $CBS$  is the current belief state and expresses the current status of the outside world. It is also important to define an *active process* and a *suspended process*.

**Definition 4.** Let  $\langle GS, OD, IA, ANS \rangle$  be a process and  $CBS$  be a current belief state. A process is active with respect to  $CBS$  if  $OD \subseteq CBS$ . A process is suspended with respect to  $CBS$  otherwise.

The definition of an active process emphasizes that it is a process whose outside defaults have to be consistent with the current belief state.

### 4.3. Process Reduction Phase

In this phase, changes may occur in the process set. In the following description, changed  $PS$ ,  $AAQ$  and  $CBS$  are specified as  $NewPS$ ,  $NewAAQ$  and  $NewCBS$ ; otherwise they stay unchanged.

**Initial Step:** Let  $GS$  be an initial goal set. The tuple  $\langle GS, \emptyset, \emptyset, ANS \rangle$  is given to the proof procedure where  $ANS$  is a set of variables in  $GS$ . That is,  $PS = \{\langle GS, \emptyset, \emptyset, ANS \rangle\}$ . Let  $AAQ = \emptyset$  and  $CBS = \Delta$ .

**Iteration Step:** Do the following:

- ▷ **Case 1:** If there is an active process  $\langle GS, \emptyset, \emptyset, ANS \rangle$  with respect to  $CBS$  in  $PS$ , terminate the process by returning outside defaults  $OD$ , inside assumptions  $IA$ , and instantiation for variables  $ANS$ . This case may only be applied on the first iteration step since  $OD$  and  $IA$  are empty sets;
- ▷ **Case 2:** If there is no active process, terminate the process by reporting a failure of the goal;
- ▷ **Case 3:** Select an active process  $\langle GS, OD, IA, ANS \rangle$  with respect to  $CBS$  from  $PS$  and select an extended literal  $L$  in  $GS$ . Let  $PS' = PS - \{\langle GS, OD, IA, ANS \rangle\}$  and  $GS' = GS - \{L\}$ . For the selected extended literal  $L$ , do the following:
  - **Case 3.1:** If  $L$  is a positive ordinary literal,  $NewPS = PS' \cup \{\langle \{body(R)\} \cup GS'\theta, OD, IA, ANS\theta \rangle \mid \exists R \in \mathcal{P} \text{ and } \exists \text{most general unifier } \theta \text{ so that } head(R)\theta = L\theta\}$ .
  - **Case 3.2:** If  $L$  is a ground negative ordinary literal or a ground abducible then:
    - \* **Case 3.2.1:** If  $L \in IA$  then  $NewPS = PS' \cup \{\langle GS', OD, IA, ANS \rangle\}$ .
    - \* **Case 3.2.2:** If  $\bar{L} \in IA$  then  $NewPS = PS'$ .
    - \* **Case 3.2.3:** If  $L \notin IA$  then  $NewPS = PS' \cup \{\langle NewGS, OD, IA, \cup\{L\}, ANS \rangle\}$  where  $NewGS = \{fail(BS) \mid BS \in resolvent(L, \mathcal{P} \cup \mathcal{I})\} \cup GS'$  and  $resolvent(L, T)$  is defined as follows:
      - If  $L$  is a ground negative ordinary literal,  $resolvent(L, T) = \{\{L_1\theta, \dots, L_k\theta\} \mid H \leftarrow L_1, \dots, L_k \in T \text{ so that } \bar{L} = H\theta \text{ by a ground substitution } \theta\}$
      - If  $L$  is a ground abducible,  $resolvent(L, T) = \{\{L_1\theta, \dots, L_{i-1}\theta, L_{i+1}\theta, \dots, L_k\theta\} \mid \perp \leftarrow L_1, \dots, L_k \in T \text{ so that } L = L_i\theta \text{ by a ground substitution } \theta\}$ .
  - **Case 3.3:** If  $L$  is  $fail(BS)$ , then
    - \* If  $BS = \emptyset$ ,  $NewPS = PS'$ ;
    - \* If  $BS \neq \emptyset$ , then do the following:
      - (1) Select  $B$  from  $BS$  and let  $BS' = BS - \{B\}$ .
      - (2) **Case 3.3.1:** If  $B$  is a positive ordinary literal,  $NewPS = PS' \cup \{\langle NewGS \cup GS', OD, IA, ANS \rangle\}$  where  $NewGS = \{fail(\{body(R)\} \cup BS')\theta \mid \exists R \in \mathcal{P} \text{ and } \exists \text{MGU } \theta \text{ so that } head(R)\theta = B\theta\}$
      - Case 3.3.2:** If  $B$  is a ground negative ordinary literal or a ground askable literal or an abducible,  $NewPS = PS' \cup \{\langle \{fail(BS')\} \cup GS', OD, IA, ANS \rangle\} \cup \{\langle \bar{B} \rangle \cup GS', OD, IA, ANS \rangle\}$ .
  - **Case 3.4:** If  $L$  is a ground askable literal,  $Q@S$ , then do the following:
    - (1) If  $L \notin AAQ$  and  $\bar{L} \notin AAQ$ , then send the question  $Q$  to the slave agent  $S$  and  $NewAAQ = AAQ \cup \{L\}$ .
    - (2) If  $\bar{L} \in OD$  then  $NewPS = PS'$  else  $NewPS = PS' \cup \{\langle GS', OD \cup \{L\}, IA, ANS \rangle\}$ .

### 4.4. Fact Arrival Phase

In this phase the current belief state is revised according to the information received from the sensors. Supposing that an answer  $Q$  is returned from a sensor  $S$ . Let  $L = Q@S$ . After finishing a step of process reduction, let us do the following:

- If  $\bar{L} \in CBS$ , then  $NewCBS = CBS - \{\bar{L}\} \cup \{L\}$

- Else if  $L \notin CBS$ , then  $NewCBS = CBS \cup \{L\}$ .

There might be some askable literals that are not included in the initial belief set. If this occurs, processes that are using such askable literals and those using their complements are suspended until the answers are returned.

#### 4.5. Correctness of the Proof Procedure

The correctness of the procedure is guaranteed by stable model semantics [39]. Thus, the following definitions are given for the semantics of the previously described logical program.

**Definition 5.** Let  $T$  be a set of rules and integrity constraints. The set of ground rules obtained by replacing all the variables in every rule or every integrity constraint  $T$  by every ground term is denoted as  $\Pi_T$ .

**Definition 6.** Let  $T$  be a set of rules and integrity constraints. Let  $M$  be a set of ground atoms and  $\Pi_T^M$  be the following program:  $\Pi_T^M = \{H \leftarrow B_1, \dots, B_l \mid H \leftarrow B_1, \dots, B_l, \sim A_1, \dots, \sim A_h, \in \Pi_P \text{ and } A_i \notin M \text{ for each } i = 1, \dots, h.\}$ . Let  $\min(\Pi_T^M)$  be the least model of  $\Pi_T^M$ . A stable model for a logic program  $T$  is  $M$  iff  $M = \min(\Pi_T^M)$  and  $\perp \notin M$ .

**Definition 7.** Let  $T$  be a set of rules and integrity constraints, and  $\Theta$  be a set of ground abducibles.

For any process evaluation strategy, when an answer is received with a set of outside defaults and a set of inside assumptions from the proof procedure, the answer is correct with respect to a generalized stable model with respect to inside assumptions and the program. This program is obtained from the original one and the current world belief.

**Theorem 1.** Let  $SF_{OM} = \langle \Sigma, \mathcal{E}, \Delta, \mathcal{A}, \mathcal{P}, \mathcal{I} \rangle$  be a speculative framework where  $\mathcal{P}$  is a call-consistent logic program whose set of integrity constraints is satisfiable. Let  $GS$  be an initial goal set. Suppose that  $GS$  is reduced to  $\emptyset$  when  $OD$  is outside defaults.  $IA$  are inside assumptions,  $ANS$  is the set of variables instantiations in  $GS$ , and  $CBS$  is the current belief set. Let  $GS'$  be a goal obtained from  $GS$  by replacing all the variables in  $GS$  by  $ANS$ . Then, there is a generalized stable model  $M(\Theta)$  for  $\mathcal{P} \cup \mathcal{I} \cup \mathcal{F}(CBS)$ , such that  $M(\Theta) \models GS'$  and  $OC \subseteq CBS$  and  $IA \subseteq \Theta$ .

**Proposition 1.** Since  $\mathcal{P}$  is a call consistent logic program, so is  $\mathcal{P} \cup \mathcal{I} \cup \mathcal{F}(CBS)$ . Then, an abducible derivation may be constructed [39] using a set of reduction steps applied to  $\langle GS, \emptyset, \emptyset, ANS' \rangle$  to  $\langle \emptyset, OD, IA, ANS \rangle$  where  $ANS'$  is a set of variables in  $GS$ . This derivation is correct for generalized stable model semantics for a call-consistent logic program [39]. Thus,  $M(\Theta) \models GS\theta$  and  $OD \subseteq \Theta$ .

An execution example of the program introduced in Section 4.1 is presented in Appendix A. For the reduction process, the following strategy is used:

- ▷ When a positive literal is reduced, new processes are created according to the rule order in the program, which are unifiable with the positive literal;
- ▷ A newly created or newly resumed process and the most left literal is always selected.

In the execution trace in Appendix A for *guide(1,3)* the selected literal is underlined in the active process that was sorted out. *AAQ* and *CBS* are only shown when a change occurs. In order to reduce the execution example and to facilitate its reading and interpretation, the literals have been abbreviated. Thus, *gps\_sensor* is represented by *g*, *recognizer* by *r*, *user\_travel* by *u*, *included* by *i*, *path* by *p*, *show\_next\_point* by *snp*, and *show\_user\_warning* by *suw*.

The program starts with the objective of guiding the user from location 1 to location 3. According to Figure 6 there are two possibilities, i.e., making their way through location 2 or through locations 2, 4, 5 and 6. The system starts processing two branches for each next possible location: one branch lets

the execution continue if the location is valid (i.e., it is included in the possible locations set) and the user is moving towards it, the other stands for the situation in which the location is not valid and the user is moving to it. In the former one, the system keeps guiding the user to the next point, whereas in the latter it alerts the user and guides him to the correct track. The execution of an alternative combination of what was previous described is suspended since that move is not valid (e.g., alert the user if the next location is valid).

During the execution of a particular move that may change positions, the information source may be queried. While this information is not returned the program continues its execution, so the moves which are queried  $\sim user\_travel(1,2) @gps\_sensor$  are paused (since the default value is negative). The only move that proceeds is the one that assumes the default value  $user\_travel(1,2)@gps\_sensor$ . Whenever an answer is returned from the information source, the computation is revised. The current move may be paused and a previously paused move may be resumed. The system may re-initialize, pause, or resume moves until the destination point is reached.

Instead of suspending the computation due to not knowing the answer to set the next user's movement, Speculative Computation enables the continuity of the computational task, i.e., it sets a default value as the way to have an answer to set the next user's movement and therefore the computational process does not come to an end (e.g., Step 8).

At Step 3,  $show\_next\_point(2)$  is assumed and the integrity constraints are checked. Thus, it is assured that there is no contradiction by checking if  $\sim included(2)@recognizer$  is not derived. In this step an ordinary abduction operation is carried out.

When an answer is returned and it confirms the default value, nothing changes. Thus, in these situations Speculative Computation is in an advanced stage of the computation and it does not have to be revised (e.g., Step 11).

At different stages of the computational process, an ongoing procedure may have to be suspended once the askable literal is in opposition with the information contained in the CBS. For instance, this situation occurred at Step 33.

The execution trace referred to above denotes an example of the computational process. Its purpose is to check and to demonstrate that it is possible to set the orientation problem through speculative computation. Indeed, according to the proof procedure referred to above, the application of Speculative Computation is an expressive and adequate option.

## 5. Applying Trajectory Pattern Mining to Adapt the Path

The module of Speculative Computation described in Section 4 has the goal of guiding the user and predicting possible wrongdoings. Indeed, the user may be alerted before taking the wrong turn and possibly getting lost. However, this speculative module needs to be fed with the travelling route, including the points where an alert should be triggered.

The Trajectory Pattern Mining Module has a double goal: to adapt the route to the user and to load this information into the Speculative Computation Module. By adapting the path to the user preferences, it is possible to achieve better results in the guiding process, since it is the system that adapts to the user needs and not the opposite. On the other hand, if one considers the cognitive disabilities of the user, it is important to highlight that a known trajectory (or part of the entire trajectory), despite being longer, is preferred by the user. Thus, the usual rigid way of traditional guiding systems is overcome. Indeed, the default method for calculating the best route (e.g., shortest path) is only considered when the system does not have information about the user habits (already travelled paths).

A sequence of time-stamped Cartesian coordinates defines the trajectory (or route) done by a person with cognitive disabilities. This data is obtained from the GPS sensor of the mobile device. Indeed, the location is composed by latitude, longitude, and timestamp values. This process resorts to an active recording of the user location whenever he is receiving the necessary aid from the application, i.e., is travelling using the guiding information provided by the application. Figure 7 depicts the

trajectory mining process. This may derive the trajectory by two different approaches based on the previous information that was stored. If there is no information about the user, since he may be using the application for the first time, or when the chosen destination does not have enough information to retrieve a pattern, the system calculates the path based on the default routing algorithm. This step is achieved through the external tool GraphHopper [40]. On the other hand, with the continuous use of the system, it is possible to retrieve previous information about the person with cognitive disabilities and to use this available spatial-temporal data to provide the best adapted trajectory to the user.

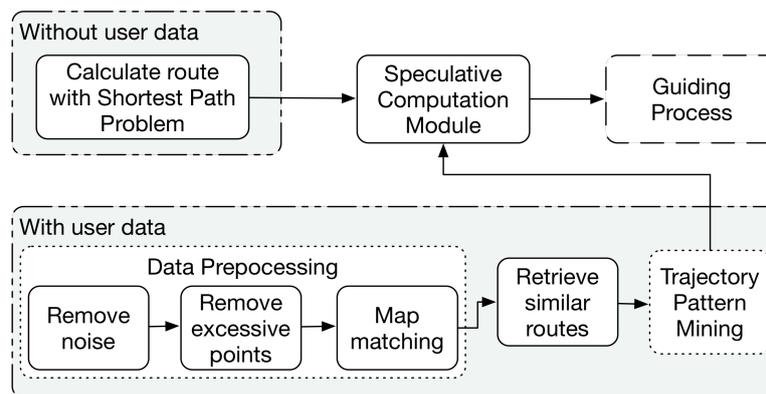


Figure 7. Process for generating adapted routes.

The raw data obtained from the device GPS module needs to be pre-processed in order to be used in the mining process to adapt the path to the user preferences. Indeed, this pre-processing phase (Figure 7) includes noise removal (from the raw data), excessive points removal (locations that do not increase the detail of the trajectory), and map matching. The noise removal phase is responsible for removing all locations where the GPS error is notorious, i.e., locations that, for some reason, like signal reflection in the surrounding buildings, have an error higher than the usual. In the second step, it is important to reduce the number of points by eliminating those that do not add value to the mining process, i.e., locations that are in the middle of a straight line. Finally, the last process is the map matching, where the goal is to get the remaining data and match it to the roads defined in the Open Street Map (Open Street Map, available at <http://www.openstreetmap.org/>) databases. Figure 8 illustrates the pre-processing stage. In Figure 8a, the raw data recorded from the mobile device GPS module is represented on the map, whereas in Figure 8b the final stage of the pre-processing phase is visible, which includes noise removal and map matching. For this pre-processing phase we use GraphHopper Directions Api [40], which has a map matching feature and is an open source tool.

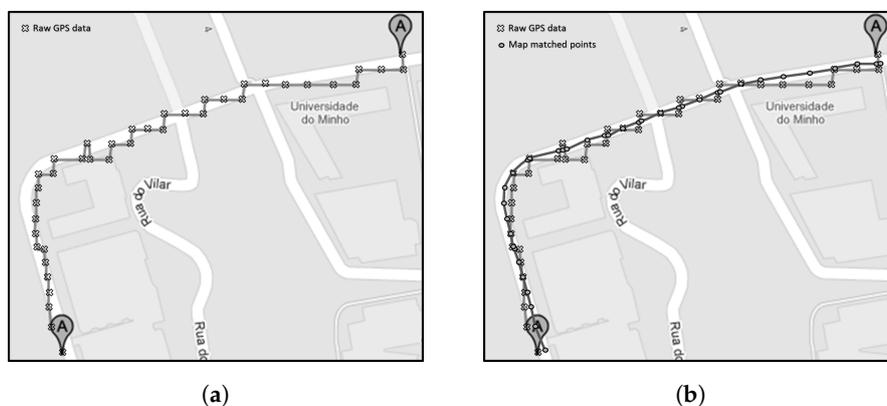


Figure 8. Trajectory data pre-processing (extracted from [41]). (a) Route obtained from database raw data; (b) Error removal from raw data.

When the pre-processing stage ends, the trajectory data mining method may start and extract the frequent trajectories (patterns). Without these previous steps, the results would be worse due to the noise that would be considered in the mining process. For the used trajectory data mining, two different methods are being used with good results: the PrefixSpan algorithm [42], which mines frequent sequential patterns; and OPTICS [43], which is a clustering algorithm that groups the locations in clusters. For this process we rely on an open source Application Programming Interface (API) named “SPMF - An Open-Source Data Mining Library” [44] that provides an implementation of both algorithms. Using the PrefixSpan algorithm, the extraction of frequent trajectories considers all travelling paths that have been done by the user. The results of the algorithm (frequent trajectories) consider the support of a sequential pattern, i.e., an input value that defines the minimum percentage of sequences in which the pattern occurs, considering the entire set of travelled paths. To adjust the path to the user preferences, in the mining process the system only considers paths travelled by that user, i.e., the user that is using the application and receiving the guiding information. The execution of the algorithm searches for patterns in the dataset according to the defined threshold. The obtained results (patterns) may represent an entire trajectory, or be part of the trajectory. In the former, the returned path represents the entire path that will be used to guide the user. There may be a need to adjust the starting point, but the remainder of the path (considering the selected destination) is the same. In the latter, a new travelling route is created based on the mined patterns, which is adapted to the user. When the recently created route is much longer than the default path, the system uses the shortest one, instead of the adapted one.

By applying the clustering method (OPTICS) the processed spatial-temporal data is grouped in clusters. For this process it is important to define some input parameters like the minimum number of points that are needed to define a core and its neighborhood, and the radius  $\epsilon$  that defines the maximum distance that a point may be from the core of the cluster (defining its neighborhood). After executing the OPTICS algorithm, the retrieved locations are grouped in clusters. Thus, it is necessary to create the connections between them in order to obtain a trajectory and guide the user from his current location to the pre-selected destination.

In the case of using the shortest path or in the case of applying the trajectory data mining method, the obtained trajectory needs to be converted in order to be used in the execution of the speculative computation.

## 6. Conclusions

An active participation in society is vital for people with cognitive disabilities and, having this in mind, numerous scholars have studied and developed various methods to orientate them, both indoors and outdoors. The independent traveling allows the user to maintain an employment (within their capabilities). The degree of independence for the main user and for the caregiver is increased, since the person with disabilities can be remotely controlled, and the caregiver may be otherwise engaged without neglecting his role as a caregiver.

The present orientation systems have to be adapted to the characteristics of the users, i.e., these systems must be easy to operate when the orientation is being performed. If these system specifications are not matched, it may imply a more complex mental activity, leaving the user confused. In this situation the orientation system may become infeasible.

For people with reduced orientation capabilities there are several projects under consideration. Using these systems, people with cognitive disabilities may have a more active life, reducing the worry of getting lost both indoors and outdoors. Depending on the type of cognitive disability and its degree of incidence, the user can keep a job or perform other activities. However, the user has to learn two types of orientation systems, since a system adapted to the outside world cannot be used indoors or vice versa. The system presented in this work stands for an answer to all these shortcomings, making a move to providing a better quality of life to disabled people.

Through this work we present an orientation system for people with mild (or moderate) cognitive disabilities that has an anticipation feature that alerts the user before he takes the wrong turn. This is achieved by applying a Speculative Computation module, which needs to be loaded by the traveling path before its execution. This specificity enables the independent development of a pattern mining module, so that travel path is adapted to the user (instead of using the shortest one by default). Caregivers are not disregarded and a localization feature was considered. Thus, these users may remotely access the traveling path of the person with disabilities.

In the future we intend to test the entire system in real case scenarios in order to check and, if necessary, add new features or improve the ones that were developed. People with cognitive disabilities that are able to use a smartphone will be selected for these real tests, since the performed tests were done using non-impaired people who tried to resemble a cognitive disability issue.

**Author Contributions:** All the authors proposed ideas for developing and improving the system; J.R. was the main programmer of the system; J.R. and T.O. developed and programmed the Speculative Computation Module; all the authors contributed to the analysis and interpretation of the data; and all the authors wrote the paper in a collaborative way.

**Acknowledgments:** This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT—Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013. The work of João Ramos is supported by a doctoral FCT grant SFRH/BD/89530/2012. The work of Tiago Oliveira is also supported by the FCT grant with the reference SFRH/BD/85291/2012.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A. Execution Example of the Speculative Computation Framework

An execution example is given below in which the user is guided from location 1 to location 3. In order to reduce the amount of text in the example, we used the abbreviated form of the elements of each tuple. Thus  $u(1,2)@g$  is the askable literal  $user\_travel(1,2)@gps\_sensor$ ;  $i(1)@r$  is the literal  $included(1)@recognizer$ ;  $p(1,2)$  represents  $path(1,2)$ ;  $snp(2)$  is the predicate  $show\_next\_point(2)$ ; and  $suw(2)$  represents the predicate  $show\_user\_warning$ . The sets  $AAQ$  and  $CBS$  are shown when a change occurs.

In this execution example the turn instructions consider the locations depicted in Figure 6. Thus, the system calculates the adapted traveling path, which takes locations 1-2-3, and that the user will probably take the wrong turn towards location 4. In this case a warning is prompted to the user which, for this example, will make the user take the correct path.

1.

$$\begin{aligned} PS &= \{ \langle \{ \underline{guide(1,3)} \}, \emptyset, \emptyset \rangle \} \\ AAQ &= \emptyset \\ CBS &= \{ u(1,2)@g, u(2,3)@g, u(2,4)@g, u(4,5)@g, \\ &u(5,6)@g, u(6,3)@g, i(1)@r, i(2)@r, \\ &i(3)@r, \sim i(4)@r, i(5)@r, i(6)@r \} \end{aligned}$$

2. By Case 3.1

$$\begin{aligned} PS &= \{ \langle \{ \underline{p(1,Y)}, snp(Y), u(1,Y)@g, guide(Y,3) \}, \\ &\emptyset, \emptyset \rangle, \langle \{ \underline{p(1,Y)}, u(1,Y)@g, \\ &suw(Y), guide(Y,3) \}, \emptyset, \emptyset \rangle \} \end{aligned}$$

3. By Case 3.1

$$\begin{aligned} PS &= \{ \langle \{ \underline{snp(2)}, u(1,2)@g, guide(2,3) \}, \emptyset, \emptyset \rangle, \\ &\langle \{ \underline{u(1,2)@g}, suw(2), guide(2,3) \}, \emptyset, \emptyset \rangle \} \end{aligned}$$

4. By Case 3.2.3

$$\begin{aligned} PS &= \{ \langle \{ \underline{fail(\{ \sim i(2)@r \})}, u(1,2)@g, guide(2,3) \}, \\ &\emptyset, \{ \underline{snp(2)} \} \rangle, P_1 (P_1 = \langle \{ \underline{u(1,2)@g}, suw(2), guide(2,3) \}, \emptyset, \emptyset \rangle) \} \end{aligned}$$

5. By **Case 3.3.2**

$$PS = \{ \langle \{ fail(\emptyset), u(1,2)@g, guide(2,3) \}, \emptyset, \{ snp(2) \} \rangle, \\ \langle \{ i(2)@r, u(1,2)@g, guide(2,3) \}, \emptyset, \{ snp(2) \} \rangle, \\ P_1 \}$$

6. By **Case 3.3**

$$PS = \{ \langle \{ \underline{i(2)@r}, u(1,2)@g, guide(2,3) \}, \emptyset, \{ snp(2) \} \rangle, \\ P_1 \}$$

7. By **Case 3.4**

*i(2)* is asked to the sensor *r*

$$PS = \{ \langle \{ \underline{u(1,2)@g}, guide(2,3) \}, \{ i(2)@r \}, \\ \{ snp(2) \} \rangle, P_1 \}$$

$$AAQ = \{ i(2)@r \}$$

8. By **Case 3.4**

*u(1,2)* is asked to the sensor *g*

$$PS = \{ \langle \{ \underline{guide(2,3)} \}, \{ i(2)@r, u(1,2)@g \}, \\ \{ snp(2) \} \rangle, P_1 \}$$

$$AAQ = \{ i(2)@r, u(1,2)@g \}$$

From the beginning to this iteration the system shows location 2 as the next destination to the user and, since the real value has not been retrieved from the information source, the system assumed that the user will correctly travel towards location 2. Continuing the current execution branch, the system will derivate the possibility of correctly traveling to location 3 and indicate this to the user, or the user may turn to location 3 and this is considered as a user mistake.

9. By **Case 3.1**

$$PS = \{ \langle \{ \underline{p(2,Y)}, snp(Y), u(2,Y)@g, guide(Y,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ \langle \{ p(2,Y), u(2,Y)@g, suw(Y), guide(Y,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, P_1 \}$$

10. By **Case 3.1**

$$PS = \{ \langle \{ \underline{snp(3)}, u(2,3)@g, guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ \langle \{ snp(4), u(2,4)@g, guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ \langle \{ u(2,3)@g, suw(3), guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ \langle \{ u(2,4)@g, suw(4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ P_1 \}$$

11. *u(1,2)* is returned from *g*

Nothing changes.

12. By **Case 3.2.3**

$$PS = \{ \langle \{ \underline{fail(\sim i(3)@r)}, u(2,3)@g, guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(3) \} \rangle, \\ P_2 (P_2 = \langle \{ snp(4), u(2,4), guide(4,3) \}, \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle), \\ P_3 (P_3 = \langle \{ u(2,3)@g, suw(3), guide(3,3) \}, \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle), \\ P_4 (P_4 = \langle \{ u(2,4)@g, suw(4), guide(4,3) \}, \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle), \\ P_1 \}$$

## 13. By Case 3.3.2

$$PS = \{ \langle \{ \underline{fail}(\emptyset), u(2,3)@g, guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(3) \} \rangle, \\ \langle \{ i(3)@r, u(2,3)@g, guide(3,3) \}, \{ i(2)@r, \\ u(1,2)@g \}, \{ snp(2), snp(3) \} \rangle, \\ P_2, P_3, P_4, P_1 \}$$

## 14. By Case 3.3

$$PS = \{ \langle \{ i(3)@r, u(2,3)@g, guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(3) \} \rangle, \\ P_2, P_3, P_4, P_1 \}$$

## 15. By Case 3.4

$i(3)$  is asked to the sensor  $r$

$$PS = \{ \langle \{ \underline{u(2,3)@g}, guide(3,3) \}, \\ \{ i(2)@r, u(1,2)@g, i(3)@r \}, \{ snp(2), snp(3) \} \rangle, \\ P_2, P_3, P_4, P_1 \}$$

$$AAQ = \{ i(2)@r, u(1,2)@g, i(3)@r \}$$

## 16. By Case 3.4

$u(2,3)$  is asked to the sensor  $g$

$$PS = \{ \langle \{ \underline{guide(3,3)} \}, \{ i(2)@r, u(1,2)@g, \\ i(3)@r, u(2,3)@g \}, \{ snp(2), snp(3) \} \rangle, \\ P_2, P_3, P_4, P_1 \}$$

$$AAQ = \{ i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g \}$$

## 17. By Case 3.1

$$PS = \{ \langle \emptyset, \{ i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g \}, \\ \{ snp(2), snp(3) \} \rangle, \\ \langle \{ \underline{snp(4)}, u(2,4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2) \} \rangle, \\ P_3, P_4, P_1 \}$$

At this iteration the current branch has reached the end, i.e., following this execution branch the user will correctly travel between location 1 and 3. Thus, after considering this as a possible solution to the guidance process, the system starts derivating a suspended branch.

## 18. By Case 3.2.3

$$PS = \{ \langle \{ \underline{fail}(\sim i(4)@r), u(2,4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(4) \} \rangle, \\ P_3, P_4, P_1, P_5 \{ P_5 = \langle \emptyset, \{ i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g \}, \{ snp(2), snp(3) \} \rangle \} \}$$

## 19. By Case 3.3.2

$$PS = \{ \langle \{ \underline{fail}(\emptyset), u(2,4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(4) \} \rangle, \\ \langle \{ i(4)@r, u(2,4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(4) \} \rangle, \\ P_3, P_4, P_1, P_5 \}$$

## 20. By Case 3.3

$$PS = \{ \langle \{ \underline{i(4)@r}, u(2,4), guide(4,3) \}, \\ \{ i(2)@r, u(1,2)@g \}, \{ snp(2), snp(4) \} \rangle, \\ P_3, P_4, P_1, P_5 \}$$

While waiting for the answer about the inclusion of location 4 as a valid travel point, the system assumes that it is not part of the included set and suspends this execution branch.

## 21. By Case 3.4

*i*(4) is asked to the sensor *r*

$$\begin{aligned}
 PS &= \{ \langle \{u(2,4), guide(4,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, i(4)@r\}, \{snp(2), snp(4)\} \rangle, \\
 &\quad \langle \{u(2,3)@g, suw(3), guide(3,3)\}, \\
 &\quad \{i(2)@r, u(1,2)@g\}, \{snp(2)\} \rangle, \\
 &\quad P_4, P_1, P_5 \} \\
 AAQ &= \{i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g, i(4)@r\}
 \end{aligned}$$

## 22.

*u*(2,3) has already been asked to the sensor *g*

$$\begin{aligned}
 PS &= \{ \langle \{suw(3), guide(3,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,3)@g\}, \{snp(2)\} \rangle, \\
 &\quad P_4, P_1, P_5, P_6 \ (P_6 = \langle \{u(2,4), guide(4,3)\}, \{i(2)@r, u(1,2)@g, i(4)@r\}, \{snp(2), snp(4)\} \rangle) \}
 \end{aligned}$$

## 23. By Case 3.2.3

$$\begin{aligned}
 PS &= \{ \langle \{fail(i(3)@r), guide(3,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,3)@g\}, \{snp(2), suw(3)\} \rangle, \\
 &\quad P_4, P_1, P_5, P_6 \}
 \end{aligned}$$

## 24. By Case 3.3.2

$$\begin{aligned}
 PS &= \{ \langle \{fail(\emptyset), guide(3,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,3)@g\}, \{snp(2), suw(3)\} \rangle, \\
 &\quad \langle \{\sim i(3)@r, guide(3,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,3)@g\}, \{snp(2), suw(3)\} \rangle, \\
 &\quad P_4, P_1, P_5, P_6 \}
 \end{aligned}$$

## 25. By Case 3.3

$$\begin{aligned}
 PS &= \{ \langle \{\sim i(3)@r, guide(3,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,3)@g\}, \{snp(2), suw(3)\} \rangle, \\
 &\quad P_4, P_1, P_5, P_6 \}
 \end{aligned}$$

Since location 3 is in the set of the valid points to travel, the current execution branch is suspended since it is contrary to the real value.

## 26.

*i*(3) has already been asked to the sensor *r*

$$\begin{aligned}
 PS &= \{ \langle \{guide(3,3)\}, \{i(2)@r, u(1,2)@g, \\
 &\quad u(2,3)@g, \sim i(3)@r\}, \{snp(2), suw(3)\} \rangle, \\
 &\quad \langle \{u(2,4)@g, suw(4), guide(4,3)\}, \\
 &\quad \{i(2)@r, u(1,2)@g\}, \{snp(2)\} \rangle, P_1, P_5, P_6 \}
 \end{aligned}$$

## 27. By Case 3.4

*u*(2,4) is asked to the sensor *g*

$$\begin{aligned}
 PS &= \{ \langle \{suw(4), guide(4,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,4)@g\}, \{snp(2)\} \rangle, \\
 &\quad P_1, P_5, P_6, P_7 \ (P_7 = \langle \{guide(3,3)\}, \{i(2)@r, u(1,2)@g, u(2,3)@g, \sim i(3)@r\}, \{snp(2), suw(3)\} \rangle) \} \\
 AAQ &= \{i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g, \\
 &\quad i(4)@r, u(2,4)@g\}
 \end{aligned}$$

...

## 28. By Case 3.2.3

$$\begin{aligned}
 PS &= \{ \langle \{fail(i(4)@r, guide(4,3)\}, \{i(2)@r, \\
 &\quad u(1,2)@g, u(2,4)@g\}, \{snp(2), suw(4)\} \rangle, \\
 &\quad P_1, P_5, P_6, P_7 \}
 \end{aligned}$$

## 29. By Case 3.3.2

$$PS = \{ \langle \{ \underline{fail}(\emptyset), guide(4,3) \}, \{ i(2)@r, u(1,2)@g, u(2,4)@g \}, \{ snp(2), suw(4) \} \rangle, \langle \{ \sim i(4)@r, guide(4,3) \}, \{ i(2)@r, u(1,2)@g, u(2,4)@g \}, \{ snp(2), suw(4) \} \rangle, P_1, P_5, P_6, P_7 \}$$

## 30. By Case 3.3

$$PS = \{ \langle \{ \sim i(4)@r, guide(4,3) \}, \{ i(2)@r, u(1,2)@g, u(2,4)@g \}, \{ snp(2), suw(4) \} \rangle, P_1, P_5, P_6, P_7 \}$$

31.

*i(4)* has already been asked to the sensor *r*

$$PS = \{ \langle \{ guide(4,3) \}, \{ i(2)@r, u(1,2)@g, u(2,4)@g, \sim i(4)@r \}, \{ snp(2), suw(4) \} \rangle, P_1, P_5, P_6, P_7 \}$$

32. *u(2,3)* is returned from *g*

Nothing changes.

33.  $\sim u(2,4)$  is returned from *g*

By Fact Arrival Phase:

$$CBS = \{ u(1,2)@g, u(2,3)@g, \sim u(2,4)@g, u(4,5)@g, u(5,6)@g, u(6,3)@g, i(1)@r, i(2)@r, i(3)@r, \sim i(4)@r, i(5)@r, i(6)@r \}$$

At this iteration all branches have been derived to their maximum considering the real values returned from the information sources or the ones in the initial default set. Thus, the system is able to return the outside defaults and inside assumptions which, for this execution example, is to indicate the user to travel to location 2 and then to location 3.

34.

$\{ i(2)@r, u(1,2)@g, i(3)@r, u(2,3)@g \}$  and  $\{ snp(2), snp(3) \}$  are returned as outside defaults and inside assumptions.

## References

1. American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders: DSM-IV*, 4th ed.; American Psychiatric Association: Washington, DC, USA, 1994; p. 915.
2. Schalock, R.L.; Borthwick-Duffy, S.A.; Bradley, V.J.; Buntinx, W.H.E.; Coulter, D.L.; Craig, E.M.; Gomez, S.C.; Lachapelle, Y.; Luckasson, R.; Reeve, A.; et al. *Intellectual Disability: Definition, Classification, and Systems of Supports*, 11th ed.; American Association on Intellectual and Developmental Disabilities: Washington, DC, USA, 2010; p. 259.
3. WebAIM—Web Accessibility In Mind. *Cognitive Disabilities*; WebAIM: Logan, UT, USA, 2007. Available online: <http://www.webaim.org/articles/cognitive/> (accessed on 15 June 2016).
4. Dawe, M. Desperately seeking simplicity: How young adults with cognitive disabilities and their families adopt assistive technologies. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal, QC, Canada, 22–27 April 2006; ACM: New York, NY, USA, 2006; pp. 1143–1152.
5. American Psychiatric Association. *Diagnostic and Statistical Manual of Mental Disorders (DSM-5)*, 5th ed.; American Psychiatric Publishing: Washington, DC, USA, 2013; p. 991.
6. Woods, B.; Aguirre, E.; Spector, A.E.; Orrell, M. Cognitive stimulation to improve cognitive functioning in people with dementia. *Cochrane Database Syst. Rev.* **2012**, 2. [CrossRef] [PubMed]

7. United Nations—Division for Social Policy and Development Disability. *United Nations Convention on the Rights of Person's with Disabilities*; United Nations: New York, USA, 2006.
8. Sadri, F. Multi-Agent Ambient Intelligence for Elderly Care and Assistance. In Proceedings of the AIP Conference, Vancouver, BC, Canada, 25–29 June 2007; Volume 2007, pp. 117–120.
9. *AAL Joint Programme Call AAL-2010-3 Project no. AAL 2010-3-012*; Mylife: Losangeles, CA, USA, 2012.
10. Lanyi, C.S.; Brown, D.J. Design of Serious Games for Students with Intellectual Disability. In Proceedings of the International Conference on Interaction Design International Development, Mumbai, India, 20–24 March 2010; Joshi, A., Dearden, A., Eds.; British Computer Society: Swinton, UK, 2010; pp. 44–54.
11. Liu, A.L.; Hile, H.; Kautz, H.; Borriello, G.; Brown, P.A.; Harniss, M.; Johnson, K. Indoor Wayfinding: Developing a Functional Interface for Individuals with Cognitive Impairments. In Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS), Portland, OR, USA, 23–25 October 2006.
12. Liu, A.L.; Hile, H.; Kautz, H.; Borriello, G.; Brown, P.A.; Harniss, M.; Johnson, K. Indoor wayfinding: Developing a functional interface for individuals with cognitive impairments. *Disabil. Rehabil. Assist. Technol.* **2008**, *3*, 69–81. [[CrossRef](#)] [[PubMed](#)]
13. Carmien, S.; Dawe, M.; Fischer, G.; Gorman, A.; Kintsch, A.; Sullivan, J.F. Socio-technical environments supporting people with cognitive disabilities using public transportation. *ACM Trans. Comput.-Hum. Interact.* **2005**, *12*, 233–262. [[CrossRef](#)]
14. Dawe, M. Let Me Show You What I Want: Engaging Individuals with Cognitive Disabilities and their Families in Design. In Proceedings of the Extended Abstracts on Human Factors in Computing Systems, San Jose, CA, USA, 28 April–3 May 2007; pp. 2177–2182.
15. Liu, A.L.; Hile, H.; Borriello, G.; Kautz, H.; Brown, P.A.; Harniss, M.; Johnson, K. Informing the Design of an Automated Wayfinding System for Individuals with Cognitive Impairments. In Proceedings of the 3rd International Conference on Pervasive Computing Technologies for Healthcare, London, UK, 1–3 April 2009; Volume 9, p. 8.
16. Ramos, J.; Anacleto, R.; Costa, A.; Novais, P.; Figueiredo, L.; Almeida, A. Orientation System for People with Cognitive Disabilities. In *Ambient Intelligence—Software and Applications*; Novais, P., Hallenborg, K., Tapia, D.I., Rodríguez, J.M.C., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 153, pp. 43–50.
17. Ramos, J.; Anacleto, R.; Novais, P.; Figueiredo, L.; Almeida, A.; Neves, J. Geo-localization System for People with Cognitive Disabilities. In *Trends in Practical Applications of Agents and Multiagent Systems*; Pérez, J.B., Hermoso, R., Moreno, M.N., Rodríguez, J.M.C., Hirsch, B., Mathieu, P., Campbell, A., Suarez-Figueroa, M.C., Ortega, A., Adam, E., et al., Eds.; Springer: Cham, Switzerland, 2013; Volume 221, pp. 59–66.
18. Fraunhofer Portugal. AlzNav. Available online: <http://www.fraunhofer.pt/en/fraunhoferaicos/-projects/internalresearch/alznav.html> (accessed on 10 December 2012).
19. Patterson, D.J.; Liao, L.; Fox, D.; Kautz, H. Inferring High-Level Behavior from Low-Level Sensors. In *International Conference on Ubiquitous Computing*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2864, pp. 73–89.
20. Sadilek, A.; Kautz, H. Recognizing Multi-Agent Activities from GPS Data. *Artif. Intell.* **2010**, *39*, 1134–1139.
21. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall Series In Artificial Intelligence; Chapter Probabilis; Prentice Hall: Upper NJ River, NJ, USA, 2003; Volume 60, pp. 269–272.
22. Hertzberg, J.; Chatila, R. AI Reasoning Methods for Robotics. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Chapter 10, pp. 207–223.
23. Dahan, H.; Cohen, S.; Rokach, L.; Maimon, O. *Proactive Data Mining with Decision Trees*; Springer Science & Business Media: Berlin, Germany, 2014; p. 94.
24. Rokach, L.; Maimon, O. Decision Trees. In *Data Mining and Knowledge Discovery Handbook*; Springer US: Boston, MA, USA, 2005; pp. 165–192.
25. Basheer, I.A.; Hajmeer, M. Artificial neural networks: Fundamentals, computing, design, and application. *J. Microbiol. Methods* **2000**, *43*, 3–31. [[CrossRef](#)]
26. Jain, A.K.; Mao, J.C.; Mohiuddin, K.M. Artificial neural networks: A tutorial. *Computer* **1996**, *29*, 31–44. [[CrossRef](#)]
27. Noble, W.S. What is a support vector machine? *Nat. Biotechnol.* **2006**, *24*, 1565–1567. [[CrossRef](#)] [[PubMed](#)]
28. Bennett, K.P.; Campbell, C. Support vector machines: Hype or hallelujah? *ACM SIGKDD Explor. Newslett.* **2000**, *2*, 1–13. [[CrossRef](#)]

29. Uusitalo, L. Advantages and challenges of Bayesian networks in environmental modelling. *Ecol. Model.* **2007**, *203*, 312–318. [[CrossRef](#)]
30. Liu, A.L.; Hile, H.; Borriello, G.; Brown, P.A.; Harniss, M.; Kautz, H.; Johnson, K. Customizing directions in an automated wayfinding system for individuals with cognitive impairment. In Proceedings of the Eleventh International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS), Pittsburgh, PA, USA, 25–28 October 2009; p. 27.
31. Kolodner, J.L. An introduction to case-based reasoning. *Artif. Intell. Rev.* **1992**, *6*, 3–34. [[CrossRef](#)]
32. Satoh, K. Speculative Computation and Abduction for an Autonomous Agent. *IEICE Trans. Inf. Syst.* **2005**, *88*, 2031–2038. [[CrossRef](#)]
33. Kakas, A.C.; Kowalski, R.A.; Toni, F. The Role of Abduction in Logic Programming. *Handb. Log. Artif. Intell. Log. Progr.* **1998**, *5*, 235–324.
34. Jain, P. Wireless Body Area Network for Medical Healthcare. *IETE Tech. Rev.* **2011**, *28*, 362. [[CrossRef](#)]
35. Montón, E.; Hernandez, J.; Blasco, J.; Hervé, T.; Micallef, J.; Grech, I.; Brincat, A.; Traver, V. Body area network for wireless patient monitoring. *IET Commun.* **2008**, *2*, 215–222. [[CrossRef](#)]
36. Wolf, L.; Saadaoui, S. Architecture Concept of a Wireless Body Area Sensor Network for Health Monitoring of Elderly People. In Proceedings of the 4th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 11–13 January 2007; pp. 722–726.
37. Shamoo, A.E.; Resnik, D.B. *Responsible Conduct of Research*, 3rd ed.; Oxford University Press: Oxford, UK, 2015; p. 360.
38. Yip, C.; Han, N.L.; Sng, B. Legal and ethical issues in research. *Indian J. Anaesth.* **2016**, *60*, 684–688. [[PubMed](#)]
39. Kakas, A.C.; Mancarella, P. On the relation between Truth Maintenance and Abduction. In Proceedings of the First Pacific Rim International Conference on Artificial Intelligence (PRICAI), Nagoya, Japan, 1990; pp. 438–443.
40. Karich, P.; Schroder, S. GraphHopper Directions API with Route Optimization. Available online: <https://graphhopper.com> (accessed on 20 June 2016).
41. Ramos, J.; Oliveira, T.; Satoh, K.; Neves, J.; Novais, P. An Orientation Method with Prediction and Anticipation Features. *Iberoam. J. Artif. Intell.* **2017**, *20*, 82–95. [[CrossRef](#)]
42. Pei, J.; Han, J.; Mortazavi-Asl, B.; Wang, J.; Pinto, H.; Chen, Q.; Dayal, U.; Hsu, M.C. Mining sequential patterns by pattern-growth: The PrefixSpan approach. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1424–1440.
43. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. Optics: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [[CrossRef](#)]
44. Fournier-Viger, P. SPMF—Open-Source Data Mining Library. Available online: <http://www.philippe-fournier-viger.com/spmf/index.php> (accessed on 30 June 2016).



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).