Location-Aware JADE Agents in Indoor Scenarios

Stefania Monica, Federico Bergenti
Dipartimento di Matematica e Informatica
Parco Area delle Scienze 53/A, 43124 Parma, Italy
Email: {stefania.monica, federico.bergenti}@unipr.it

Abstract—**This paper presents a novel JADE add-on that enables the implementation of location-aware agents by interfacing an underlying ranging technology which provides accurate distance measures both indoor and outdoor. First, the paper motivates the work and it presents the features and the architecture of the add-on. Then, the paper provides a detailed description of the implemented localization algorithms and it validates them in an indoor scenario. The experimental results show that accurate indoor localization can be achieved and that the presented add-on can be used to support location-aware agents with sufficient accuracy for targeted educational and ludic applications.**

I. Introduction

The use of software agents on mobile devices dates back to first cellular telephone prototypes capable of running Java applications, and JADE was one of the first tools that enabled FIPA agents on devices that were then called Java-enabled phones [1]. We used to call them *nomadic agents* [2] at the time—to make a clear distinction with then popular mobile agents—and they were considered one of the most promising applications of agent technology. More recently, the porting of JADE to Android devices [3], and the widespread adoption of JADE and related technologies in crucial parts of large-scale networks [4], revitalized the idea of having FIPA agents on mobile appliances to support large communities of users in their daily activities, which include both collaborative (see, e.g., [5], [6]) and competitive tasks (see, e.g., [7], [8]). Notably, the appliances of today offer much more resources than first Java-enabled phones, and the challenge of implementing agents for mobile devices is no longer about fitting complex software into constrained-resource devices; rather, it is about interfacing agents with the physical world they live in to ensure that user can be provided with contextualized services. This paper tackles the problem of interfacing agents with the physical world by presenting a novel software module which can be used to develop JADE agents that can sense the presence of nearby localization beacons or agents, both in outdoor and indoor scenarios. We prototyped such a module as a conventional JADE add-on which uses *Ultra Wide Band* (*UWB*) signals (see, e.g., [9]) to measure the distances between the appliance where the agent is running and target beacons or other appliances.

The acronym UWB was first used by the US Department of Defense in late '80s and it became popular after the Federal Communications Commission allowed the unlicensed use of UWB devices in February 2002 under specific emission constraints [10]. In 2004, IEEE established standardization group IEEE 802.15.4a with the aim of defining a new physical layer for the already existing IEEE 802.15.4 standard for *Wireless Personal Area Networks* (*WPANs*). In 2007 the new IEEE 802.15.4a standard was finally completed and since then it provides a standardized physical layer for short-range, low data rate communications, and for high-precision ranging using low-power devices [11].

The use of UWB signals is particularly promising for high-precision localization because it ensures high ranging accuracy. As a matter of fact, due to their large bandwidth, UWB signals are characterized by very short duration pulses—usually in the order of one nanosecond—which guarantee accurate *Time of Flight* (*ToF*) estimation for signals traveling between nodes. This implies that the distance between a transmitter and a receiver can be accurately determined, yielding high ranging accuracy. At the opposite, pulses received via multiple paths using conventional narrow-band signals can easily overlap, causing wrong ToF estimates, hence wrong range estimates [11]. Besides their short pulses, UWB signals are also characterized by low duty cycle which leads to low energy consumption. Moreover, since UWB signals occupy a large portion of the spectrum, in order to avoid interference problems with other devices operating in the same frequency spectrum, UWB systems normally use low-power transmissions. Finally, UWB signals are characterized by their capability of penetrating through obstacles thanks to the large frequency spectrum that characterizes them (which includes low-frequency components as well as high-frequency ones) [12]. Such a feature is particularly interesting in indoor environments where the presence of walls and objects can cause *Non-Line-of-Sight* (*NLoS*) effects between sensors.

Such unique aspects make UWB technology a good candidate for accurate and low-power positioning systems. One of the main drawbacks that caused the slow adoption of UWB technology for accurate indoor localization was that UWB transceivers were normally very expensive because of the intrinsic challenges that their design and construction involve, which also include high frequency logics for measuring very short delays. Only recently, in 2013, a company named *BeSpoon* (www.bespoon.com) started producing add-on modules for smartphones integrating an UWB transceiver and an antenna at a price compatible with the consumers' demands. They also provide a smartphone, called *spoonphone*, which natively accommodates the UWB module and which provides needed drivers for Android. This makes the development of accurate ranging techniques attractive also for general-purpose technologies, like JADE, which are not intended to tackle the issues of interfacing with proprietary, and expensive, hardware.

This paper is organized as follows. Section II describes the architecture of the add-on and it details the implemented localization algorithms. Section III describes the experimental campaign that we performed to validate the usability of the proposed approach. Section IV concludes the paper.

II. Distance- and Position-Aware JADE Agents

The presented add-on for JADE uses the tools and techniques that JADE provides to integrate an underlying ranging technology, like UWB signaling, with the common agent loop. In details, it provides two customizable JADE behaviours that can be used to access the services of the add-on.

The first behaviour, called `RangeChangeBehaviour`, is in charge of connecting with the underlying ranging technology to measure the distances with targeted beacons or smart appliances. The behaviour can accommodate any ranging technology, provided that an implementation of a specific interface is available. In the presented prototype we opt for an implementation that we developed using BeSpoon APIs which acquires real-time information from the UWB transceiver installed in a spoonphone. The `RangeChangeBehaviour` is scheduled when an above-threshold change in the distances from selected beacons or smart appliances occurs, or when the agent decides to change the set of devices that it monitors. In details, the implemented prototype uses hardware-specific unique identifiers associated with transceivers that allow targeting beacons and smart appliances indifferently. This way the agent developer is free to track beacons and smart appliances with a single, generic mechanism.

The second behaviour that ships with the add-on is called `LocationChangeBehaviour` and it uses the first behaviour to acquire needed ranging information to inform an agent about its current location in a predefined reference frame. The behaviour can be configured to acquire needed information for performing localization from a set of beacons, commonly known as *Anchor Nodes* (*ANs*), and its main duty is to invoke a pluggable localization algorithm to estimate the location of the agent every time the distances from ANs change significantly. The developer is free to implement custom localization algorithms but the add-on contains two general-purpose algorithms that can be used if no additional information besides the distance from a fixed number of ANs is available. Such algorithms are discussed in details below.

For the sake of simplicity, the implemented localization algorithms currently consider only a bi-dimensional scenario, i.e., they assume that all the nodes lay on the same plane. The same algorithms can be extended to the case of a three-dimensional scenario (e.g., see [13]). Both implemented localization algorithms are based on the ToF between some nodes with known positions, the ANs, and the *Target Node* (*TN*), whose position is to be estimated.

Denoting as M the number of ANs, we indicate the coordinates of the ANs as

$$\underline{s}_i \;=\; [x_i, y_i]^T \qquad i \in \{1, \ldots, M\}. \tag{1}$$

The (unknown) TN position and its estimate are denoted as $\underline{u} = [x, y]^T$ and $\hat{\underline{u}} = [\hat{x}, \hat{y}]^T$, respectively. Moreover, the true and the estimated distances between the $i-$th AN and the TN are denoted as

$$\begin{aligned} r_i &\triangleq ||\underline{u} - \underline{s}_i|| = \sqrt{(\underline{u} - \underline{s}_i)^T(\underline{u} - \underline{s}_i)} \qquad i \in \{1, \ldots, M\} \\ \hat{r}_i &\triangleq ||\hat{\underline{u}} - \underline{s}_i|| = \sqrt{(\hat{\underline{u}} - \underline{s}_i)^T(\hat{\underline{u}} - \underline{s}_i)} \qquad i \in \{1, \ldots, M\}. \end{aligned}$$

Observe that if the coordinates of the ANs and the true distances $\{r_i\}_{i=1}^M$ are known, the true position of the TN can be found according to simple geometric considerations. As a matter of fact, the TN lies on each of the circumferences $\{\mathcal{C}_i\}_{i=1}^M$, centered in $\{\underline{s}_i\}_{i=1}^M$ with radii $\{r_i\}_{i=1}^M$ and, therefore, its coordinates satisfy the following system of equations

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = r_1^2 \\ \cdots \\ (x - x_M)^2 + (y - y_M)^2 = r_M^2. \end{cases} \tag{2}$$

Since the true distances are unknown, we can only rely on their estimates $\{\hat{r}_i\}_{i=1}^M$. The circumferences $\{\hat{\mathcal{C}}_i\}_{i=1}^M$ centered in $\{\underline{s}_i\}_{i=1}^M$ with radii $\{\hat{r}_i\}_{i=1}^M$ lead to the system of equations

$$\begin{cases} (x - x_1)^2 + (y - y_1)^2 = \hat{r}_1^2 \\ \cdots \\ (x - x_M)^2 + (y - y_M)^2 = \hat{r}_M^2. \end{cases} \tag{3}$$

Obviously, due to the errors that affect the range estimates $\{\hat{r}_i\}_{i=1}^M$, the circumferences $\{\hat{\mathcal{C}}_i\}_{i=1}^M$ would hardly intersect in a unique point, hence proper localization algorithms need to be used. For such algorithms we assume that the errors which affect the range measurements $\{\hat{r}_i\}_{i=1}^M$ can be modeled as independent additive random variables $\{\nu_i\}_{i=1}^M$, namely

$$\hat{r}_i = r_i + \nu_i \qquad i \in \{1, \ldots, M\}. \tag{4}$$

Let us define as $\underline{\nu}$ the vector whose $i-$th element is ν_i and let us denote as $\underline{\underline{Q}}$ its covariance matrix.

Many range-based localization algorithms have been proposed in the literature, and they can be broadly classified into *passive* and *active* [14]. Passive localization relies on the fact that wireless communications strongly depend on the environment and it is based on the analysis of the scattering caused by obstacles found along signal propagation and/or of the variance of a measured signal. Such analysis allows finding changes in the received signals that can be used to detect and locate targets [15].

In active techniques, instead, all nodes are equipped with sensors and with an electronic device which sends needed information to support a proper localization algorithm. Since all implemented algorithms follow in this category, we focus on range-based localization with active tags, which can be based on the ToF, the *Angle of Arrival* (*AoA*), or the *Received Signal Strength* (*RSS*) of the signals [10]. As explained in the introduction, we are mostly interested in ToF-based localization algorithms because are particularly well suited when dealing with UWB signaling. In details, if two synchronized nodes communicate, the node receiving the signal can determine the *Time of Arrival* (*ToA*) of the incoming signal from the timestamp of the sending node. If the nodes are not synchronized, *Time Difference of Arrival* (*TDoA*) techniques can be employed, which are based on the estimation of the difference between the arrival times of UWB signals traveling between the TN and ANs. A large number of ToF-based localization techniques have been proposed in the literature. Among them, it is worth mentioning iterative methods [16], graph-based methods [17], closed-form algorithms [18] and optimization methods [19]. Observe that the accuracy of some of such algorithms may strongly depend on the number of ANs [20] and on their topology [21].

In the following subsections, the two implemented localization algorithms are described in details, namely: the *Circumference Intersection* (*CI*) algorithm and the *Two-Stage Maximum-Likelihood* (*TSML*) algorithm. From now on, we assume that $M = 3$ ANs are used to locate the TN.

A. Circumference Intersection (CI) algorithm

This subsection describes the CI localization algorithm, which is a very intuitive algorithm that can be considered as the basis of other more elaborate algorithms (see, e.g. [22]).

In order to better explain the CI algorithm, let us make a few geometric considerations on the considered problem. As already observed, the position of the TN coincides with the (unique) intersection of the three circumferences in (2). Since such circumferences are unknown, in order to find the TN position estimate $\underline{\hat{u}} = [\hat{x}, \hat{y}]$ it is necessary to consider proper localization approaches based on the set of equations (3). According to the CI algorithm, since the circumferences do not intersect in a single point, we intersect pairs of them.

More precisely, we define the following three sets—each of which contains two points—obtained by intersecting the three different pairs of circumferences

$$I_1 = \{\underline{p}_{12}, \underline{q}_{12}\} \triangleq \hat{\mathcal{C}}_1 \cap \hat{\mathcal{C}}_2 \tag{5}$$

$$I_2 = \{\underline{p}_{13}, \underline{q}_{13}\} \triangleq \hat{\mathcal{C}}_1 \cap \hat{\mathcal{C}}_3 \tag{6}$$

$$I_3 = \{\underline{p}_{23}, \underline{q}_{23}\} \triangleq \hat{\mathcal{C}}_2 \cap \hat{\mathcal{C}}_3 . \tag{7}$$

We then choose a point from each of the three sets, namely $\underline{p}_1 \in I_1$, $\underline{p}_2 \in I_2$, and $\underline{p}_3 \in I_3$, so that the three selected points are the nearest ones to each other. More precisely, since such points belong to circumference intersections, it can be shown that they can be chosen as

$$||\underline{p}_1 - \underline{p}_2|| = \min_{\underline{p} \in I_1, \underline{q} \in I_2} ||\underline{p} - \underline{q}|| \tag{8}$$

$$||\underline{p}_1 - \underline{p}_3|| = \min_{\underline{q} \in I_3} ||\underline{p}_1 - \underline{q}|| . \tag{9}$$

Given these three points, the TN position estimate is chosen as their baricenter.

We remark that the intersection between two circumferences can be empty. Assume, for instance, that in (5) the set I_1 is empty, namely the circumferences $\hat{\mathcal{C}}_1$ and $\hat{\mathcal{C}}_2$ do not intersect. In this case, the two nearest points of $\hat{\mathcal{C}}_1$ and $\hat{\mathcal{C}}_2$ are found, and if their distance is below a given threshold, they are considered the two intersection points and the set I_1 is redefined as the set containing such two points. Otherwise, the TN position estimate is found based on the remaining intersections, whenever possible.

B. Two-Stage Maximum-Likelihood (TSML) algorithm

This subsection describes a TSML method that uses ToA information, hence known as TSLM-ToA method [23]. Such method is a well-known localization algorithm, and it is particularly interesting because it was shown that it can attain the Cramer-Rao Lower Bound, which is a lower bound for the variance of an estimator [24]. A detailed derivation of this method can be found in [25].

The starting point of the TSML algorithm is once again the quadratic system (3), where we set $M = 3$ as in the case of the CI algorithm. In order to solve it, a two-step approach based on *Maximum-Likelihood* (*ML*) technique can be considered. First, let us define as a_i the Euclidean norm of the coordinate vector of the $i-$th AN, namely

$$a_i \triangleq ||\underline{s}_i|| = \sqrt{x_i^2 + y_i^2} \qquad i \in \{1, \ldots, 3\}. \tag{10}$$

Moreover, let us introduce the new variable

$$\hat{n} \triangleq ||\underline{\hat{u}}||^2 = \hat{x}^2 + \hat{y}^2 \tag{11}$$

so that system (3) can be rewritten, in matrix notation, as

$$\underline{\underline{G}}_1 \, \underline{\hat{\omega}}_1 = \underline{\hat{h}}_1 \tag{12}$$

where

$$\begin{gathered} \underline{\underline{G}}_1 = -2 \begin{pmatrix} x_1 & y_1 & -0.5 \\ \vdots & \vdots & \vdots \\ x_3 & y_3 & -0.5 \end{pmatrix} \\ \underline{\hat{\omega}}_1 = \begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{n} \end{pmatrix} \qquad \underline{\hat{h}}_1 = \begin{pmatrix} \hat{r}_1^2 - a_1^2 \\ \vdots \\ \hat{r}_3^2 - a_3^2 \end{pmatrix} . \end{gathered} \tag{13}$$

Let us also introduce $\underline{\omega}_1$ and $\underline{h}_1$ the vectors analogous to $\underline{\hat{\omega}}_1$ and to $\underline{\hat{h}}_1$, respectively, where estimated quantities are substituted by true ones, namely

$$\underline{\omega}_1 = \begin{pmatrix} x \\ y \\ n \end{pmatrix} \qquad \underline{h}_1 = \begin{pmatrix} r_1^2 - a_1^2 \\ \vdots \\ r_3^2 - a_3^2 \end{pmatrix} . \tag{14}$$

While (12) might look like a linear system, it is not, since the third element of the solution vector $\underline{\hat{\omega}}_1$ depends on the first two according to (11). The solution $\underline{\hat{\omega}}_1$ of the system (12) can be determined through an ML approach. In particular, as suggested in [18], let us define the error vector

$$\underline{\psi}_1 \triangleq \underline{\hat{h}}_1 - \underline{\underline{G}}_1 \underline{\omega}_1 . \tag{15}$$

Given a positive definite matrix $\underline{\underline{W}}_1$, the weighted *Least Square* (*LS*) solution of (12) that minimizes $\underline{\psi}_1^T \underline{\underline{W}}_1 \underline{\psi}_1$ is

$$\underline{\hat{\omega}}_1 = (\underline{\underline{G}}_1^T \, \underline{\underline{W}}_1 \, \underline{\underline{G}}_1)^{-1} \underline{\underline{G}}_1^T \, \underline{\underline{W}}_1 \, \underline{\hat{h}}_1 . \tag{16}$$

The simplest choice of the weighting matrix $\underline{\underline{W}}_1$ is the identity matrix. In [25] it is shown that the choice of $\underline{\underline{W}}_1$ which minimizes the variance of $\underline{\hat{\omega}}_1$ is

$$\underline{\underline{W}}_1 \triangleq \mathrm{Cov}[\underline{\psi}_1]^{-1} = (4 \underline{\underline{B}} \, \underline{\underline{Q}} \, \underline{\underline{B}})^{-1} \tag{17}$$

where $\underline{\underline{Q}}$ is the covariance matrix of the ToA range measurements $\{r_i\}_{i=1}^M$, $\underline{\underline{B}}$ is a diagonal matrix whose elements are $\{r_i\}_{i=1}^M$, and the last equality follows from the fact that, from (4) and (13), $\underline{\psi}_1$ can be written as

$$\underline{\psi}_1 = \underline{\hat{h}}_1 - \underline{h}_1 = 2 \underline{\underline{B}} \, \underline{\nu} + \underline{\nu} \odot \underline{\nu} \simeq 2 \underline{\underline{B}} \, \underline{\nu} \tag{18}$$

where $\odot$ denotes the entrywise product and the last approximation is obtained neglecting second order perturbations. With this choice of the weighting matrix one obtains that $\mathrm{Cov}[\underline{\hat{\omega}}_1] = (\underline{\underline{G}}_1^T \, \underline{\underline{W}}_1 \, \underline{\underline{G}}_1)^{-1}$.

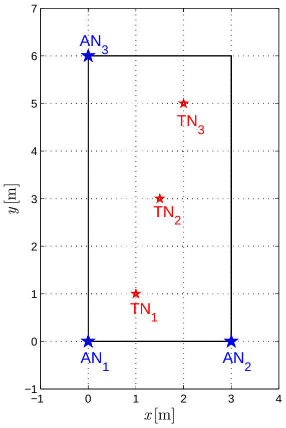
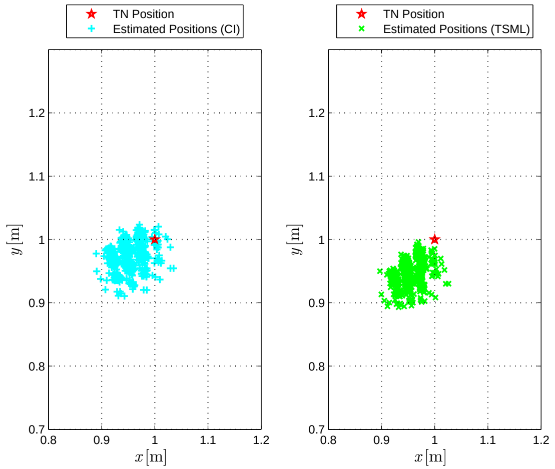


Fig. 1. The positions of the three considered ANs are shown (blue stars), together with the three different TN positions (red stars).



(a)

(b)

Fig. 2. The 1000 position estimates of TN$_1$ obtained using: (a) the CI algorithm; and (b) the TSML algorithm.

The second stage of the algorithm is meant to take into account the dependence of $\hat{n}$ on the other elements of the unknown vector in the system of equations (12) and it involves the solution of the system

$$\underline{\underline{G}}_2 \, \hat{\underline{\omega}}_2 = \hat{\underline{h}}_2 \tag{19}$$

where $\hat{\underline{\omega}}_2 = (\hat{x}^2, \hat{y}^2)^T$ and

$$\underline{\underline{G}}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad \hat{\underline{h}}_2 = \begin{pmatrix} [\hat{\underline{\omega}}_1]_1^2 \\ [\hat{\underline{\omega}}_1]_2^2 \\ [\hat{\underline{\omega}}_1]_3 \end{pmatrix} \tag{20}$$

with $[\hat{\underline{\omega}}_1]_j$ denoting the $j-$th component of $\hat{\underline{\omega}}_1$. The linear system (19) can be solved, once again, through the ML technique. Defining the error vector

$$\underline{\psi}_2 \triangleq \hat{\underline{h}}_2 - \underline{\underline{G}}_2 \, \underline{\omega}_2 \tag{21}$$

the weighted LS solution of (19) that minimizes the weighted norm of $\underline{\psi}_2$ with a positive definite matrix $\underline{\underline{W}}_2$ is

$$\hat{\underline{\omega}}_2 = (\underline{\underline{G}}_2^T \, \underline{\underline{W}}_2 \, \underline{\underline{G}}_2)^{-1} \underline{\underline{G}}_2^T \, \underline{\underline{W}}_2 \, \hat{\underline{h}}_2. \tag{22}$$

As considered to solve (12), the simplest choice of the weighting matrix $\underline{\underline{W}}_2$ is the identity matrix. In [25], it is shown that the choice of $\underline{\underline{W}}_2$ which minimizes the variance of $\hat{\underline{\omega}}_2$ is

$$\underline{\underline{W}}_2 \triangleq \mathrm{Cov}[\underline{\psi}_2]^{-1} = (4\underline{\underline{B}}_2 \, \mathrm{Cov}[\hat{\underline{\omega}}_1] \, \underline{\underline{B}}_2)^{-1} \tag{23}$$

where $\underline{\underline{B}}_2 = \mathrm{diag}(x, y, 0.5)$. Finally, the position estimate can be expressed as

$$\hat{\underline{u}} = \underline{\underline{U}} \left[\sqrt{[\hat{\underline{\omega}}_2]_1}, \sqrt{[\hat{\underline{\omega}}_2]_2} \right]^T \tag{24}$$

where $\underline{\underline{U}} = \mathrm{diag}(\mathrm{sign}(\hat{\underline{\omega}}_1))$.

III. Experimental Setup and Results

This section shows localization results obtained with the two implemented algorithms, as described in the previous section, using a JADE agent running on a spoonphone used as TN. We use three localization beacons as ANs and we put them at three corners of a rectangular room whose sides measures 3 m and 6 m. Using the same notation introduced in Section II, the ANs position in the experimental setup are denoted as

$$\underline{s}_1 = [0, 0]^T \qquad \underline{s}_2 = [3, 0]^T \qquad \underline{s}_3 = [0, 6]^T \tag{25}$$

where the coordinates are expressed in meters. In Fig. 1 the ANs positions are shown (blue stars) in the considered room.

Inside the same room, we consider three different TN positions, denoted as red stars in Fig. 1. First, we put the TN in the points whose coordinates, expressed in meters, are

$$\underline{u}_1 = [1, 1]^T. \tag{26}$$

Observe that this point, denoted as TN$_1$ in Fig. 1, is close to one of the corners of the room and close to AN$_1$. The second TN position we consider is

$$\underline{u}_2 = [1.5, 3]^T \tag{27}$$

which corresponds to the point in the middle of the room, denoted as TN$_2$ in Fig. 1. Finally, the coordinates of the last TN position, expressed in meters, are

$$\underline{u}_3 = [2, 5]^T. \tag{28}$$

Observe that this point is symmetric of $\underline{u}_1$ with respect to the center of the room.

For each TN position $\{\mathrm{TN}_i\}_{i=1}^3$, we first acquire the three distance estimates between the TN and the three ANs and we use such distances to feed the two localization algorithms described in Section II. This process is iterated $N_I = 1000$ times, thus obtaining 1000 position estimates for each of the two localization algorithms and for each TN position. For each iteration $j \in \{1, \ldots, N_I\}$ we define the distance between the true TN position and its estimate in the j-th iteration as

$$d_j = ||\hat{\underline{u}}_j - \underline{u}|| \tag{29}$$

where $\hat{\underline{u}}_j$ is the TN position estimate in the j-th iteration.

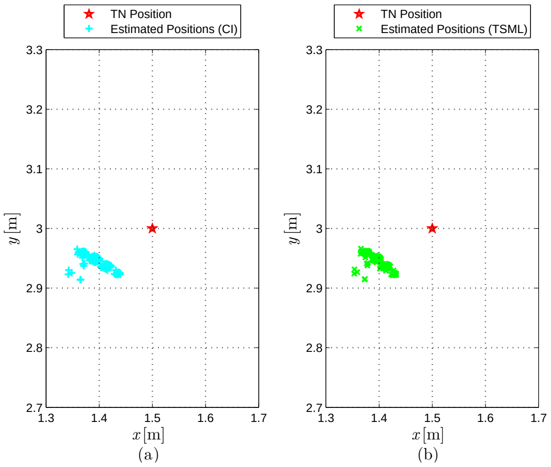


Fig. 3. The 1000 position estimates of TN_2 obtained using: (a) the CI algorithm; and (b) the TSML algorithm.

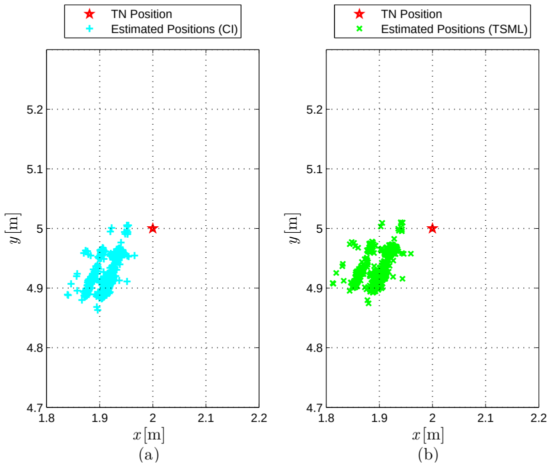


Fig. 4. The 1000 position estimates of TN_3 obtained using: (a) the CI algorithm; and (b) the TSML algorithm.

TABLE I. VALUES OF MINIMUM, MAXIMUM, AND AVERAGE DISTANCES BETWEEN THE TRUE TN POSITIONS AND THEIR ESTIMATES.

	$d_{\min}$ [cm]	$d_{\max}$ [cm]	d_{avg} [cm]
TN_1 - CI	0.03	12.2	5.7
TN_1 - TSML	1.4	13.8	7.2
TN_2 - CI	9.6	17.6	11.6
TN_2 - TSML	10.1	16.5	11.4
TN_3 - CI	4.6	19.5	11.5
TN_3 - TSML	5.7	20.8	12.2

We can define the minimum, the maximum, and the average distance between the true TN position and its estimates as

$$\begin{aligned} d_{\min} &= \min_{j\in\{1,\ldots,N_I\}} d_j \\ d_{\max} &= \max_{j\in\{1,\ldots,N_I\}} d_j \\ d_{\text{avg}} &= \frac{1}{N_I}\sum_{j=1}^{N_I} d_j . \end{aligned} \tag{30}$$

The performance of the localization algorithms is evaluated in terms of the values in (30).

Fig. 2 is relative to the position estimates of the target TN_1 of Fig. 1. More precisely, Fig. 2 (a) shows the 1000 position estimates obtained using the CI algorithm and Fig. 2 (b) shows the 1000 position estimates obtained according to the TSML algorithm. A comparison between Fig. 2 (a) and Fig. 2 (b) shows that the CI algorithm performs slightly better that the TSML algorithm in terms of the distance between the true position and the estimated ones. As a matter of fact, from Table I it can be observed that while the average distance d_{avg} when using the CI algorithm is 5.7 cm, the value of d_{avg} obtained with the TSML algorithm is 7.2 cm. Analogous considerations hold when considering $d_{\min}$ and $d_{\max}$. More precisely, the value of $d_{\min}$ relative to the CI algorithm is only 0.03 cm and it becomes 1.4 cm when using the TSML algorithm, while the values of $d_{\max}$ correspond to 12.2 cm and 13.8 cm, respectively.

Fig. 3 refers to the position estimates of the target TN_2 of Fig. 1, corresponding to the case where the TN is in the middle of the room. The 1000 position estimates obtained using the CI algorithm are shown in Fig. 3 (a) while Fig. 2 (b) shows the 1000 position estimates obtained with the TSML algorithm. In this case, the performance of the two algorithms are similar, as also shown in Table I, which shows that the values of d_{avg} obtained with the CI and the TSML algorithms differ by only 2 mm. Observe that in this case the values of d_{avg} are greater than those obtained when considering TN_1, meaning that the localization in the center of the room is less accurate. In particular, Table I shows that the distances between the true TN position and its estimates are always greater than $d_{\min} \simeq 10$ cm.

Finally, Fig. 4 is relative to the position estimates of the target TN_3 of Fig. 1 obtained: (a) using the CI algorithm; and (b) using the TSML algorithm. In this case, the average distance d_{avg} between the true TN position and its estimates is 11.5 cm when using the CI algorithm and 12.2 cm with the TSML algorithm, as shown in Table I. The performance of the two algorithms are similar also in terms of $d_{\min}$ and $d_{\max}$. Observe that the values of $d_{\max}$ are greater than those obtained in the previous two TN positions. The values of $d_{\min}$, instead, are greater than those relative to TN_1 but they are smaller than those relative to TN_2.

IV. CONCLUSIONS

This paper presents a novel JADE add-on that enables the implementation of distance- and location-aware agents. Using this add-on an agent can measure the distance that separates the smart appliance that hosts it from target beacons and other smart appliances. Experimental results summarized in Table I show that the average error in locating the smart appliance in an empty room is less than 15 cm, which ensures sufficient accuracy for considered education and ludic applications.

References

[1] F. Bergenti and A. Poggi, "Ubiquitous information agents," *Int'l J. Cooperative Information Systems*, vol. 11, no. 34, pp. 231–244, 2002.

[2] F. Bergenti, A. Poggi, B. Burg, and G. Caire, "Deploying FIPA-compliant systems on handheld devices," *IEEE Internet Computing*, vol. 5, no. 4, pp. 20–25, 2001.

[3] F. Bergenti, G. Caire, and D. Gotta, "Agents on the move: JADE for Android devices," in *Procs. Workshop From Objects to Agents*, 2014.

[4] F. Bergenti, G. Caire, and D. Gotta, "Large-scale network and service management with WANTS," in *Industrial Agents: Emerging Applications of Software Agents in Industry*. Elsevier, 2015, pp. 231–246.

[5] F. Bergenti and A. Poggi, "Agent-based approach to manage negotiation protocols in flexible CSCW systems," in *Procs.* 4^{th} *Intl Conf. Autonomous Agents*, 2000, pp. 267–268.

[6] F. Bergenti, A. Poggi, and M. Somacher, "A collaborative platform for fixed and mobile networks," *Communications of the ACM*, vol. 45, no. 11, pp. 39–44, 2002.

[7] F. Bergenti, G. Caire, and D. Gotta, "An overview of the AMUSE social gaming platform," in *Procs. Workshop From Objects to Agents*, 2013.

[8] F. Bergenti, G. Caire, and D. Gotta, "Agent-based social gaming with AMUSE," in *Procs.* 5^{th} *Int'l Conf. Ambient Systems, Networks and Technologies (ANT 2014) and* 4^{th} *Int'l Conf. Sustainable Energy Information Technology (SEIT 2014)*, ser. Procedia Computer Science. Elsevier, 2014, pp. 914–919.

[9] S. Monica and G. Ferrari, "Accurate indoor localization with UWB wireless sensor networks," in *Proceedings of the* 23^{rd} *IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE 2014)*, track on Capacity-Driven Processes and Services for the Cyber-Physical Society* (CPS), Parma, Italy, June 2014, pp. 287–289.

[10] Z. Sahinoglu, S. Gezici, and I. Guvenc, *Ultra-wideband positioning systems: Theoretical limits, ranging algorithms and protocols*. Cambridge, U.K.: Cambridge University Press, 2008.

[11] J. Zhang, P. V. Orlik, Z. Sahinoglu, A. F. Molisch, and P. Kinney, "UWB systems for wireless sensor networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 313–331, February 2009.

[12] S. Gezici and H. V. Poor, "Position estimation via Ultra-Wide-Band signals," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 386–403, February 2009.

[13] S. Monica and G. Ferrari, "Optimized anchors placement: An analytical approach in UWB-based TDOA localization," in *Proceedings of the* 9^{th} *International Wireless Communications & Mobile Computing Conference* (IWCMC 2013), Cagliari, Italy, July 2013, pp. 982–987.

[14] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1067–1080, November 2007.

[15] D. Dardari and R. D'Errico, "Passive ultrawide bandwidth RFID," in *Proceedings of the IEEE Global Telecommunications Conference* (GLOBECOM '08), New Orleans, LA, December 2008, pp. 1–6.

[16] C. Mensing and S. Plass, "Positioning algorithms for cellular networks using TDOA," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP 2006), Toulouse, France, May 2006, pp. 513–516.

[17] Y. Zheng, W. Chenshu, C. Tao, Z. Yiyang, G. Wei, and L. Yunhao, "Detecting outlier measurements based on graph rigidity for wireless sensor network localization," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 374–383, January 2013.

[18] Y. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, August 1994.

[19] S. Monica and G. Ferrari, "Swarm intelligent approaches to auto-localization of nodes in static UWB networks," *Applied Soft Computing*, vol. 25, pp. 426–434, December 2014.

[20] S. Monica and G. Ferrari, "Impact of the number of beacons in PSO-based auto-localization in UWB networks," in *Proceedings of the International Conference on the Applications of Evolutionary Computation* (EvoApplications '13)*, track on Nature-inspired Techniques for Communication Networks and other Parallel and Distributed Systems* (EvoCOMNET '13), Vienna, Austria, April 2013, pp. 42–51.

[21] S. Monica and G. Ferrari, "Particle swarm optimization for auto-localization of nodes in wireless sensor networks," in *Proceedings of the* 11^{th} *International Conference on Adaptive and Natural Computing Algorithms* (ICANNGA '13), Lausanne, Switzerland, April 2013, pp. 456–465.

[22] S. Monica and G. Ferrari, "An experimental model for UWB distance measurements and its application to localization problems," in *Proceedings of the IEEE International Conference on Ultra Wide Band* (ICUWB 2014), Paris, France, September 2014, pp. 297–302.

[23] S. Monica and G. Ferrari, "A swarm intelligence approach to 3D distance-based indoor UWB localization," in *Proceedings of the International Conference on the Applications of Evolutionary Computation* (EvoApplications '15)*, track on Nature-inspired Techniques for Communication Networks and other Parallel and Distributed Systems* (EvoCOMNET '15), Copenaghen, Denmark, April 2015, pp. 42–51.

[24] K. C. Ho, X. Lu, and L. Kovavisaruch, "Source localization using TDOA and FDOA measurements in the presence of receiver location errors: Analysis and solution," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 684–696, February 2007.

[25] G. Shen, R. Zetik, and R. S. Thomä, "Performance comparison of TOA and TDOA based location estimation algorithms in LOS environment," in *Proceedings of the* 5^{th} *Workshop on Positioning, Navigation and Communication* (WPNC 2008), Hannover, Germany, March 2008, pp. 71–78.