Widespread Mobile Devices in Applications for Real-time Drafting Detection in Triathlons

Iztok Fister[a], Dušan Fister[a], Simon Fong[b], Iztok Fister Jr.[a]
[a] University of Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, 2000 Maribor, Slovenia
Email: (iztok.fister, dusan.fister, iztok.fister2)@uni-mb.si
[b] University of Macau, Faculty of Science and Technology, Av. Padre Tomas Pereira, Taipa, Macau SAR
Email: ccfong@umac.mo

***Abstract*— Today, global object-positioning is accomplished very precisely by GPS satellite technology. Access to this information is provided globally by widespread mobile devices with integrated GPS receivers from everywhere. On the other hand, mobile devices are connected to world-wide networks that ensure anytime access to application service (also web service) infrastructure based on application servers. Information everywhere at anytime is a key issue of pervasive computing. As a proof of concept that reflects a power of the pervasive computing, the application for drafting-detection in triathlon competitions was developed. This shows that the widespread mobile devices with GPS feature are appropriate for solving of real problems addressing the precise object-positioning.**

***Index Terms*— pervasive computing, real problem, mobile devices, GPS, web services, application server, triathlon**

I. Introduction

The development of mobile technologies enables users to manage information during their lives, as well as within business environments from *everywhere* on the world. However, this can only be realized by convenient applications. Such applications integrate software, hardware, infrastructure and services [18] and provide an *anytime* access to the information. The paradigm "*information everywhere at anytime*" represents the goal of, e.g., *Pervasive* or *Ubiquitous Computing* [28]. Both terms describe the integration of mobile front-end devices with back-end application infrastructure. Device management and application management are the main issues for the back-end systems. On the other hand, these systems must be prepared for serving the growing demands for network access from everywhere. Furthermore, such systems' services become *context-aware* [17], i.e. the answer to context-aware services depends on the contexts' elements as, for example, who, where, when and why someone demands such service.

One of the featured topics of pervasive computing and context-aware services is *positioning* [17]. Location awareness is a basic requirement for new applications on mobile devices [8]. The first step when positioning of object on Earth is the distance calculations between a mobile device and number of reference points. As a result, the mobile device determines the position of the object, whilst the reference points are implemented as a constellation of GPS satellites around the Earth. Typically, the distance is calculated by means of the *triangulation method* [39].

The triathlon is relatively young sport because its beginning only date back to 1978, when a group of enthusiastic athletes decided to finish three marathons using different disciplines, i.e. swimming, bicycling and running, all in one day. The competition got the name Ironman and today represents one of the greatest challenges for the persistence of human beings. Interestingly, this sport is growing wide-world each day, with more and more devotees. Moreover, the triathlon was integrated into the family of Olympic sports by the International Olympic Committee in 2000.

Firstly, in the triathlon competitors should compete in his own right. However, in order to attain better results some competitors forget about fair-ply. In place of competing alone, he exploits the competitor in front of him. This prohibited support is especially employed in bicycling, where the violating competitor rides his bike directly behind an other competitor. Thereby, the violating competitor saves his power for later efforts and rides his bicycle faster. This phenomenon is known as *drafting* (also *slipstreaming*), and is punished by referees. Typically, the referee on a motorcycle can eliminate the drafting competitor from the competition, for even up to five minutes. The rules for drafting-detection are regulated by the World Triathlon Corporation (WTC), and are discussed later.

In European triathlon competitions especially, drafting has been growing and reflects unfavorably on this sport. Referees try to restrict this phenomenon by punishing the drafting competitors. Its detection without modern technology has become impossible because of the increasing number of competitors (more than 2,000 per triathlon). Despite having the appropriate technology for solving this problem, a concrete solution does not exist on the market today.

This article demonstrates that drafting in triathlon competitions can be detected, in practice. An application for drafting-detection in triathlon competitions has been developed in order to prove this concept. This application

doi:10.4304/jetwi.5.3.310-321

consists of two parts:

- pervasive application on a mobile front-end device and
- context-aware service provided by a back-end system.

This pervasive application acts as a gateway that obtains the position of a competitor when riding his bicycle, and transmits this information to the context-aware service trough a worldwide wireless network. This context-aware service acts as follows. Firstly, it identifies the competitor, then, it compares the position of that competitor with the positions of other competitors within his neighborhood. Note that this neighborhood is determined from the WTC rules. If the identified competitor violating the drafting rules, the referees on the motorcycles are notified. Besides trying to find a solution to this problem, the focus is also on technical issues faced when developing this kind of pervasive application. Firstly, its structure is identified [15], the particular elements are then developed according to the recommendations in [6], [8]. Note that this application can be easily integrated within timing system controlled by the domain-specific language EasyTime, as proposed in [13], [14].

The structure of this article is as follows. In Section 2, the phenomenon of drafting in triathlon competitions is explained in detail. Beforehand, however, the characteristics of triathlon competitions are discussed. Section 3 describes the proposed application for drafting-detection in triathlon competitions. In Section 4 experiments and results are presented. In Conclusions results are summarized and directions for further development are placed.

II. Drafting-detection in triathlon competitions

This Section is divided into two parts. The first describes the main characteristics of triathlon competitions, whilst the latter focuses on the drafting-detection in triathlon competitions. In this sense, the rules of the WTC are presented for detecting and punishing this phenomenon. Although today several kind of triathlon exist, this article concentrates on Ironman. This kind of triathlon is still one the most prominent.

A. Ironman

Ironman (also the long triathlon) is held under the auspices of the WTC Association. It consists of three marathons covering different disciplines, in other words (Fig. 1):

- 3.8 kilometer swim,
- 180 kilometer of bicycling and
- 42.2 kilometer run.

The competitors start with the swim, continue with the bicycling, and finish with the run. All disciplines are performed in continuation. Between particular disciplines, however, competitors need to prepare themselves for the next discipline. This preparation is performed in *transition areas*. Here, two transition areas exist. In the first, the competitor takes off his swim suite and prepares himself for bicycling (TA1 in Fig. 1), whilst in the second he takes off his bicycle gear and prepares himself for the run (TA2 in Fig. 1). As a result, the completed achievement of the competitor consists of finishing all three particular disciplines and the time spent in each transition area.

B. Drafting

The phenomenon of drafting arises when one of the competitors purposely rides a bicycle directly behind an other and, thereby, avoids the persistence of wind. A drafting competitor can increase his average speed when bicycling whilst, at the same time, save energy consumption. Usually, in Ironman not only one single competitor drafts but a whole group of them together. Moreover, by exchanging the leading positions within the group (the leading competitor surrenders his position to a fresh competitor riding behind him and goes to take some rest at the rear of his drafting group), thus an additional speed up is achieving.

However, such a grouping has nothing to do with a time-trial competition, where a single competitor overcomes the course. Moreover, the results of those competitors within that group do not express the powers of individuals, and represent drafting violations that are punishable by referees. In fact, the WTC prescribes the following rules in the official Ironman competitions in order to avoid drafting [38]:

- drafting of another bike or any other vehicle is disallowed,
- competitors must keep 7 meters (4 bike lengths) distance between their bikes, except when overtaking,
- overtaking occurs when the overtaking competitors front wheel passes the leading edge of the competitor being overtaken,
- overtaking-competitors may pass on the left for up to 20 seconds, but must move back to the right-side of the road, after passing,
- overtaken competitors must immediately fall-back 7 meters (4 bicycle lengths), before attempting to regain the lead from a front runner.

As illustrated in Fig. 2, competitor B is violating the drafting condition because he is riding his bicycle 4 meters behind competitor A. Although competitor C is 8 meters behind competitor A he is in the so-called drafting zone of competitor B because he is only 4 meters behind competitor B. Note that the drafting zone is defined as an area that is 7 meters long and 2 meters wide, and is in relation to the leading competitor within the group. For example, each competitor located within the drafting area of competitor A in Fig. 2 forms the drafting neighborhood of competitor A. Whenever a competitor enters this zone for more than 20 seconds a drafting violation occurs.

To date, referees are responsible for drafting-detection in Ironman. They monitor competitors along the bicycle-course, from a motorcycle. Thereby, they try to stay as inconspicuous as possible. The time of drafting, as well as the distances between drafting competitors are estimated

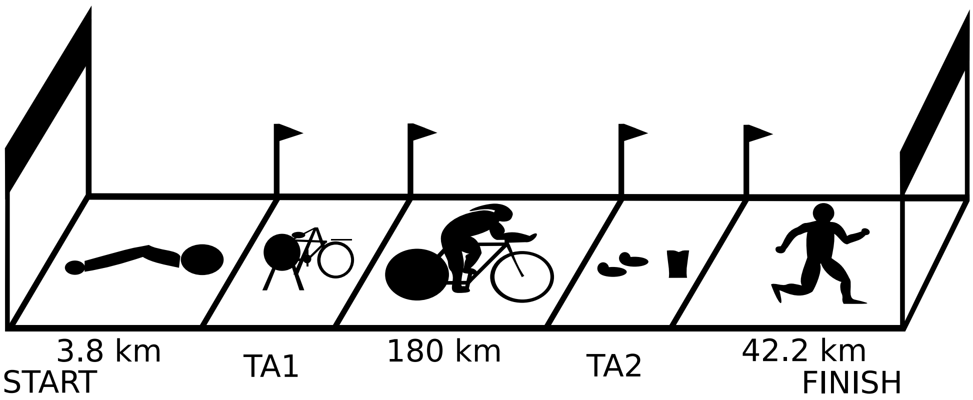

Figure 1. Ironman

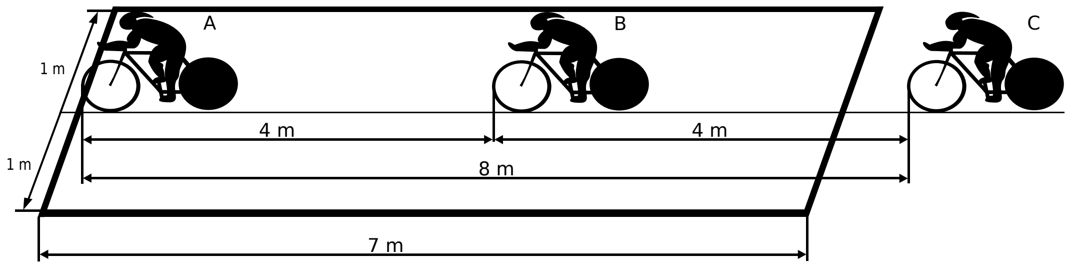

Figure 2. Drafting

by the referees approximately. Furthermore, they could be dealing with a limited number of drafting violations all at the same time. Likewise, they remain powerless when drafting is caused by a group of competitors. As a result, the automaton of drafting-detection is necessary.

III. Design of the application for drafting-detection in Ironman

In order to detect drafting along a bicycle course in Ironman, each competitor needs to be equipped with some mobile device that is capable of positioning his location as precisely as possible, and to transfer this position to a server via some ubiquitous network (Fig. 3). The positioning of particular object on the Earth is today enabled due to satellite technology, i.e. the Global Positioning System (GPS). On the other hand, the Internet is a really truly ubiquitous network today. Widespread use of smart mobile devices can incorporate both demands for drafting-detection: the support of global positioning whilst having access to the Internet, and therefore, appears to be the most suitable for this application. Moreover, referees use the same kind of devices for obtaining information about drafting-competitors.

Smart mobile devices, connected to the Internet due to widespread mobile networks, form complex ecosystem, where all parts work together seamlessly [15]. The mobile ecosystem is divided into the following elements:

- networks,
- devices,
- platforms (operating system, application framework),
- applications and
- web services.

The remainder of this article presents how these elements are addressed when designing this application for drafting-detection in Ironman.

A. Networks

A mobile communication ecosystem is needed in order to make a mobile ecosystem possible for communicating with the Internet. This is comprised of technologies, standards, and networks [31] that have been developing since 1950. A survey of wireless networks from their beginning to recent days, is presented in Table I.

As can be seen from Table I, the evolution of wireless networks can be divided into generations. For example, generation G1 captured analogue mobile telephones, where an user occupies the circuit switched line's whole duration of connections (multiple access to the line is disallowed).

During generation G2, a digital voice transmission via circuit switched networks, i.e. GSM (Global System for Mobile communications), was extended with GPRS (General Packet Radio Services) that allow packet data transfer [26]. The switching between data and voice is conducted by TDMA (Time Division Multiple Access). In addition to GSM, standards, such as USDC (US Digital Cellular) and PDC (Pacific Digital Cellular) are emerged on non-European markets. During the generation 2.5G, GSM was enhanced by EDGE (Enhanced Data rates for Global Evolution) that increase data transmission rates.

The 3G standards are CDMA2000 (Code Division Multiple Access 2000), TD-SCDMA (Time Division - Synchronous CDMA) and UMTS (Universal Mobile Telecommunications Systems). The latter is the successor

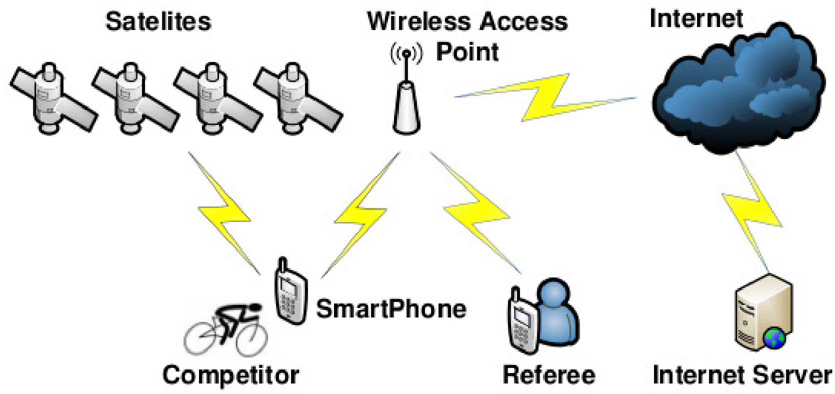

Figure 3. Drafting-detection in Inronman

TABLE I.
SURVEY OF WIRELESS NETWORKS.

Generation	1G	2G	2.5G	3G	3.5G	NGMN
Wireless Technologies	Analog	GPS/GPRS USDC PDC	EDGE	UMTS CDMA2000 TD-SCDMA	HSPA HSDPA HSUPA	EPS WinMAX
Features	No Multiple Access	TDMA	TDMA	CDMA	FDMA	FDMA

of GSM. These standards support the data packet transfer only. Besides higher data transmission rates CDMA was also specified for line-sharing. During generation 3.5G, UMTS was evolved to standards as: HSPA (High Speed Packet Access), HSDPA (High Speed Downlink Packet Access) and HSUPA (High Speed Uplink Packet Access). For line-sharing, the frequency division (FDMA) was introduced that remains the main multiple access mechanism.

The recent state during the evolution of mobile-networks is represented as NGMN (Next Generation Mobile Networks) that embrace two standards: EPS (Evolved Packet System) and WiMAX (Worldwide Interoperability for Microwave Access). The former is an evolution of UMTS systems, whilst the latter is new technology. Both the mentioned technologies can be considered as the last leg to 4G.

B. Devices

Pervasive devices combine the following four paradigms: they are strongly decentralized, diversified, connected, and easy to use [18]. Essentially, the fourth paradigm demands that the complex mobile and Internet technology is hidden behind a friendly user-interface. This interface between the user and machine is at the heart of *human-computer interface* (HCI), i.e. a discipline that has reached its maximum prosperity by the growth of pervasive computing [17].

Pervasive devices are divided into four main categories [18]:

- information access devices,
- intelligent appliances,
- smart controls and
- entertainment systems.

The first category of devices includes pocket-sized smart-phones (iPhones, BlackBerrys, etc.) that allow on-line access to information services (corporate databases, Internet pages, etc.). Intelligent appliances are pervasive devices with specific intelligence, like GPS navigation, industry controller, information car system, smart houses, etc. New kinds of entertainment systems address the world of modern broadcasting, such as interactive digital television, video on demand, etc.

Today, pervasive devices are a combination of more categories. For example, smart-phones incorporate the following features: classical telephone, information access via wireless networks, GPS navigation, IpTV, etc.

1) Global Positioning System: GPS is based on a set of broadcasting satellites that are used as reference points to calculate the position of an object on the Earth. It consists of three segments: space, user and control. The space segment is composed of 24 to 31 satellites orbiting within GPS constellation [2] alligned to the rotation of the Earth, orbiting at an altitude of approximately 20.200 kilometers. The user segment is composed of hand-held receivers (e.g., Polar, SmartPhone, etc.) or devices fixed on a vehicle (e.g., Garmin navigation system). The correct operations of the satellites are provided by the control segment.

The task of a GPS receiver is to identify almost four satellites, to determine the distances to each one, and to use this information for calculating the position of an object on the Earth. This calculation is based upon the mathematical principle of triangulation [39]. Note that a GPS receiver determines a three-dimensional position for the object within geographical coordinate system. Additionally, the Coordinated Universal Time (UTC) is transmitted by the satellites. In the geographical coordinate system, the position is represented by its longitude, latitude, and altitude.

GPS obtains two levels of services: Standard Positioning Service (SPS) and Precise Positioning Service (PPS). The former can position the object with a precision of less than 20 meters [23], and is devoted to world-wide usage. The latter is more precise (e.g., up to 10 centimeters), therefore, it is deployed for military purposes. However, cost of SPS is much less than the cost of PPS. The mentioned precision is valid when the position is determined by four active satellites only. In practice, the number of active satellites can be increased and, consequently, the position of the object can be better calculated.

On the other hand, the absolute position of a competitor in Ironman is less important than the relative distance to the other competitor using drafting-detection. As a result, this can additionally increase the precision of distance calculation.

2) Differential Global Positioning System: The classical GPS (also stand-alone GPS) cannot be used for the precise positioning of an object on the Earth. This is due to a variety of risks that influence on the GPS performance. These risks relate to the effects of the ionosphere and troposphere, satellite maintenance, unscheduled satellite failures, satellite unavailability due to scheduled maintenance, repairs, repositioning and testing [29]. These anomalies may result in an unpredictable range of errors above the operational tolerances of GPS, which cause degraded availability, reliability, accuracy and safety (integrity monitoring).

Therefore, a supplementary navigation method, named *differential* GPS (DGPS) is used to significantly improve the accuracy and integrity of the stand-alone GPS [7], [11], [16], [20], [21], [25], [30], [35]–[37].

C. Platform

The platform denotes a core programming language in which all the application software is written [15]. Usually, the platform includes the hardware architecture, operating system and application framework (programming languages, run-time libraries or graphic user interface). Each mobile device running an operating system is treated as smart-phone. The operating system performs core services or tools that enable applications to talk to each other and share data or services. The application framework enables the development of a new application. It runs on top of operating system and provides support for sharing core services, e.g., communication, messaging, graphics, positioning, security, etc. Table II displays a review of the most significant mobile platforms with associated operating systems and application frameworks.

Note that all platforms are split into three categories: licensed (e.g., BREW, Windows Mobile), proprietary (e.g., Palm, BlackBerry, iPhone, Nokia) and open-source (Android). The development of the application for drafting-detection in Ironman was performed on Android.

1) Android: In order to write well-formed Android applications, a good understanding of Android's key concepts (e.g., Linux kernel, OpenGL, SQL database, etc.) is necessary. The overall system architecture is illustrated in Fig. 4.

As can be seen from this figure, the Android system architecture is divided into five layers as follows [6]:

- Linux kernel,
- libraries,
- Android runtime,
- application framework and
- applications and widgets.

Each layer depends on the services provided by the layers below it. Android is built on top of a Linux kernel. This is a stable and proven foundation that supplements Android with many operating system services, such as: memory management, process management, networking, etc. An Android developer never use Linux directly but over its utilities.

Android libraries are shared between applications. These are written in C or C++, and compiled for the particular hardware architecture. The most important libraries implement function, such as: surface manager, 2D and 3D graphics, media codecs, and browser engine. Note that these libraries do not represent applications. Conversely, they are used by higher-level applications to call the lower-level services.

Android runtime consists of a Dalvik virtual machine, and the core Java libraries. The Dalvik virtual machine is Google's implementation of Java, optimized for mobile devices. The Java core libraries are also adapted for this virtual machine. That is, all application code are written in Java, compiled from traditional *.java* and *.jar* files to *.dex*, and executed on a Dalvik virtual machine.

The application framework provides high-level building blocks that help the Android developers to create any new application. The framework consists of several managers for the handling of: activities (Android's synonym for process), contents (sharing data between applications), resources (text strings, bitmaps, etc.), positioning (GPS devices), and notifications (messages, appointments, proximity alerts, etc.). The framework is pre-installed as part of Android (Android SDK). However, it can also be extended with new components, as necessary.

The applications and widgets layer present the higher level of the Android architecture. This level is only visible by the end-user. In Android, the end-user interacts with the application over the whole screen. On the other hand, widgets (also gadgets) only operate within a small rectangle of the main screen.

D. Application for drafting-detection in Ironman

Application for drafting-detection in Ironman was written with regard to guidance found in publications, such as [6], [8]. It is a graphical front-end for the Android operating system (Fig. 5). This application supports the following functions:

TABLE II.
REVIEW OF MOBILE PLATFORMS.

Platform	Operating System	Application Framework
BREW	BREW OS	BREW
Windows Mobile	Windows Mobile	Windows Mobile
Palm	Palm OS	WebKit
BlackBerry	BlackBerry OS	Java API
iPhone	Mac OS X	Cocoa Touch
Nokia	Symbian	Qt
Open Source	Android	Android SDK

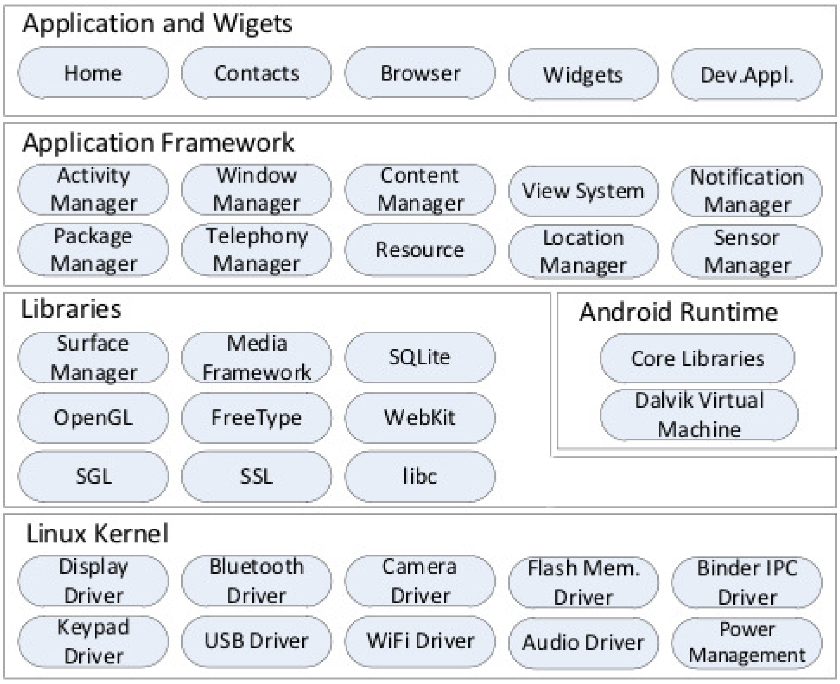

Figure 4. Android Architecture

- about-box,
- setup,
- start GPS agent,
- stop GPS agent and
- exit.

All functions can be activated by pressing the corresponding button. The about-box function displays a current version and the copyright information for the application. Setup enables the end-user (e.g., the competitor) to enter the context-aware and control variables, i.e. the starting number, the URL address of the web service, and the timer interval that determines a frequency of transferring the GPS information to a server. These variables are stored within the application preferences area and implemented by the Android's *PreferenceActivity* class [6]. Obviously, these variables are shared between other applications' activities. The start GPS agent function represents the main part of the application, i.e. the *Send* class. The execution logic of this class is illustrated in Algorithm 1. The stop GPS agent function ends the GPS agent, whilst the exit function finishes the application.

Note that because of this article's limitations, *Send* class (Algorithm 1) is not presented in details. Some variables are omitted but their omission is denoted by dots. Additionally, only the more important functions are presented here.

Send Java class (Fig. 1) extends the *LocationListener* activity class that implements the GPS listener. This class includes several global variables. The more important are the variables denoting the following classes:

- *locationManager* of class *LocationManager*,
- *envelope* of class *SoapSerializationEnvelope*,
- *androidHttpTransport* of class *HttpTransportSE* and
- *myTimer* of class *Timer*.

The *LocationManager* class implements interactions with a GPS device. The next two classes, i.e. *SoapSerializationEnvelope* and *HttpTransportSE*, are devoted to communication with web service provider, whilst the *Timer* class initiates communication with the web service provider.

The function *onCreate()* is called when the *Send* class is created. Firstly, three preference variables, i.e. *str_number*, *tim_tick*, and *URL*, are initialized. In order to create a *LocationManager*, connection with the GPS device is established, whilst *SoapSerializationEnvelope* establishes a connection with the web service provider,

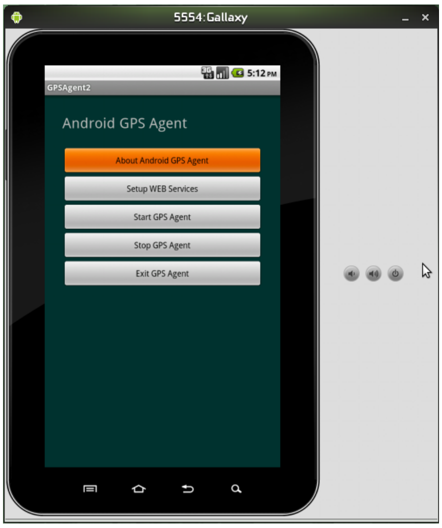

Figure 5. Android Application for Drafting Detection

as determined by the *URL* preference variable. Moreover, a GPS timer is activated by a *Timer* class. Each *tim_tick* seconds, this timer calls a method *TimerMethod* that calls a *postData(longitude, latitude, altitude, utm)* function. The function parameters represent the position of GPS device at *utm* time and are defined globally. This position is obtained by *onLocationChanged()*, which is called when the position of GPS device is changed.

The *postData()* function implements a transfer of positioning data to the web service provider. Firstly, a new SOAP request is created addressing the correct web service namespace (*NAMESPACE*) and method name (*METHOD_NAME*). Then, this request is filled with the GPS position, serialized by the *Ksoap2* library and transfers the SOAP message to the web service directly [22]. Note that all wrap and unwrap SOAP messages are performed by the *Ksoap2* library automatically and, therefore, these messages are transparent for the programmer.

SOAP (Service Oriented Architecture Protocol) is a default message transfer protocol in SOA (Service Oriented Architecture) [10]. SOAP messages are used by wrapping application specific XML messages within a plained XML based envelope structure [4].

E. Web services

The term web services describes a set of open standards that enables web based applications to communicate over the Internet with each other and with clients [3]. This set consists of protocols, such as: Extensible Markup Language (XML), Service Oriented Architecture Protocol (SOAP), Web Service Description Language (WDSL) and Universal Description Discovery and Integration (UDDI), where XML is intended for tagging the data, SOAP is a message transfer protocol, WDSL is used for describing the available services, and UDDI is for finding the services over the Internet. Furthermore, web services allow organizations to communicate transparently with other organizations.

System Oriented Architecture (SOA) is a de-facto standard for web service message exchange [34]. SOA is based on the principle of distributed application services that communicate over the Internet with each other through messages.

Because of many standards, the development of web services is difficult. In order to simplify the development, Apache has prepared an Axis2 engine [27] that allows developers to develop web services on a higher level. This engine can be used for developing the drafting-detection web service, as well. Additionally, the developing tool Eclipse [1] was employed.

The web service for drafting-detection in Ironman is context-aware because each SOAP message containing the position of a GPS device in geographical coordinates is identified by the starting number of competitor. In fact, the enacting of a web service can be divided into three tasks:

- a transformation of geographic coordinates to UTM,
- a distance calculation and
- a drafting-detection.

These tasks are described in detail in the rest of the article.

1) Transformation of geographical coordinates to UTM: Usually, a geographical coordinate system is used by GPS, where a position is represented by:

- a latitude and
- a longitude.

Note that GPS devices also provides an altitude. The longitude represents the angle from the center to a particular *parallel* on the surface of the Earth (direction East-West). Longitude is an angle from the center to a particular *meridian* on the surface of the Earth (direction North-South). Both values can be represented as degrees in the form of decimal number or in discrete form: degree, minutes and seconds (DMS).

The Earth is divided by the equator into Northern and Southern hemispheres, whilst the Prime Meridian divides

Algorithm 1 Android Java Class Implementing GPS Listener.

```
public class Send extends Activity implements LocationListener {
  ...
  private LocationManager locationManager;
  private SoapSerializationEnvelope envelope;
  private HttpTransportSE androidHttpTransport;
  private Timer myTimer;
  ...
  public void onCreate(Bundle savedInstanceState) {
    str_number = Prefs.getStartNumber(getApplicationContext());
    tim_tick = Prefs.getTimerTick(getApplicationContext());
    URL = Prefs.getURL(getApplicationContext());
    locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);
    androidHttpTransport = new HttpTransportSE(URL);

    myTimer = new Timer(); // activate the GPS timer
    myTimer.schedule(new TimerTask() {
      @Override
      public void run() { TimerMethod(); }
    }, 0, tim_tick*1000);
  };

  private void TimerMethod() {
    this.runOnUiThread(Timer_Tick);
  };

  private Runnable Timer_Tick = new Runnable() {
    public void run() {
      if(gps_status == GPS_ENABLED)
        postData(longitude, latitude, altitude, utm);
    }
  };

  public void postData(double lon, double lat, double alt, long unt) {
   SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
    PackRequest(request, str_num, lon, lat, alt, unt);
    envelope = new SoapSerializationEnvelope(SoapEnvelope.VER11);
    envelope.setOutputSoapObject(request);
    try {
      androidHttpTransport.call(SOAP_ACTION, envelope);
    } catch(Exception e) {
      Error.setText(e.getLocalizedMessage());
    }
  };

  public void onLocationChanged(Location location) {
    ...
    longitude = location.getLongitude();
    latitude = location.getLatitude();
    ...
    gps_status = GPS_ENABLED;
  };
}
```

it into Eastern and Western hemispheres. Latitude captures the values from 0° to 90° in the Northern and the values from 0° to $-90°$ in the Southern hemispheres. On the other hand, longitude captures the values from 0° to 180° in the Eastern and the values from 0° to $-180°$ in the Western hemispheres.

With geographic coordinates it is not easy to calculate. Therefore, these need to be transformed into the metric 3-dimensional coordinate system UTM (Universal Transverse Mercator system). This system represents a Mercator projection of the Earth to a plane and is divided into 60 longitude and 30 latitude zones. Each position in this coordinate system is presented as quadruple $\langle lon\_zone, lat\_zone, east, north\rangle$, where lon_zone and lat_zone are the numbers of the longitude and the latitude zone, whilst $east$ is the projected distance from the Prime Meridian, and the $north$ the projected distance from the equator. Both distances are defined in meters.

Although the basis of the geographical coordinates transformation into coordinate system UTM represents a basic trigonometric and algebraic functions the transformation formulas are complex [24]. Therefore, for the necessity of this proof of concept we decided to use the existing implementation of author [32] in Java.

2) Distance calculation: However, the traditional Euclidian distance is used for distance calculation. Let us suppose that the positions of the two competitors $A = (x_1, y_1)$ and $B = (x_2, y_2)$ are given. Then, the distance between both is expressed in 2-dimensional space, as follows:

$$dist2(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \qquad (1)$$

However, in 3-dimensional space the Euclidian distance is expressed as:

$$dist3(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}, \qquad (2)$$

if we suppose that the positions of competitors are given as $A = (x_1, y_1, z_1)$ and $B = (x_2, y_2, z_2)$.

3) Drafting-detection: In order to handle a lot of requests addressed by competitors' GPS positions (usually, one position per second) and to make code for serving these requests as simple as possible, an efficient data structure is necessary for the web service. The *map* table is placed on the top of the data structure hierarchy. This maps the starting number of competitor to his current place in the race. In fact, each competitor is represented by the 6-tuple $\langle i, x, y, z, t, l\rangle$ (denoted as $item$ data structure), where i denotes the starting number of the competitor, (x, y, z) the UTM position, t the time of event registration and l the calculated number of kilometers covered by the i-th competitor, i.e. his traveled path length. However, the calculated path length l is obtained by projection of the competitor's current position to the line connecting the points that are sampled the bicycle course with the precise GPS device in small intervals of time (e.g., 1 second).

However, the path length l has an impact on the competitor's current placing. The higher the path length the better his current place. Essentially, this place is gained by a sorting of the map array with regard to the descending number of kilometers covered. Because at one time only one request is handled (serialization), a small number of exchanges is necessary by this sorting algorithm.

An algorithm for drafting-detection in Ironman (Algorithm 2) is an implementation of WTC rules, as described in Sect. II-B.

Note that, in Algorithm 2, an additional 2-dimensional array $viol[i][j]$ is used that contains the starting time of drafting violation between competitors i and j in seconds. However, if the drafting-detection is occurring, i.e. the Euclidian distance $dist2(i, i-1)$ between competitor i and its predecessor $i-1$ amounts to less than 7 meters, for more than 20 seconds, a drafting violation is announced. Note that the Euclidian distance $dist2()$ in 2-dimensional space is used here. However, if the time of sampling

Algorithm 2 Algorithm for the drafting-detection in Ironman.

```
if(dist2(item[map[i]].l, item[map[i-1]].l) < 7) { // for i > 1
    if(viol[map[i]][map[i-1]] == 0) {
        viol[map[i]][map[i-1]] = item[map[i]].t;
    } else if((item[map[i]].t-viol[map[i]][map[i-1]]) > 20) {
        System.out.println( "Competitor " + i + " drafts the competitor " + i-1);
    }
} else
    viol[map[i]][map[i-1]] = 0;
}
```

the competitor's position is relatively small, the third dimension can be neglected.

IV. Experiments and results

The goal of experiments was to show that widespread mobile devices can be used in real-time applications for precise object positioning. In this sense, three experiments were conducted:

- comparing the precision of various GPS devices by positioning of a reference point on the Earth,
- comparing the reference distances on the Earth with the distances that were calculated using data measured by GPS devices statically,
- comparing the reference distances on the Earth with the distances that were calculated using data measured by GPS devices dynamically and
- simulation of drafting-detection.

In the first experiment, a reference point on the Earth was selected and its absolute position was measured by four various GPS devices, as follows:

- differential GPS logger Sanav ML-7,
- differential GPS logger Garmin Etrex-H,
- smart-phone Samsung Gallaxy and
- smart-phone HTC Wildfire.

The main characteristic of the GPS logger is that it can log the current position of a GPS device at a predefined time interval into an internal storage. These positions are copied into a personal computer for additional analysis. Typically, data are saved in GPGGA records [19]. Additionally, these loggers are able to provide differential GPS correction.

The results of the experiment are illustrated in Table III. Note that the data in this table were obtained within intervals of one second. In fact, the average measurements of latitude, longitude, altitude, and distance, calculated according to Equation (1) are presented. However, the average point (line Total in the table) was taken as reference point to calculate the distance. Furthermore, the standard deviations of the average measurements (Stdev1, Stdev2, Stdev3 and Stdev4) are also presented in the table.

It can be seen from Table III that the position of the selected point was measured differently by each device. On average, the position was measured within an accuracy of 1.32 meters. According to altitude, the most accurate was the Garmin Etrex-H device because the measurements were performed at an altitude of 190 meters.

In the second experiment, 14 co-linear points were selected within a plain on the Earth. These points were arranged at distances of one meter between each other. Then, a walk across these was initiated and the appropriate distances from the starting point were calculated using the GPS positions. At each reference point, a halt of 10 seconds was taken. Because this time was enough for the GPS devices to precisely calculate the current positions this experiment was identified as a statical measuring of distances between reference points. Here, the same types of GPS devices were employed as in the first experiment. The average results of calculating the distances after 10 walks, are presented in Picture 6, where the line *Distance* denotes the real reference points.

As can be seen from Picture 6, the logger differential GPS devices (Sanav ML-7 and Garmin Etrex-H) measured the GPS distances of the reference points less precisely than the smart-phones (Samsung Gallaxy and HTC Wildfire), in reality.

The third experiment was performed similarly to the second. However, no halt was taken between walking. Thereby, as the devices have insufficient time for determining the current position precisely, the experiment was also identified as dynamically measuring the distances between the reference points. In this experiments, the uniform movement of GPS devices across reference points was observed. The moving speed of the walk across reference points was 1 m/sec. Obviously, this movement is much closer to reality than the movement in the second experiment. The results from calculating the distances obtained from the GPS positions, are presented in Picture 7. Note that the line *Distance* denotes the real reference points, whilst the other lines were calculated from the reported GPS positions. From Picture 7, it can be observed that all the GPS devices used in this experiment followed the real *Distance* line closely except for the smart-phone Samsung Gallaxy.

Finally, the drafting-detection was simulated. The simulation was performed as follows. A competitor A competes with competitor B over a bicycle course of length 3.332 kilometers. Note that the course was flat and only one lap was ridden. Competitor B started one minute after competitor A. Each competitor was equipped with a HTC Wildfire smart-phone. Data about the sports activity were transmitted to the Internet and at the same time, logged into internal storage, while the simulation can be tracked on the Internet using Google Maps online (Fig. 8). Further, the logged data can be downloaded on a personal computer for further analysis. From researchers point of view, however, an offline analysis of logged data was interested in order to explore if these data were accurate enough that the algorithm for drafting detection could be convinced that the drafting condition was reliable arisen.

The results of the simulation are presented in Fig. 9 that the drafting-violation of competitor B by competitor A at 1.591 kilometers was detected, i.e. after 4:52 minutes of the race. Competitor B remained within the drafting zone of competitor A for 2:46 minutes (or the whole 733 meters). After 7:28 minutes (at 2,324 meters) competitor B overtook competitor A and completed the course in

TABLE III.
PRECISIONS OF VARIOUS GPS DEVICES BY POSITIONING A REFERENCE POINT ON THE EARTH.

Device	Latitude	Stdev1	Longitude	Stdev2	Altitude	Stdev3	Dist	Stdev4
ML-7	46.6159988	2.30E-05	16.1487849	9.30E-06	218.65	1.12	1.33	1.57E+00
Etrex-H	46.6160046	9.31E-06	16.1487547	1.05E-05	186.29	0.22	1.21	0.22E+00
Gallaxy	46.6160096	3.27E-06	16.1487881	4.48E-06	235.08	1.04	1.42	4.98E-06
Wildfire	46.6160095	4.99E-14	16.1487544	2.14E-14	238.00	0.00	1.31	0.00E+00
Total	46.6160056	8.90E-06	16.1487705	6.08E-06	219.51	0.59	1.32	0.45E+00

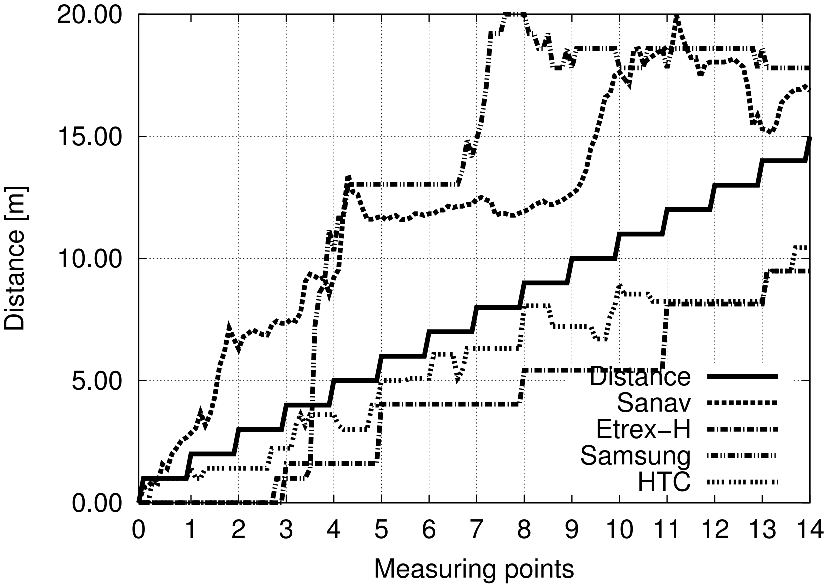

Figure 6. Comparison between the calculated distances obtained by the GPS devices statically

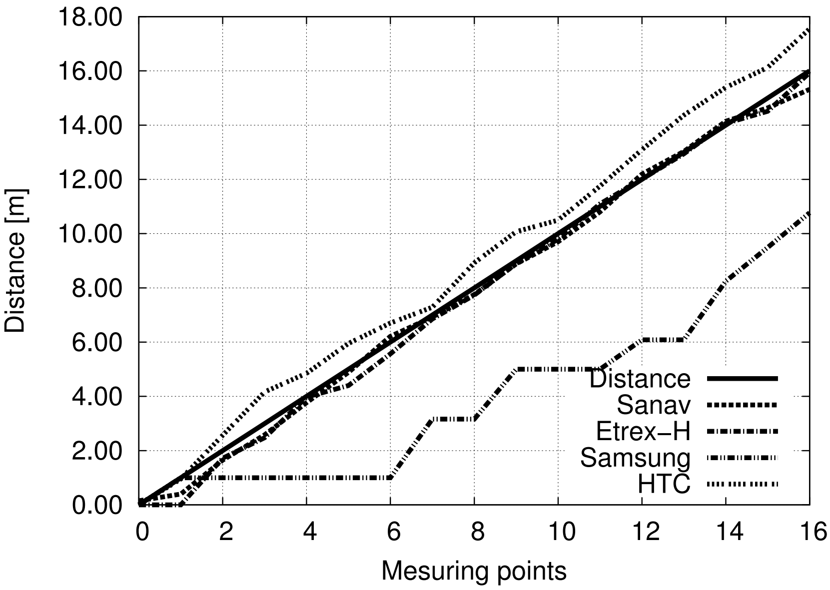

Figure 7. Comparison between the calculated distances obtained by GPS devices dynamically

9:38 minutes, whilst competitor B finished in 10:59 minutes. That is, competitor B overtook competitor A for 1:21 minutes. This simulation showed that the drafting condition could be successfully detected using the mobile smart-phones.

In summary, HTC Wildfire shows that it can be a good candidate for usage in real-time application for drafting-detection in Ironman because it includes a very precise GPS receiver and the reliable UMTS transmitter for connection to the Internet. However, to use this smart-phone on a bicycle would be awkward because of too much size and weight. Competitors in bicycle races are very sensitive to any excessive weight loaded on the bicycle. Furthermore, an additional problem represents the

Figure 8. Simulation of drafting (Powered by Google Maps)

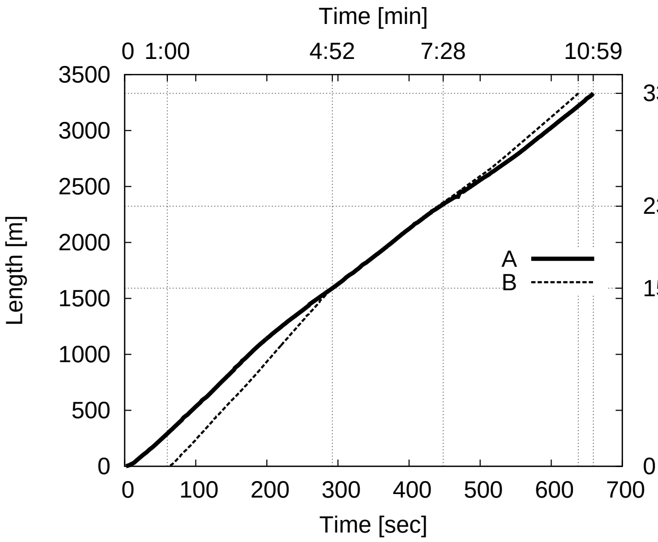

Figure 9. Simulation of drafting-detection

power consumption of smart-phones, which is too high when these are fully-operational.

Smart-phones are dedicated to general usage and support various applications that additionally spend the power consumption. Therefore, only specialized hardware devices with precise GPS, efficient HSPA, and low power consumption, can solve this problem as a whole in the future.

V. Conclusions

In this proof of concept, we have shown that drafting-detection in Ironman is not an illusion and could be used in the near future in practice. This speculation is confirmed by the following facts. The Galileo GPS navigation system that will improve the precision of differential GPS is approaching the end of its construction. An evolution of the mobile network is converging into the fifth generation 5G. The explosion of ubiquitous computation is leading to the creation of specialized hardware devices with integrated GPS and HSPA features, and low-power consumption. These devices are more suitable for use in the application for drafting-detection in Ironman than widespread mobile devices, e.g., smart-phones. In fact, the similar technology is used today by TV transmissions of the greatest bicycle races in the world, e.g., Tour de France, Giro d'Italia, Vuelta a Espana. There, some competitors bear mobile devices that reflect they positions in the race on a graphic illustrating the race course and broadcast this graphic to televisions around the world.

Although we have focused in Ironman, however, this application can be employed without any changes in other triathlons as well. In future work, this real-time application would be integrated into the domain-specific language EasyTime that controls timing systems for measuring time in various sporting competitions.

References

[1] Abbott, D.: Embedded Linux Development using Eclipse. Elsevier Inc., Burlington, US (2009).

[2] Agnew, D. C., Larson, K. M.: Finding the repeat times of the GPS constellation. GPS Solutions, Springer Verlag, Berlin, vol. 11, no. 1, p. 71-76 (2007).

[3] Alonso, G., Casati, F., Kuno, H., Machiraju, V.: Web Services: Concepts, Architectures and Applications. Springer Verlag, Berlin (2010).

[4] Bosak, J., Bray, T.: XML and the Second-Generation Web. Scientific American, vol. 280, no. 5, p. 89-93 (1999).

[5] Braasch, M. S., Van Dierendonck, A. J.: GPS receiver architectures and measurements. Proceedings of the IEEE, p. 48–64 (1999).

[6] Burnette, E.: Hello, Android: Introducing Google's Mobile Development Platform. Pragmatic Programmers LLT., New York, US (2010).

[7] Chen, G., Grejner-Brzezinska, D. A.: Land-Vehicle Navigation Using Multiple Model Carrier Phase DGPS/INS, Proceeding of the 2001 American Control Conference, Arlington, VA, vol. 3, p. 2327–2332 (2001).

[8] Darcey, L., Conder, S.: Android: Wireless Application Development. Addison Wesley, Upper Saddle River NJ, US (2011).

[9] Dierendonck, A. J.: GPS Receivers. In: B. W. Parkinson, and J. J. Spilker (Eds.): Global Positioning System: Theory and Applications. American Institute of Aeronautics and Astronautics, Inc, vol. 1 (1996).

[10] Erl, T.: SOA: Principles of Service Design. Prentice Hall Inc. Upper Saddle River, NJ, US (2009).

[11] Farrell, J., Givargis, T.: Differential GPS Reference Station Algorithm-Design and Analysis. IEEE Transactions on Control Systems Technology, vol. 8, no. 3, p. 519–531 (2000).

[12] Fister, I. Jr., Fister, I., Mernik, M., Brest, J.: Design and implementation of domain-specific language EasyTime. Computer Languages, Systems & Structures, Arcticle in press, doi: 10.1016/j.cl.2011.04.001 (2011).

[13] I. Jr. Fister, I. Fister, M. Mernik, J. Brest.: Design and implementation of domain-specific language Easytime. Computer Languages, Systems and Structures, 2011, 37(4), 276–304.

[14] I. Jr. Fister, M. Mernik, I. Fister, D. Hrnčič.: Implementation of EasyTime Formal Semantics using a LISA Compiler Generator. Computer Science and Information Systems, 2012, 9(3): 1019–1044.

[15] Fling, B.: Mobile Design and Development. O'Really Media Inc., Sebastopol, CA, (2009).

[16] Gehue, H., Hewerdine, W.: Use of DGPS Corrections with Low Power GPS Receivers in a Post SA Environment. Proceedings of IEEE Aerospace Conference, Big Sky, MT, vol. 3, p. 1303–1308 (2001).
[17] Genco, A., Sorce, S.: Pervasive Systems and Ubiquitous Computing. WIT Press, Southampton, UK, (2010).
[18] Hansmann, U., Merk, L., Nicklous, M. S., Stober T.: Pervasive Computing. Springer-Verlag, Berlin, Germany, (2003).
[19] Ibrahim, D.: Design of a GPS data logger device with street-level map interface. Advances in Engineering Software, Elsevier Science Ltd., Oxford, UK, vol. 41, no. 6, p. 859–864 (2010).
[20] Ito, M., Kobayashi, K., Watanabe, K.: Study on the Method to Control the Autonomous Vechile by Using DGPS. Proceeding of the 41st SICE Annual Conference, Vol. 4, Osaka, Japan, p. 2382–2384 (2002).
[21] Kaplan, E. D.: Understanding GPS: Principles and Applications. Artech House, Boston, US (1996).
[22] Knutsen, J.: Web Service Clients on Mobile Android Devices. MSC, Norwegian University of Science and Technology, Department of Computer and Information Science, Trondheim, (2009).
[23] Misra, P., Enge, P.: Global Positioning System: Signals, Measurements, and Performance. Ganga-Jamuna Press, Massachusetts, US (2010).
[24] TM8358.2: The Universal Grids: Universal Transverse Mercator (UTM) and Universal Polar Stereographic (UPS). Defense Mapping Agency, USA (1989).
[25] Parkinson, B. W., Spilker Jr., J. J. (eds.): Global Positioning System: Theory and Applications, Progress in Astronautics and Aeronautics. American Institute of Aeronautics and Astronautics, Inc, vol. 164 (1996).
[26] Patil B., Saifullah Y., Faccin S., Sreemanthula S., Aravamudhan L., S.Sharma, Mononen R.: IP in Wireless Networks. Prentice Hall Inc., Upper Saddle River, NJ, US (2003).
[27] Perera, S., Herath, C., Ekanayake, J., Chinthaka, E., Ranabahu, A., Jayasinghe, D., Weerawarana, S., Daniels, G.: Axis2, Middleware for Next Generation Web Services. Proceedings of the IEEE International Conference on Web Services, IEEE Computer Society, Washington, DC, USA, p. 833–840 (2006).
[28] Poslad, S.: Ubiquitous Computing: Smart Device, Environment, and Interactions. John Willey & Sons Ltd., Chichester, UK (2009).
[29] Prasad, R., Ruggieri, M.: Applied Satellite Navigation Using GPS, GALILEO, and Augmentation Systems. Artech House, Boston, US (2005).
[30] Raquet, J. F.: Multiple GPS Receiver Multipath Mitigation Technique. IEEE Proceedings of Radar, Sonar and Navigation, vol. 149, pp. 195–201 (2002).
[31] Saad, A. Z.: Next Generation Mobile Communications Ecosystem. John Willey & Sons Ltd., Chichester, UK (2011).
[32] Coordinate conversions made easy: http://www.ibm.com/developerworks/ java/library/j-coordconvert (2011).
[33] Schmid, A., Neubauer, A., Ehm, H., Weigel, R., Lemke, N., Heinrichs, G., Winkel, J., Avila-Rodriguez, J. A., Kaniuth, R., Pany, T., Eisfeller, B., Rohmer, G., Niemann, B., Overbeck, M.: Enabling Location-Based Services with a Combined Galileo/GPS Receiver Architecture, Proceedings of the ION GNSS Conference, p. 1–12 (2004).
[34] Schroth, C., Janner, T.: Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services. IEEE Computer Society, IT Professional, vol. 9, no. 3, p. 36-41 (2007).
[35] Shuxin, C., Yongsheng, W., Fei, C.: A Study of Differential GPS Positioning Accuracy. 3th International Conference on Microwave and Millimeter Wave Technology 2002, p. 361–364 (2002).
[36] Soares, M. G., Malheiro, B., Restivo, F. J.: An Internet DGPS Service for Precise Outdoor Navigation. IEEE Conference Emerging Technologies and Factory Automation, vol. 1, p. 512–518 (2003).
[37] Vickery, J. L., King, R. L.: An Intelligent Differencing GPS Algorithm and Method for Remote Sensing. IEEE International Geoscience and Remote Sensing Symposium, vol. 2, p. 1281–1283, (2002).
[38] World Triathlon Corporation: IRONMAN Rules. WTC Technical Report (2010).
[39] Žalik, B.: An efficient sweep-line Delaunay triangulation algorithm. Computer-Aided Design, vol. 37, no. 10, p. 1027-1038 (2005).