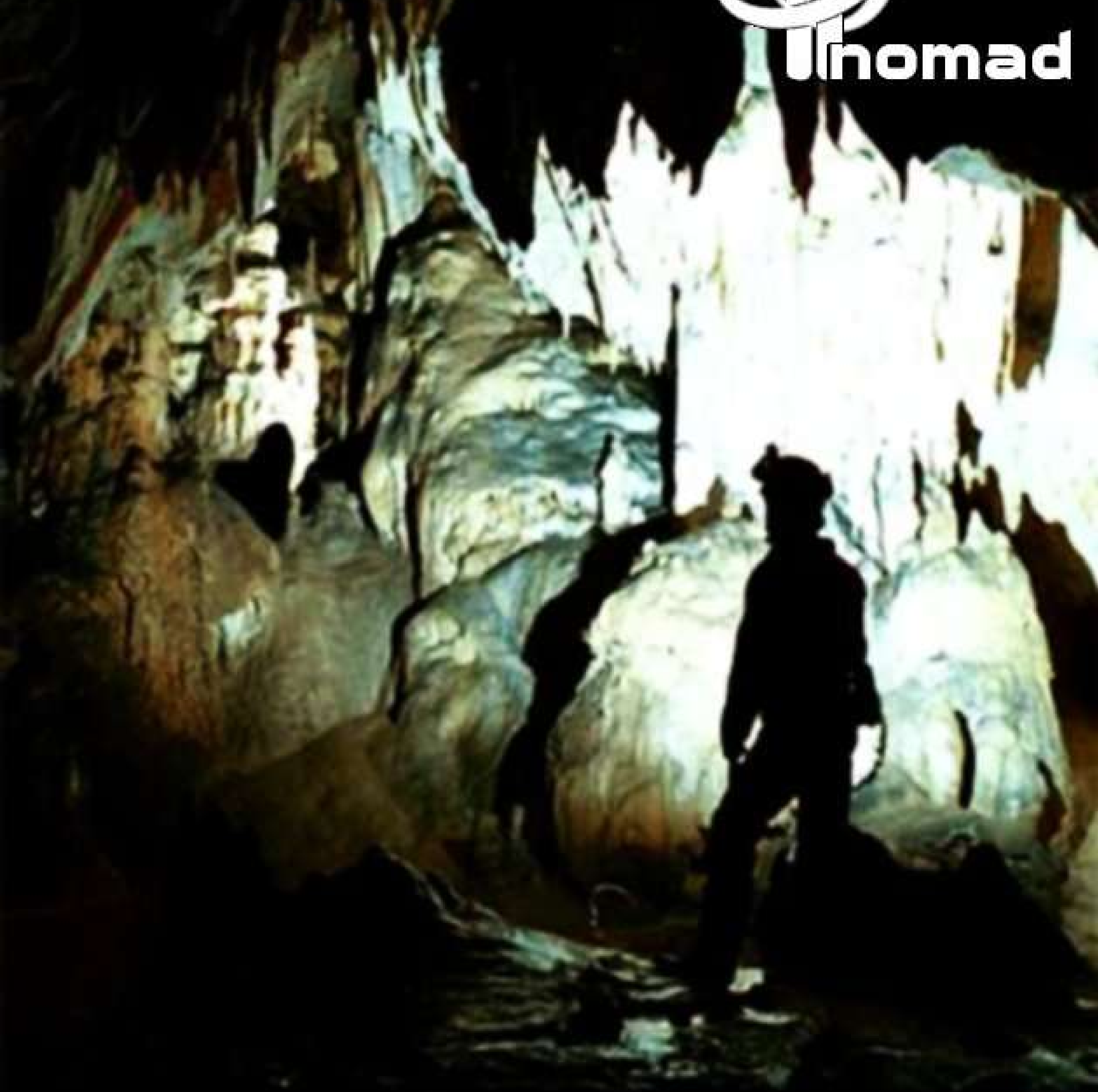
Politehnica University in Bucharest

NOMAD Positioning System

CSIDC 2005 Final Report



Team	**Doru Arfire**	**doruarfire@cs.pub.ro**
	Catalin Ioana	**catalinioana@cs.pub.ro**
	Bogdan Lucaciu	**bogdanlucaciu@cs.pub.ro**
	Marius Muresan	**mariusmuresan@cs.pub.ro**
Team Mentor	**Nicolae Tapus**	**ntapus@cs.pub.ro**

1.Abstract

In the past few years we have entrusted our orientation needs to the Global Positioning System. Although the GPS is an effective method to determine a person's location, it has its limitations – it only works in open environments and it has limited accuracy.

The **NOMAD positioning system** was designed and developed to go beyond the boundaries imposed by conventional positioning systems, providing navigation and orientation facilities when and where they are needed. It is inexpensive, completely mobile, it doesn't depend on any external systems like satellites, and it works in almost any environment, including underground locations, large buildings or busy cities with skyscrapers.

NOMAD is an user-centric system that was designed to provide position information by tracking, recording and analyzing the human movement. This is accomplished by the use of a **Motion Tracking Unit** (MTU) - a small device carried by the user that collects data from high-accuracy sensors (digital accelerometers, compass and accelerometer), processes the information and sends it over a wireless connection to the **NOMAD** Guide software that runs on a mobile platform.

The **NOMAD** Guide computes the traveled path in real-time and instantly reports visual mapping information to the user. It can be used for creating dynamic maps of unknown locations, based solely on the paths traveled by the user, or for guided navigation on existing maps.

Mine engineers, military, speleologists are often doing their jobs in the underground, and are subjected to all the specific perils of such activities. Tourists often venture in caves or other underground locations without proper maps, equipment or experience. Almost all large cities have vast underground sewage systems, where technicians perform daily maintenance activities on the various systems running in the underground. Even with a map, these are confusing whereabouts where conventional positioning systems do not work.

NOMAD gives an enriched orientation experience, being not only a navigation guide for the users, but also a mean to include additional information to the map for future analysis. Users can create multimedia maps by adding images, voice or text notes to key points on the location, turning their experience into an expedition journal.

Although the **NOMAD** system is designed to be completely mobile and independent to other systems, it can also be used in a distributed network, where collaborative users cover distinct areas of an underground system. The **NOMAD** Guide Software allows users exploring the same territory to periodically synchronize their **NOMAD** maps, sharing the team experience and observations.

Another aspect of the **NOMAD** System are the on-line communities, meeting grounds for possible underground explorers. This feature is accomplished by the Online Repository, a web-based application that allows users to publish, download and review maps directly from the **NOMAD** Guide, or from a normal web browser on their desktop computer.

2. System Overview

The **NOMAD** system consists of two hardware components that communicate through a simple protocol: the **Motion Tracking Unit** and the **NOMAD Guide**.

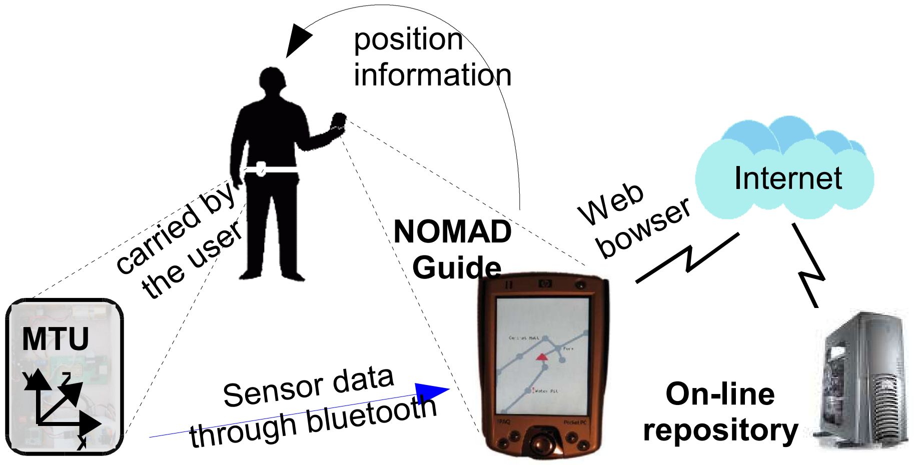


Figure 2.1. - Nomad System Overview

The Motion Tracking Unit is a custom device that gathers, records and analyzes the movement of the user.

The data collected by the MTU is transmitted to the **NOMAD** Guide. This software uses this information to build a 3D representation of the path taken by the user, and displays it in real-time to the user. The format used to store the maps in allows the user to add meaningful information to certain sections of the path.

The **NOMAD** System covers most navigation and orientation related needs in underground systems:

- **Topological survey –** tourists or specialists can use **NOMAD** to build and share dynamic maps of underground areas. These types of maps could be used for searching, orientation or they can be used as a base for further accurate cartography.
- **Orientation –** existing maps can be used for reaching a certain key location in the system. Another important feature would be finding the way back to the entry point, by using the path dynamically created by **NOMAD**.
- **Searching –** archaeologists are looking for various artifacts in closed places, geologists and speleologists are searching for rock formations in caves and rescue teams are looking for lost persons in mines or other underground areas. The searched area may be completely new to the user, or an existing map can be used as a search guide.

2.1. Performance Requirements

The performance constraints are given by the mobile capability, accuracy and real time features of the **NOMAD** System. All the chips used in the system where chosen to meet the power consumption constraints, all of them are low power, especially the Bluetooth wireless module.

For the real-time constraints there are two requirements, the communication module has 0.3-0.8 seconds latency, while the processing is done in less then 0.5 seconds. Another important constraint is given by the sensors accuracy. **NOMAD** can detect 1 meter altitude variation and a maximum of 1.2g acceleration.

2.2. Design Methodology

An incremental spiral model was used due to its flexibility and the need of constant testing. Throughout the life cycle, the project had three releases, each representing a functional version of the **NOMAD** System.

Each release was planned to have 3 main phases: design, implementation and testing. The first two releases consisted mainly of adding software modules and developing the MTU device; while in the 3rd release added final features and interface the MTU with the software modules.

Team members worked in pairs in order to reduce implementation errors and to enhance creativity. The tasks were shared among team members, based on personal experience, knowledge and skills, in order to cover successfully both the areas hardware and software.

Each phase started with a meeting in which the output of the new release was specified. In order to verify that these specifications are met the team also drafted a set of tests, one for each distinguishable feature of the product.

2.3. Innovation

While conventional positioning systems use satellites, **NOMAD** is a truly mobile system, relying solely on movement tracking equipment attached to the user. Actually, **NOMAD** puts the user in the center of the whole system: the user's movement is recorded and processed in order to obtain actual location information, and this information can be enriched and updated by the user through a innovative mechanism of adding sensible information to a map: meta-information.

Meta-information is an abstraction that allows the user to add information to a map enriching it with specific details, according to the user's activity and needs. For example, a team of speleologists can add information related to stalactites, stalagmites and other rock formations, while a rescue team would use information related to dangerous sections of a cave, located tourists, water pits.

Unlike classic underground orientation methods where at least two persons consulted paper maps while traveling in dark and unfriendly environments, **NOMAD** navigation was designed for minimal user interaction. Using paper maps is error-prone, difficult and distracts the users' attention from their main activity.

The **NOMAD** also works as a real-time journal of the path traveled by the user. This, combined with the ability to add domain specific information to the map, offers a novel way to travel and keep track of your way back with minimum effort.

Topological information gathered with the **NOMAD** System can be shared with other users by using the services of On-line Repositories. Underground enthusiasts can share their experiences and create veritable on-line communities.

3. Implementation

3.1. System Description

3.1.1. Functional Overview

NOMAD builds a dynamic map of the underground system, based solely on the user movements in the system. By analyzing direction, altitude and speed variations, it generates an approximate map that reproduces the route traveled by the user.

The acquisition of these variations is done by the Motion Tracking Unit (MTU) , that is rigidly attached to the user, most likely with a belt. These variation are periodically beamed to a mobile computing unit (like a Pocket PC) where they are interpreted and converted to useful location information by the **NOMAD** Guide Software.

The user can also set key points on the route (near a dangerous or interesting object), with additional meta-information attached, like images, text or voice notes. The user can contribute to the system accuracy by marking more key points to the map.

These maps can be stored in a centralized repository, and can be synchronized between team members navigating in distinct areas of a very complex system.

Although the primary use of **NOMAD** is aided navigation, it can be easily connected to additional modules (induction radio underground communication systems, air-quality testing units, etc), but it's very important that the base MTU remains as light and non-intrusive as possible.

3.1.2. System Components

From a hardware point of view the **NOMAD** system consists of a custom device, the Motion Tracking Unit (MTU), a mobile application, the NOMAD Guide, and a web portal, the **NOMAD** Online Repository, that acts like a central repository for sharing maps between users. The MTU and the **NOMAD** Guide communicate through a serial connection, on either a conventional serial cable or using the Bluetooth technology. The **NOMAD** Online Repository can be accessed either through a web interface or directly with the **NOMAD** Guide, using a wireless Internet connection.

The **Motion Tracking Unit** is a small custom-made device that is carried by the user. Its purpose is to collect data regarding the user's movement, process it and transmit it to the **NOMAD** Guide for further analysis, error correction and storage. In order to achieve this the MTU must be attached as rigid as possible to the user's body, in a well-known position, preferably to his belt. It measures the user's acceleration, the orientation relative to the Earth's magnetic field and the current atmospheric pressure.

The **NOMAD Guide** is an application that runs on a conventional mobile platform, a Pocket PC for example, that receives information about the motion of the user and processes it to build a visual model of the path traveled so far. It accepts input from the user in the form of additional information about the path, and controls the functioning of the MTU. The **NOMAD** Guide also allows the user to perform several adjustments to the managed maps, like map merging (building a single map from several smaller maps representing sections of the same region), map synchronization between several **NOMAD** Guides.

The **NOMAD On line Repository** is a web-based application that allows the sharing of maps built using the **NOMAD** system. The interface to this repository can be either a conventional web browser or the **NOMAD** Guide, for fast map uploading/downloading.

3.2. Motion Tracking Unit Design and Implementation

The Motion Tracking Unit consists of the following modules:

Sensor module - a hardware module responsible with the measurement of a certain parameter of the user's motion. The current implementation supports three such modules that measure the following parameters: the instantaneous acceleration of the person on three axes, the person's orientation relative to the Earth's magnetic field and the atmospheric pressure at the current location of the user.

The projections of the user's acceleration on three predefined axes - the back-front axis (X axis), the down-up axis, parallel with the user's backbone (Y axis), and the left-right axis (the Z axis) – are measured and transmitted to the core module. Because of this definition of the axes on which the acceleration is projected, the user is required to specify the position of the MTU on his belt: front, back, left or right. The value of the acceleration is used to compute the approximate distance traveled by the user. This is done by detecting when a step has occurred and then measure that step's length.

The measurement of the person's orientation is performed using a high precision digital compass, while for measuring the atmospheric pressure a pressure cell was used.

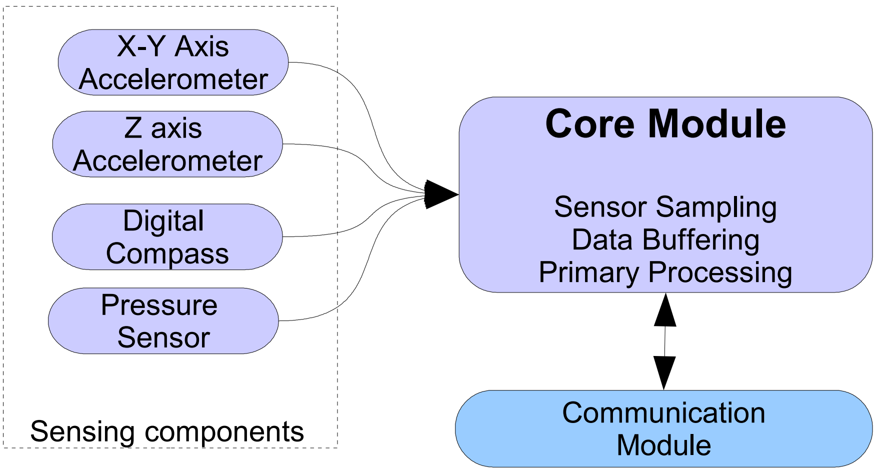


Fig. 3.2.1 – The Mobile Tracking Unit

MTU Core Module - it is responsible with the control of the MTU and the synchronization with the Sensor Module in order to obtain real-time information about the characteristics of the user's motion. It also implements a Step Detection Algorithm which analyzes raw data from the accelerometers in order to determine various parameters of the user's motion and yields a stream of data structures that encode information about each step performed by the user.

Communication Module - The Communication Module implements the Communication protocol with the mobile guide. It acts like a Slave device, listening for commands issued by the NOMAD Guide. It supports two types of communication technologies: a common serial cable and a Bluetooth connection. A number of parameters related to the way sampling is done are set by this module, according to the commands received from the Guide.

3.2.1. Hardware Components

In order to met all the functional requirements the MTU hardware design consists of a microcontroller, two accelerometers, a digital compass, a pressure sensor and a Bluetooth controller.

The role of the microcontroller is to communicate with all external sensors, gather information, make some preliminary data processing and then communicate with the PDA using a wireless Bluetooth link. It was used a microcontroller with the following features: 16 bit counter capture module, a SPI port, a high availability, low cost and low power consumption – an important issue for a mobile device. Equipped with a fast quartz crystal, the speed requirements for processing sensors output and communication were fulfilled.

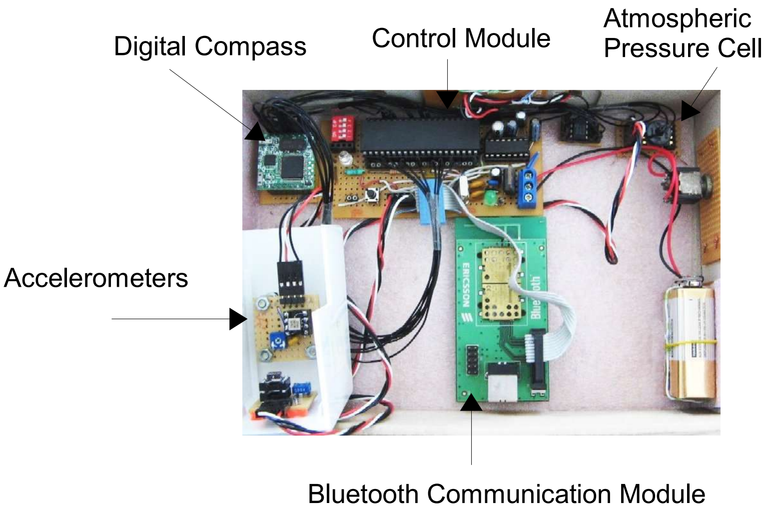


Figure 3.2.1.1 – Inside the Mobile Tracking Unit

For a 3D image of the way a person moves, a 3 axis accelerometer was needed. It was chosen a 2-axis low g accelerometer, with a range of +/-1.2g. The MTU would be firmly attached to a belt, and so, the accelerometer would indicate a person movement in an usual walking or running, which will not require more than

1.2g. In order to provide 3 axis orientation two perpendicular accelerometers were mounted. Some great features were: the digital interpretable output (duty cycle modulated), the high sensitivity and accuracy, the low cost, low power, signal conditioned and small size.

The magnetic sensor is required to measure the angle between the North-South axis and user direction of movement. A few magnetic sensors were tested, but they were inaccurate and lacked stability over time, although several methods for compensating their outputs were used. V2xe digital compass was used because is an integrated 2-axis compass and magnetic field sensing module featuring an on-board microprocessor for control and interfacing. Another important feature of V2Xe is incorporated temperature and noise stabilized oscillator/counter circuit. This means that the digital compass has a great linearity, gain and resolution measurements. The low power drain and small package size were some great additional features which the V2Xe also met. The measurement is stable over temperature and inherently free from offset drift. Regarding the measurements, the compass is interfaced with the microcontroller, taking advantages of the SPI protocol. Using a binary based protocol different parameters of the magnetic field can be measured. The most interesting one is heading, which represents the angle between the North and the user's moving direction.

The atmospheric pressure sensor is used for finding the user’s current altitude. This type of sensor can offer very useful information about climbing or descending of the user. The output of the sensor is the atmospheric pressure, which can be used to obtain the altitude using a formula similar with the one used for altimeters. The air pressure has a very low variation over altitude[1], so a high accuracy pressure sensor was needed. MPX6115A was used because it integrates on-chip, bipolar operational amplifier circuitry and thin film resistor networks to provide a high output signal and temperature compensation. Other characteristics of the sensor are: the measurement range adjusted for the usual atmospheric pressure, a fast response time and also as the increased offset stability. The small form factor and high reliability of on-chip integration make the MPX6115A pressure sensor a

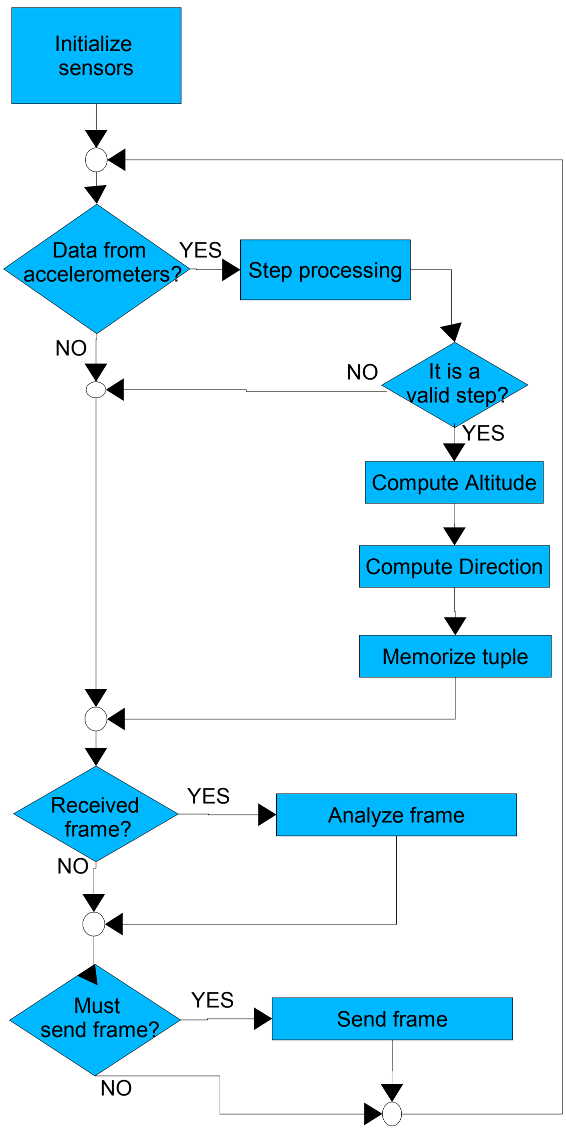


Figure 3.2.1.2 – MTU Work Flow

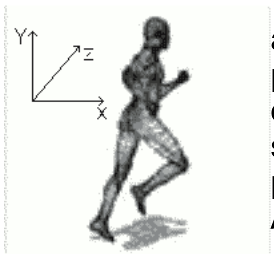
logical and economical choice for **NOMAD** System.

Communication between MTU and the PDA is made by serial cable or by bluetooth. Bluetooth is now a feature integrated in many PDA devices or there is the possibility of adding bluetooth capabilities through an extension card (CF, SDIO card). Another reasons for using bluetooth were the speed of 115.2kbps which is enough for MTU device and the low power consumption. Another way of communication is by serial cable.

The firmware software describes the way the three hardware entities: sensors module, primary acquisition module, storage and communication module function and interact. All this modules are being proper interfaced and the firmware is controlling them. The data flow diagrams summarize the way the information is received from the sensors, processed and memorized by the microcontroller and sent by the communication module.

3.2.2. Step Detection Algorithm

In order to provide sensible information from which the **NOMAD** Guide can measure the distance traveled, an algorithm was devised that determine steps taken by the user and for each step, measure quantitative parameters in relation with the length of the step.

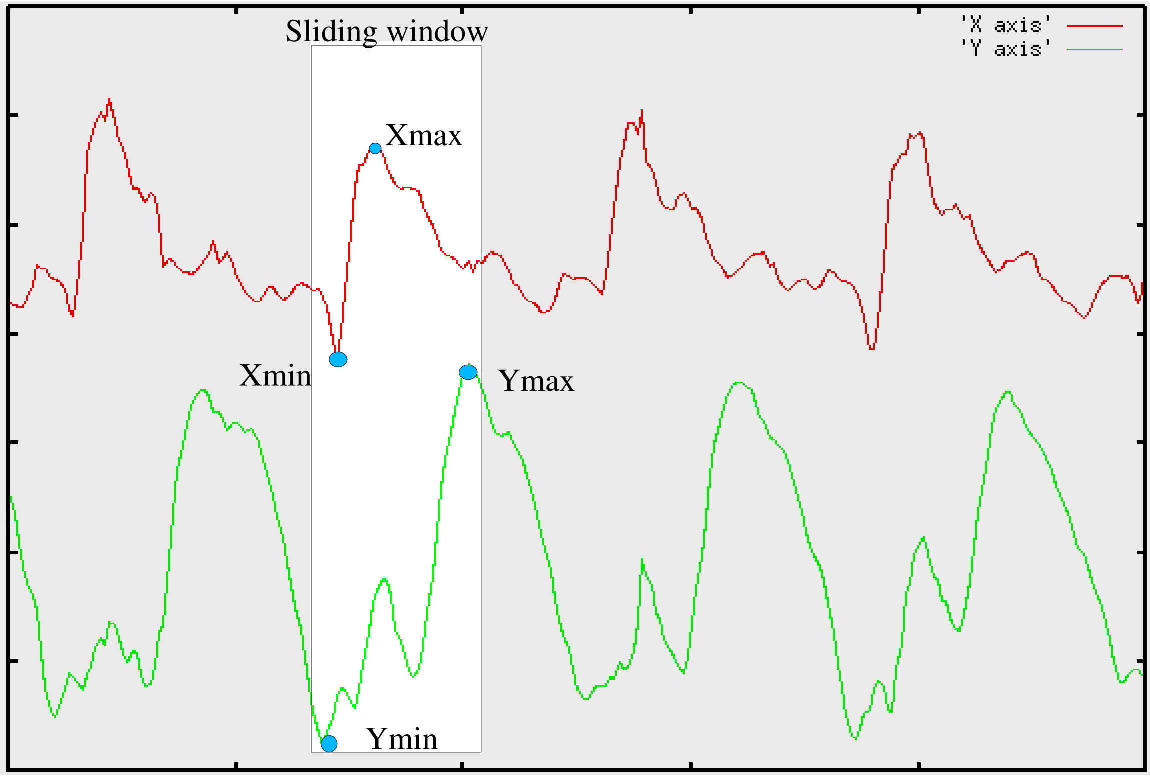


This algorithm relies only on the outputs of one of the two accelerometers. These two outputs correspond to two perpendicular axes, one oriented upwards (the axis), and the other one oriented parallel with the walking direction. As you can see in the following figure, the two accelerations follow regular patterns, depending on whether the user is walking or running. Also, the step length and duration varies.

The step detection component of the algorithm relies on the detection of the minimum and maximum values of both the considered accelerations. These form the “signature” of the step: a tuple of 4 values {X_{min}, X_{max}, Y_{min}, Y_{max}}, consisting of the minimum and maximum values of the two accelerations. This detection is done with only one pass, so it is possible to implement it so that the Motion Tracking Unit will detect in real-time the steps taken by the user, and then transmit only this tuple to the **NOMAD** Guide, that will proceed with calculating the actual step distance.

The algorithm uses a “sliding window” containing the last values for the considered accelerations. The sliding window starts with a minimum or a maximum value and extends until the first matching maximum, respectively minimum value, is received.

In order to avoid confusing a local maximum to the maximum value of a acceleration, there is the condition that the extreme values of an acceleration have to be symmetrical in report to the zero value, within certain limits. These limits are chosen close enough that they would filter out other local maximum/minimum values, and wide enough that they would allow a plausible increase/decrease in speed by the user. These considerations apply only to the Y acceleration, since only this is used to recognize when a step has occurred.



3.2.2.1 Sliding window

After determining a step, the maximum and minimum values of both accelerations are stored and the algorithm starts looking for the beginning of the next step. This means sliding the window over the second half of the step waveform, and it is done in a similar fashion as for the first half. The only difference is that the first value is a Y acceleration maximum and the last value must be the next minimum.

One last requirement for this algorithm is robustness, meaning that in case of unusual acceleration patterns, it must recover and look for the first valid step waveform. In order to achieve this two methods were applied . The first method uses the value of the acceleration on the third axis (Z) to determine whether the user's position is different from the usual walking position. This way it can be determined if a user is bending or sitting, and the step counting algorithm can be interrupted. The second method tries to determine the first minimum-maximum pair of the Y acceleration, and resumes the step counting procedure from that moment.

3.3. Software Design and Implementation

The main components of the **NOMAD** Guide Software are:

Map Management and Control– responsible for managing information about the current map. It also handles map merging and the addition of application specific information to the map. Because of it's central position in the system, it also serves as the control module for the **NOMAD** system.

MTU Interface – responsible with implementing the protocol used to communicate with the MTU. It serves as a Master to the MTU, controlling the rate at which the MTU would send brute information to the PDA, as well as setting a number of MTU parameters.

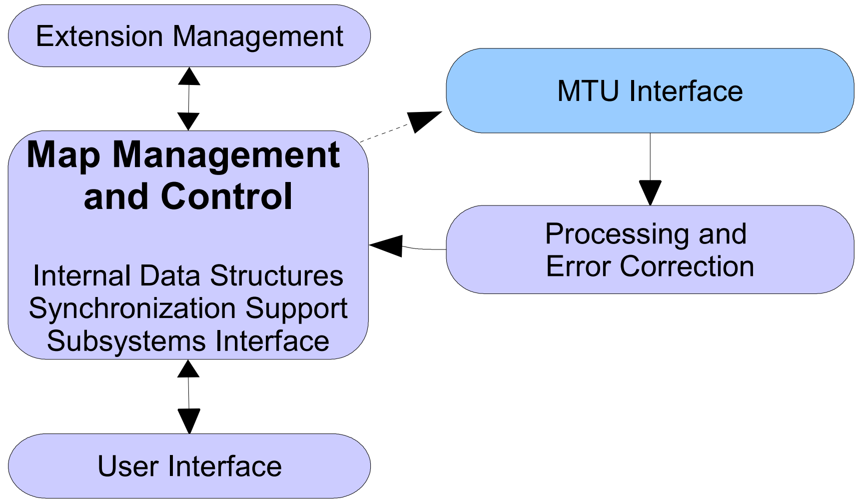


Figure 3.3.1. - The NOMAD Guide

Processing and error correction – applies certain error correction methods on the data used to build the map. It also implements a Distance Measurement Algorithm that determines the length of every step taken by the user, using the step information provided by the MTU. The Processing and Error Correction Subsystem splits the point information received from the MTU into map building blocks, like nodes and segments.

User Interface – responsible with displaying the map in an easy to follow graphical format. It also allows the user to set the **NOMAD** system parameters and add sensible information to the current map managed by the **Map Management** Component.

Extension Management– it implements a server interface to other applications. This interface can be used to include the functionality offered by the **NOMAD** system in other applications.

These components are implemented as separate threads that communicate in an event-driven fashion.

3.3.1. Distance Measurement Algorithm

The Processing and Error Correction Component of the **NOMAD** Guide Software implements an algorithm that computes the length of a every step detected by the Step Detection Algorithm on the MTU. It uses the 4-values tuple issued by the latter and computes an Euclidean distance from this tuple to the tuple corresponding to the absence of movement: $\sqrt{(Y_{max}-Y_0)^2+(Y_{min}-Y_0)^2+(X_{max}-X_0)^2+(X_{min}-X_0)^2}$, where Y_0 and X_0 are the values of the accelerations on the Y and X axes at the equilibrium.

The distance computed like this serves as a hash into a table of precomputed step lengths and the length of the current step is taken to be the step length associated to the value closest to the distance already computed.

In order to fill in the table the step lengths an their associated tuples of 11 persons have been measured, and for each person 4 walking speeds have been analyzed.

3.3.2. Map Structure

The MTU transmits data to the PDA in form of a tuple {A, B, C}, where A represents data from which the length of the last step can be retrieved, B represents the absolute orientation given by the digital compass and C is a measure of the altitude. Because the distances are fairly small, the path can be approximate accurate enough by using a list of oriented horizontal segments.

Given the fact that the raw data from the MTU represents only information about steps taken by the user, and that following the same path would most certainly not yield the same steps, the **NOMAD** system doesn't apply any interpolation on the unstructured data from the MTU, but builds on top of it another more abstract vision of the map.

The key concept of the map structure is the concept of zone. A zone is a cluster of points in a small area of a path. .There are three main types of behavior that establish a zone:

- The user spent a certain amount of time in the same small area, either by not moving at all, or by walking around.
- When the user changes direction the corners of the path are captured
- The user specifically designated the current position as a zone.

The reason for this approach is the need to associate certain pieces of informations relevant to the user to some sections of a map.

Internally, a zone is represented in a XML-like format, encapsulating all its points. The segments connecting the zones are represented in a similar fashion. This format allows an easy manipulation and, because it is mainly a text format, can be efficiently compressed and stored.

According to this definition, a path is a chain of zones connected by straight lines. The concept of zones is very important in adding meta-information to a path. Meta-information represents any useful piece of information that the user needs to add to the map, either while the map is built, or afterwards. Meta-information can be: certain text descriptions, icons, sounds recorded on site, other application specific information and other data internal to the system. Internally meta-information is represented as separate XML tags attached to a zone. Considering their importance to the core functionality of the system, there are two types of meta-information: system-oriented and user-oriented.

System-oriented meta-information defines certain characteristics of the map and allows the user to adjust the map to its own perception on the reality. There are three very important tags that allow the user to define the shape of the map: New-Path Tags, Path-Fork Tags and Fix-Point Tags.

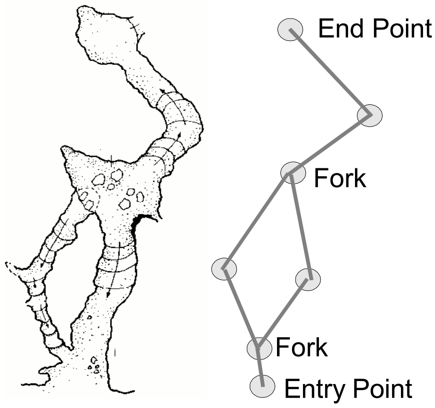


Fig. 3.3.2.1 – NOMAD Mapping example

A New-Path Tag marks a zone where a new path is attached to the current path. It is used when merging two maps into a single map, and it defines the zones where these maps glue together. A Path-Fork Tag marks a zone where a path fork has occurred. It works like multiple New-Path Tags attached to the same zone. The new paths attached to the current path are left pending and can be filled in later. A Fix-Point Tag allows the user to specify that a certain zone has a known position in GPS-like coordinates. This provides for certain corrections to section lengths along the path, proportionally to the elasticity coefficient of those sections.

Given two paths constructed either by a single user or by two different users, those two paths can be joined to form a map. For this, one or two ends of a path (the ends of a path are zones too) will be merged with one or two zones of the other path, respectively. The merging operation means that a special piece of meta-information specifying that there is an alternative path emerging from that zone will be attached to the zones targeted to be merged with the ends. This operation is not restricted to the ends of a path, and can be applied even to zones of the same path.

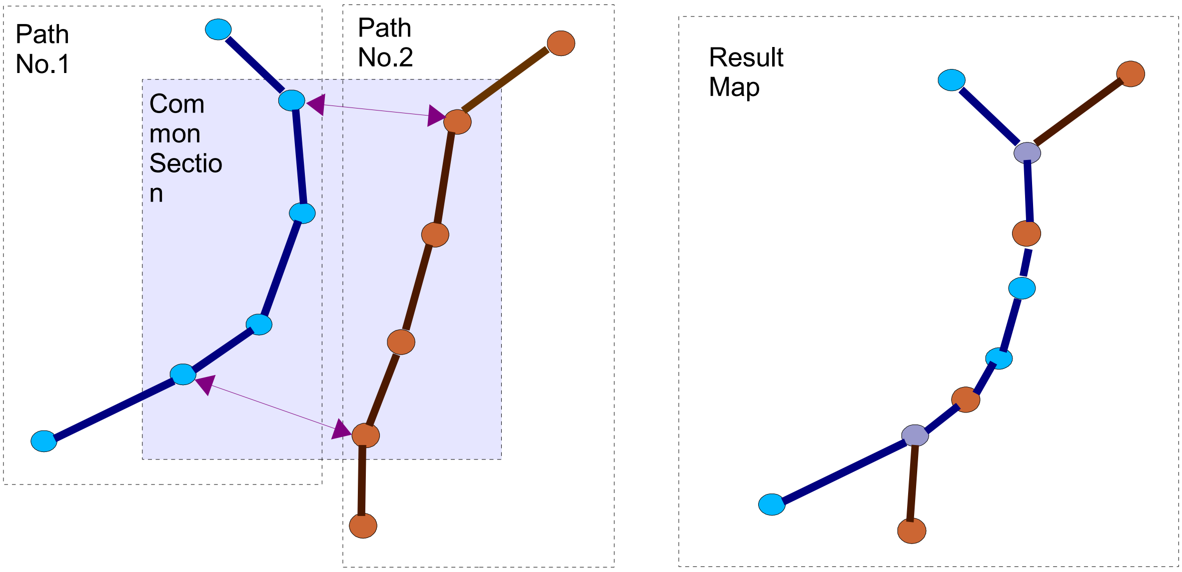


Fig. 3.3.2.2 Path merging

One other type of editing that a user can apply on a map is path merging.

This means that a user has detected that by merging two maps there are some paths that are duplicated, because they have been included in each of the two maps. To eliminate this anomaly the user can apply the path merging procedure.

This procedure consists in replacing the two representations of the path by only one of them, while preserving all the user-added information of the initial paths. The only exception to this is the case of Fork-Tags: since the user knows that the two paths are the same, it is reasonable to assume that there will be duplicate Fork-Tags, identified by the same number of paths that join at that point. So using a simple linear algorithm, the **NOMAD** system will identify corresponding Fork-Tags in the two paths and prompt the user whether these should be merged or not.

3.3.3. User Interface

Although the **NOMAD** Guide has many options that might need user interaction, a simple interface was designed, hiding the advanced features. Most of the time the only thing needed on the Pocket PC screen is the rendered map providing positioning information to the user. The user needs to have know his location, options and exit paths at all times. One should have easy access to important features like the interactive navigation guide (to a specific zone or an exit point) or the zone meta-information options. Other features are available only on request, because test case usage behavior showed that a clobbered and complex interface is confusing, and there is a strong possibility the **NOMAD** system will be needed in unfriendly environments, where users might panic if they feel they don't know their exact position in the system.

Zones and paths in the map can be associated with the real system, because the user can always view available paths, dead ends, zone icons and labels and other meta-information

The interface was created by complying with Windows Pocket PC User Interface Guidelines, and some research was also conducted with possible users of the **NOMAD** system. Useless parts were removed and other features were added on users' requests.

The rendering is implemented using the computer graphics features offered by the compact .NET framework. Various optimizations were implemented, so the user can view a dynamic and accurate map of the system:

- **zone caching** - the visual representation of zones is only calculated if it's needed - when the zone is created or changed.
- **interpolation and smoothing** - some irregular patterns created by the user's unusual movement may confuse him or others later. The paths drawn on the Pocket PC screen are smooth and easy to follow. This feature was carefully implemented so it would not affect the system's accuracy.
- **render caching** - some sections of the map can be fairly complex, so as much as possible (memory-wise) paths and zones are only rendered when the map is updated

The simple start screen does not imply limiting the user's experience. Users can still modify their maps in any way, they can change the structure by importing and merging paths or entire maps. The system itself is flexible and the maps are not just geometric representations of the location, they are a sum of information that describe the map in every aspect, containing the user's position history, his own changes, additions and other customizations, and most importantly - the associated meta-information.

The created or modified maps can also be saved on a storage device to be archived, reused and shared with others. The output is stored in a packed format, containing not just the map structure but also the additional meta-information (sound, images). The user can opt-out for some of the information that was used only in one session and are no longer necessary. A packed format (versus a collection of XML and media files) was chosen for greater usability. The saved map can be later loaded into the **NOMAD** software when the same location is visited again.

Besides storing the maps on disk, users can use the bluetooth connection on their PocketPC to synchronize their maps with other team members if there is a need for a distributed search.

User Profiles

Another exciting NOMAD feature is the possibility to have an adaptive style, behavior and knowledge library based on the user's profile. The initial NOMAD Guide comes with a sample profile – the Speleologist Profile but others can by added by users or interested companies. Studies conducted reveal that different types of possible NOMAD users have different tasks, terminology and work with different systems . Creating a system that offers facilities for every type of user is possible, but would induce a degree of complexity that is clearly not acceptable. Anything less means providing a crippled usage experience for some users. User profiles is were implemented to cover this issue.

The default user profile uses normal English terms that everyone understands and has a simple resource library that should be enough to add meta-information to any zones on the map. Switch to the speleologist profile and stalactites, flowstones and limestones are available in the resource library, paths are called galleries and so on.

NOMAD profiles can be changed and created, then shared with others via an online repository. Profiles can be exported and imported, and they are automatically embedded in the map files.

Profiles affect many parts of the software system, altering the terminology, resource libraries and rendering system. The idea behind profiles is having an adaptive system that gives the users precisely their needed features, and this is reflected all throughout the **NOMAD** Guide.

The NOMAD Online Repository

The **NOMAD** Repository is a web-based application allowing users to share maps, profiles and resource libraries. Any registered user can browse the repository, upload, download and review components available online.

The repository can be accessed via a simple web browser, or directly from the **NOMAD** software. This feature can be used to download maps based on the GPS location, report changes for existing maps or publishing new ones.

The **NOMAD** repository can also be used to find and contact users with better knowledge about the location, or read their descriptions of underground locations.

Expeditions can be organized using the online community options. **NOMAD** Repositories will eventually grow in size and offer large databases of **NOMAD** maps, some converted from old paper maps, others created on-thy-fly by **NOMAD**'s cartography options.

Customized **NOMAD** repositories can also be used as an internal company archive. Custom company software components can be attached to the **NOMAD** Guide in order to provide extended, company specific features to the users.

3.3.5. Communication Protocol

A very important aspect of the **NOMAD** System is the communication protocol between PDA and the MTU device. The communication session is peer to peer because every PDA is receiving data from a single MTU device and vice-versa. The communication protocol uses acknowledge frame type, in response for every frame sent, for ensuring the packages are understood by the receiver. Another important characteristics is that the MTU sends data only as a response to the PDA demand. The PDA may request one or many pieces of information from the MTU. If more steps are demanded, they are sent as soon as they are processed.

The frames sent over the connection have the following general format:

1 byte	1 byte	<DataLength> bytes	1 byte
Type	**Data Length**	**Payload**	**Check Sum**
0 7	8 15	16 x	x+1 x+8

Description of each field:

- Frame Type – represents a code for the frame, recorded in a 4 bit address space; it also has the meaning of a specified command to be performed by the MTU;
- DataLength – represents the size in bytes of the Payload field;
- Payload – represents the useful data; this field has a maximum length of 255 bytes, which is enough for data transfer between MTU and PDA
- Checksum – represents the sum of control calculated on the Payload; it has the role of signaling any mismatched received data;

PDA commands:

- Send next tuple of values - PDA is requiring the next available tuple of values (representing the current coordinates)
- Send all the available tuples - PDA is requiring all the available tuples, which have not been sent by now
- Send the tuples grouped by <n> - PDA is requiring the tuples sent in groups of <n>. This type of frame is sent only once, but the tuples are sent periodically, as soon as they are available. The value of <n> is required in the data field.
- Reset the tuple sending - resets all the automatically sending of the tuples
- Set accelerometer sampling options – the payload of this command specifies a number of parameters related to the way the sampling is performed
- Reset - resets the MTU

MTU Replies:

- Send <n> tuples - MTU sends <n> tuples; <n> is sent in the Data field
- Buffer full - MTU buffer is full, so in order of not losing any data, the PDA should request some values
- Tuples lost - if the MTU has overwritten some tuples, because of the space lack, it sends a frame indicating how many tuples were lost;

- Static user - This type of frame indicates the fact that the user is not moving; it additionally contains information about the orientation of the user, so that the NOMAD Guide can display real-time information on the user's motion even if this is not walking, but merely turning (admiring the landscape, for example)

3.4. Testing and Verification

The testing and verification of the **NOMAD** system has happened and evolved in parallel with the implementation. At the beginning of each of the three release cycles the testing procedures to be applied at the end of the cycle would be specified, and some of the test programs needed would be developed. These testing procedures have been compiled into three documents, one for the MTU, another for the **NOMAD** Guide and a third one for the system as a whole.

Testing the MTU

Testing procedures for the MTU involved testing the sensors in order to make sure they work as specified, as well as measuring their accuracy. Also a number of test cases have been developed in order to test the functioning of the step detection algorithm.

This supposed recording different walking patterns over a short distance (up to 100 ft) of different persons and determine the failure rate of the algorithm on each of these test cases. The failure rate was defined as: $\frac{\textit{number of unrecognized steps}}{\textit{total number of steps taken}} \cdot 100(\%)$. It was noticed that the maximum failure rate was of 6%, in case of a very irregular running pattern.

Testing the NOMAD Guide

The NOMAD Guide was tested initially using random generated plausible sequences of steps in order to emulate the functionality of the MTU, which at that moment was not ready. These sequences were transmitted to the PocketPC running the NOMAD Guide through a test application running on a conventional PC. This enabled us to verify that the User Interface Module worked as designed and also that that the Communication Protocol on the software side was functioning.

The separate threads of execution of the NOMAD Software Subsystem enabled us to test them separately using test programs that simulated the rest of the system. The advantage of this method was that it enabled us to embed into the test programs several formal program verification mechanism, the most common of them being invariants over the data-flow. One common class of errors that these kind of testing enabled us to detect were thread synchronization errors.

Testing the NOMAD System

The testing of the **NOMAD** System consisted of testing the connectivity of the MTU and the **NOMAD** Guide and their functioning as a whole. This could only be done after the communication protocol was implemented on both components. The map building functionality was tested in already mapped areas, walking on marked paths. The size of these maps grew as different calibrations of the **NOMAD** system parameters were performed.

Tools developed for testing the NOMAD system:

Motion Tracking Unit Simulator - This tool is a software application developed using the Win32 API. It is used to simulate the functionality of the MTU for testing the communication protocol and other base features of the NOMAD Guide. It implements the communication protocol on both a conventional serial cable and on a point-to-point Bluetooth connection. It also includes a generator of plausible random sequences of steps, used to test the map displaying features of the NOMAD Guide. This simulator includes invariants about the communication protocol that allow us to asses the correct functioning of the communication.

Software Modules Verification Programs - This are a class of programs developed in order to verify the correct behavior of the NOMAD Guide components. The fact that each of these components are implemented as separate threads allowed us to design this test programs to simulate the rest of the system. Invariants over the states of the system were defined in order to verify the correct functioning of the software components.

3.5. Economic Considerations

Since this system is targeting a large user base, one of the requirements of the system is that it is cheap and affordable for as many potential users as possible. This would enhance the experience of users adventuring in underground or unmapped areas, and would possibly make the world a safer place. Only the price of MTU is essential, since the NOMAD Guide Software can run on many types of portable computing device with Microsoft operating systems. The MTU's price is directly influenced by the price of the sensors used for measurements. The other parts represent computing, storage and communication components, and their total price should stay below 30$.

Sensor modules used:

two 2axis high precision accelerometers	$24.00
1 digital compass module	$55.00
1 high precision pressure sensor	$20.00

Thus, the total cost of the prototype, including the processing and communication modules, should stay below 150$, and in mass production the price should drop to less then 60%.

The primary group targeted by this device is that of tourists interested in exploring unmapped or poorly mapped places. These tourists could be interested both in the orientation and positioning features of the **NOMAD** Guide but also in forming a community for sharing and improving maps built using the **NOMAD** System.

Another group interested in **NOMAD** may come from the scientific community, in particular the archaeologists, geologists and speleologists, and could find the system very useful as a real-time journal of their scientific expeditions. These groups can organize themselves as separate communities to share scientific maps built using **NOMAD**.

Last but not least, search and rescue team and technical crews working underground can use the orientation functionality in order to guide themselves through environments where no other easy to use orientation method is available.

4. Summary

The **NOMAD** System was designed with the idea that the boundaries set by conventional orientation and positioning systems have to be broken. It is working in an novel way that it uses only the Earth's magnetic field and the user's movement to construct in real time a visual representation of the path taken by the user.

Furthermore, the **NOMAD** System allows the user to add information to various sections of the map in different formats (text, sound, video, etc.), enhancing his traveling experience and acting like a real-time journal. It records paths, information about the map structure and about the characteristics of the environment from the user's point of view.

The **NOMAD** System consists of three components: the Motion Tracking Unit, the NOMAD Guide and the **NOMAD** Online Repository. The Motion Tracking Unit (MTU) is a custom device that collects data about the user's motion, packs it and transmits it to the **NOMAD** Guide. The **NOMAD** Guide is a software application running on a conventional mobile platform that receives the data collected by the MTU and uses them to build a data structure representing the map of the path traveled by the user. It also serves as a interface through which the user can follow or edit the map, or enhance it using pieces of information related to his activity.

There are several directions in which the **NOMAD** System can be enhanced. Additional sensors like humidity sensor and gyroscope can be interfaced with the MTU in order to provide more parameters to the **NOMAD** Guide for higher precision.

Also a GPS receiver could be interfaced with the **NOMAD** for usage in insufficiently GPS-covered areas, like cities or mountain valleys. This would use the GPS coordinates as fixed points where the coverage is acceptable and the **NOMAD** method of map building in other areas. Given its high precision on small areas, the **NOMAD** can also serve as an improvement to the GPS, reducing its error.

If the mobile platform that is used to run the **NOMAD** Guide features or supports a digital camera, then pictures taken on site can be added to different sections of the path, greatly enhancing the orientation and recognition capabilities of the system.

The target market of the system is that of tourists interested in exploring unmapped or poorly mapped locations. These can improve the system by using the **NOMAD** Online Repository to share maps built by them with the community of people using **NOMAD**.

NOMAD can also be used by groups of professionals, either from the scientific community (archaeologists, speleologists, geologists), by search and rescue teams or by technical crews working in underground environments. These groups may not form a community as a whole, but a group of communities. They also can enhance the system by interfacing it with applications specific to their domain of activity that can use the services provided by **NOMAD** (orientation, way-back, positioning).

NOMAD started from the idea of offering the user maximum orientation capabilities in order to allow him to explore the world around him and break the boundaries of the unknown.

5. References

[1] Microchip MPLAB IDE and MPLAB C18

http://www.microchip.com

[2] Microchip PIC18F452 microcontroller datasheet

http://ww1.microchip.com/downloads/en/DeviceDoc/39564b.pdf

[3] Microchip MCP3301 - 13 bit AD converter datasheet

http://ww1.microchip.com/downloads/en/DeviceDoc/21700b.pdf

[4] Analog ADXL213 accelerometer datasheet

http://www.analog.com/UploadedFiles/Data_Sheets/658913561ADXL213_0.pdf

[5] Bluetooth specification

https://www.bluetooth.org/spec/

[6] PNI V2Xe 2-Axis Compass Module Manual

https://www.pnicorp.com/downloadResource/c4b6/manuals/73/V2Xe+Manual.pdf

[7] Motorola MPXA6115A pressure sensor datasheet

http://www.freescale.com/files/sensors/doc/data_sheet/MPXA6115A.pdf

[8] Encarta: Earth Magnetic Pole

http://encarta.msn.com/encyclopedia_761576907/Magnetic_Pole.html

[9] Encarta Atmosphere – atmospheric pressure

http://encarta.msn.com/refedlist_210093798_7/range_of_pressures.html

[10] A. Wigley, S. Wheelwright, “Microsoft® .NET Compact Framework (Core Reference)”, Microsoft Press, 2002

[11] John Sharp, Jon Jagger, “Microsoft Visual C# .NET Step by Step”, Microsoft Press, 2002

[12] Visual Studio .NET http://www.msdn.microsoft.com

[13] Mobile developing http://www.microsoft.com/windowsmobile

[14] Seon-Woo Lee, Kenji Mase, “Recognition of Walking Behaviors for Pedestrian Navigation”, http://www.mis.atr.co.jp/~senu/pdf/cca01.pdf

[15] The FLEXnav Precision Dead-reckoning System http://www.engin.umich.edu/research/mrl/urpr/In_Press/P112.pdf

[16] Alan Ghelberg, “Designing a Dead-Reckoning Positioning System” http://www.eng.yale.edu/pjk/EESrProj_02/ghelberg_rpt.pdf