HETEROGENEOUS TOOL KIT FOR REAL-TIME EDUTAINMENT

REINHOLDS ZVIEDRIS[1,2], ARTIS MEDNIS[2], GATIS MEDNIS[3]

[1] *University of Latvia, Faculty of Computing*
[2] *Institute of Electronics and Computer Science*
[3] *Akero Systems SIA*
reinholds@zviedris.lv, artis.mednis@edi.lv, gatis.mednis@akerosystems.lv

Abstract: *In today's world, digital technology is taking over our lives more and more including education and entertainment. In both fields situations occurs when someone needs to control or coordinate tasks in different locations to be performed just in time. There exists certain amount of modus operandi characterized by various levels of complexity that tries to solve this issue. During state-of-the-art research authors found a lack of simple solutions that could be easily adapted for specific user requirements while still suitable for wide area of needs. In this paper we describe our experience taken from developing of hardware and software tool kit suitable for education or entertainment event staging and management where real time user-assigned task control or task reassignment is required. Hardware part includes embedded device that performs SRD interface control and configured information output. Development of this part was done with energy economy in mind to ensure the utmost possible autonomous operation of device. Software part is lightweight web based application that can be accessed by event organizers and participants using HTML-enabled Internet browser from PC or smartphone. Mobile phone's built-in camera and matrix bar code interface can be used as an option for access to the system or execution of specific tasks. Over the course of last 5 years tool kit has been progressively implemented in hardware and software prototypes that found multiple applications in various "Autoliste" and "CDT" organized events.*

Keywords: game based learning, mobility, distributed systems, embedded hardware and software.

1. Introduction

In today's world, digital technology is taking over our lives more and more including education and entertainment. Both fields quite frequently encounter situations when someone needs to control or coordinate tasks in different locations to be performed just in time, sometimes with limited resources in mind. There exists multiple ways with various levels of complexity how such issues can be solved.

Authors gradually performed research and development of tool kit for edutainment over the course of last 5 years to accommodate requirements of event organizers like "Autoliste" and "CDT". Research results described in this paper isn't the final version rather it's a milestone.

History of education, entertainment and sporting events that involves orienteering dates back to end of 19th century when orienteering was the part of military training and later evolved into civilian sport. Initially it's required only a map and a compass to aid crossing of unknown land on foot. Car orienteering's concept is similar to traditional orienteering and it's initially evolved from rally and used map to navigate along unfamiliar roads. One of the pioneers in the context of car orienteering in Latvia is "Autoliste" that started to organize such events in 2001. As time progressed participants became more technology hungry and events started to evolve from simple car orienteering to more complex tasks that required more resources to control them and evaluate end results. Therefore authors of this paper and persons that are closely involved in "Autoliste" operations chose to move to new more robust solution to avoid common pitfalls for entertainment and sporting events that involves results that can be evaluated, like:

- It's resource consuming to print and prepare multiple orienteering checkpoints that needs to be placed in exact locations in very short time before the event to avoid information leakage to participants – same applies for printing of various materials required for event, like control cards, maps, road books etc.;
- It's also time consuming to use photos as method of authorizing checkpoints due to huge amount of visual data that aren't structured and needs to be processed by human that results in errors that can lead to incorrect and delayed event results;
- Usage of traditional orienteering equipment for checkpoints is limited due to that they are suitable for registering fact of visit of checkpoint, but still event organizers requires additional method of task delivery to participants.

Therefore authors developed their own solution with simple yet robust web application as its user interface and controlling system and micro-controller based embedded device as checkpoint.

The paper is structured as follows. In Section 2 we describe our system architecture, its hardware and software components. Section 3 describes some use cases of our system; Section 4 briefly touches on related work and Section 5 concludes and summarizes our work and sketches further prospects.

2. System Architecture

Architecture of authors proposed solution is presented in Figure 1 that displays all components of the system and outlines specific research goals covered by this paper.

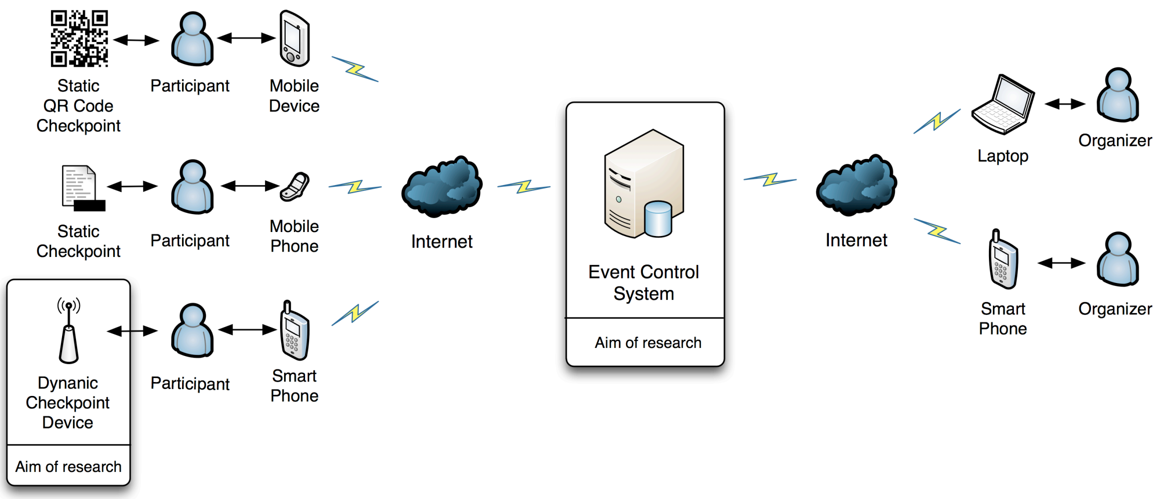

Fig. 1. **Overall system architecture with outlined research goals**

2.1. Hardware

Design and development of the hardware part was carried out with the aim to create a solution characterized by inexpensive development and maintenance costs, simple end-user interface as well as low energy consumption. Therefore, following list of technical requirements was chosen as a basis for embedded checkpoint device:

- Device should be able to use a Short Range Devices (ERC, 2011) radio transmitter operating in FM broadcasting band as end-user interface;
- Device should be able to perform its operation according pre-programmed timetable using as reference Real-Time Clock (RTC);
- Device should be able to store certain amount of data used for transmission to end-users according pre-programmed timetable;
- Device should be able to operate autonomously without battery replacement at least 350 hours (approximately two weeks).

One of the ways for inexpensive design of embedded electronic devices is usage of open source electronics prototyping platform. In our case Arduino platform and its board Arduino Pro (Arduino, 2008) was selected due its popularity and relative low hardware component costs as well as support from MansOS operating system (Strazdins, 2010; MansOS 2012) developed in cooperation between Institute of Electronics and Computer Science and University of Latvia. Checkpoint device's main board prototype is shown in Figure 2 and is based on ATmega328P (Atmel Corporation, 2011) microcontroller with external 8 MHz oscillator. To ensure connectivity to PC for firmware upload and configuration device uses Universal Serial Bus (USB) through FTDI chip based cable or board that is connected to device when required. That helps to reduce amount of components required and decrease energy consumption. To facilitate firmware upload microcontroller is programmed using appropriate boot loader software.

Short Range Devices radio operating in FM broadcasting band was selected as end-user interface due widespread usage of corresponding receiving devices. Capable receivers are installed in almost all passenger vehicles and portable receivers can be purchased for less than 5 EUR. A lot of mobile phones and smart-phones among other functionality can act as FM broadcasting band receivers, too. Data transmission through selected end-user interface was performed using Morse code. This choice was motivated by relatively simple software development and a smart additional task for end-users of embedded checkpoint device.

Short Range Devices radio transmitter was designed, using as basis device developed by Burak Incepinar (Incepinar, 2001). Modifications include programmable transmitter power supply commutation as well as modulation with programmable Morse code signal instead the audio signal from microphone. Another option was usage of ready-made transmitter module such as NS73M (Niigata Seimitsu, 2006). This option was not chosen due relative higher costs and interface complexity as well as relatively lowly requirements for transmitting frequency accuracy and transmitting power for this particular solution. During later research phase were discovered solution (Domburg, 2012) that needs even less components and is fully microcontroller based, except it's not fully stable and powerful, and has limited time of operation due to frequent modification of microcontroller's internal RC oscillator's calibration registers and their deterioration. As well authors have considered AM based transmitter (Arduino Forum, 2007), but dropped the idea because of unpopular for nowadays transmission band and modulation type used and therefore – decreasing of the availability of suitable end-user receivers.

To ensure a proper begin and end of the operation of the checkpoint device just in time as well as changes of the transmitted content according pre-programmed timetable the device contains a RTC chip PCF8593 (NXP B.V., 2010). Communication with this circuit is performed using I2C interface.

To ensure a sufficient amount of non-volatile memory to store specific content transmitted to end-users the device contains an I2C Serial EEPROM chip AT24C64C (Atmel Corporation, 2008). The size of the EEPROM used in this device – 64 KB - was selected to provide space for 2048 4 to 8 digit long checkpoint codes that can be changed from 1 to 30 times per hour providing from 68 up to 2048 hours long continuous operation time if device is equipped with large enough batteries.

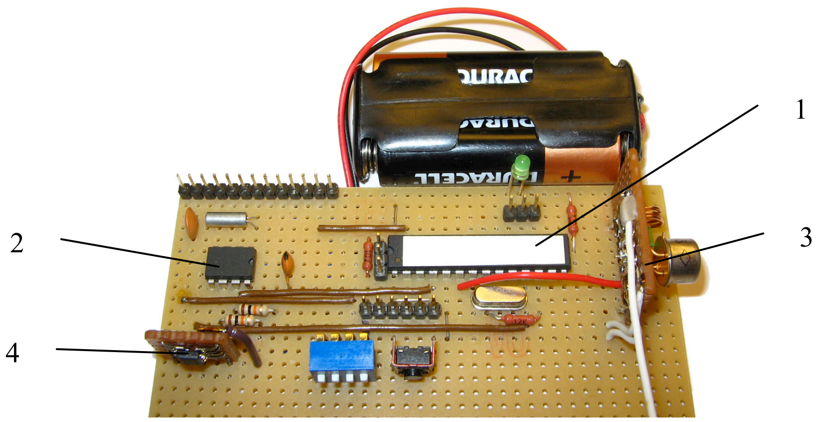

Fig. 2. **Main board of embedded checkpoint device:** 1 – microcontroller, 2 – RTC chip, 3 – FM transmitter board, 4 – EEPROM chip

Energy consumption of the developed embedded checkpoint device is determined by energy consumption of separate components as well as their duty cycles. Measurements of energy consumption during different operation modes are presented in Table 1. Analysis of these data revealed that embedded checkpoint device in standard situation can operate autonomously up to 536 hours or more than three weeks (Table 2) using two Alkaline AA batteries with typical capacity of 2000 mAh.

Table 1

Embedded checkpoint device's energy consumption during different operation modes

Mode	Consumption, mW
Sleep	0,89
Configuration (FM transmitter OFF)	11,23
Active (FM transmitter ON)	15,12

Table 2

Autonomous operation time using different checkpoint code lengths and 8 sec sleeping between two subsequent code transmissions

Checkpoint code length, digits	Code transmission time, sec	Consumption per hour, mWh	Autonomous operation time, hours (days)
4	18	10,74	536 (22)
6	27	11,87	485 (20)
8	36	12,53	460 (19)

Device firmware was designed with robustness in mind so that later for its operation is enough to have computer equipped with USB port and Serial terminal program. To configure device, first you need to switch on the configuration mode using hardware switch found on it and then through the terminal perform configuration tasks according to requirements of event where its going to be used. All configuration commands are numeric and have similar data entry format (1). Length of command and parameters in total are limited to by Atmega328P serial buffer size that is 128 bytes. After command has been processed on device user receives OK or NOK (Not OK) message preceded by command number used. NOK message usually means that command had invalid format or invalid parameters or microcontroller didn't fully read serial buffer. Configuration mode can be used to read back configured parameters stored in microcontroller's internal 1KB large EEPROM memory as well as codes stored in external EEPROM chip using the same data entry format (1) as for configuration.

$$A * B[*C]\# \tag{1}$$

where A – Command, numeric;
B – First parameter, numeric, depending on command;
C – Optionally second parameter, numeric, depending on command;
* – Divider of arguments;
\# – End of command.

2.2. Software

Design and development of the software part - Event Control System (ECS) - was carried out with the aim both to reduce administrative effort required by organizers and to provide dynamic environment for participants. Additionally the following non-functional requirements were taken into account:

- Software users might be mobile;
- Software should be available on demand;
- No specific hardware should be required.

To meet these requirements ECS was implemented as lightweight web application with both WAP 2.0 and full HTML support. That covers most mobile and smartphones currently in use, which is of particular interest for outdoor events in rural areas, where other computing and communication devices might not be readily available.
ECS addresses the following aspects of event organization:

- Task definition;
- Participant registration;
- Control of checkpoint placement and activation;
- Task assignment to participants;
- Task progress tracking;
- Task result evaluation;
- GPS track analysis;
- Traceable communication between organizers and participants.

Task definition in structured format is essential for event control automation. For this purpose data structure shown in Figure 3 was used in ECS. It consists of three parts where first one is task that is uniquely identifiable and contains some description. Task may be related to particular location defined by GPS coordinates, as well it may be available for limited time duration. There is at least one question in a task and it is also uniquely identifiable and contains some description. There is at least one answer for a question – again it is identifiable and contains description. Each answer variant has been assigned a value. Participant's result for answering particular question is determined depending on verification algorithm. Answers to some questions are required to complete task. Minimal rate of answered questions may be required to complete task. Overall task value within event may be balanced via coefficient.

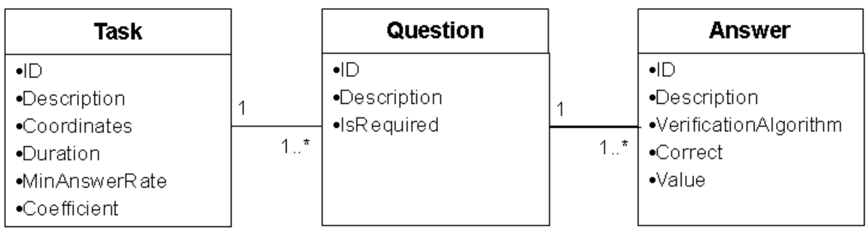

Fig. 3. **Task data structure**

Particular data structure along with answer verification algorithms of string match and numeric difference has been sufficient to define rather different types of tasks e.g.:

- Orienteering checkpoint authorization by entering code - task with GPS coordinates, 1 required question with 1 answer that is verified by exact string match;
- Timed test - task with duration and minimal rate of answered questions, i.e., 30 questions with 4 answers each that are verified by exact string match;
- Estimation of distance between two objects - task result is calculated as numeric difference between participant answer and correct answer.

Both data structure and list of available verification algorithms may be extended as new types of events and tasks are designed for ECS.
Task assignment to participants via ECS allows for fine-grained control over which tasks are available for which participants at which moment of time irrespective of organizers' current location, that could strain their resources

otherwise. It is of particular interest for events that span large territories and have many participants e.g. car orienteering. Some event design patterns that benefit from this approach are as follows:

- Making task available for participant at particular moment of time irrespective of location e.g. controlled mass start from multiple locations;
- Making task available based on participant performance so far e.g. multistage event, where next stage becomes available after predefined quota of previous stage has been completed;
- Assigning task for participant based on overall event state e.g. to facilitate even participant distribution in event area.

Task description in ECS may be any content that can be delivered via web browser - text, images, multimedia files - as shown in Figure 4. In case of mobile phone as client device available software and screen resolution had to be considered. To accommodate for potentially weak mobile network signal and therefore slow network speed, e.g., in forest or valley, typical transferred data amount for 1 request / response was reduced below 10 KB.

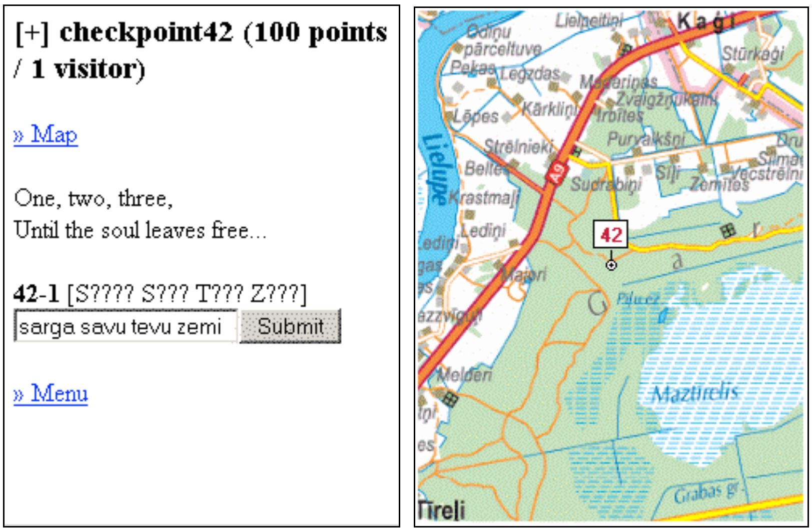

Fig. 4. **Task in ECS**

Task progress tracking and result evaluation are performed in ECS in real time - as soon participant registers an answer to a question, result for particular question is determined and completion status of corresponding task is evaluated. These data are immediately available to organizers, which allows them to monitor both task execution process and participant performance and take corrective actions if required e.g. in case of task description errors or cheating. Depending on event rules and task type some of these data are available to participants as well.

Several input methods have been evaluated in ECS to optimize answer registration on mobile phones:

- Selection from predefined answer variants. There is low amount of input errors. Screen resolution is a technical limiting factor. Question format is a task definition-limiting factor.
- Text input. There are various input errors. They can be resolved to some extent by:
  - Predefining input format;
  - Limiting input length;
  - Normalizing received input - ignoring case, diacritics, special characters;
  - Giving limited number of hints in case of wrong input.
- Quick Response code. Most mobile phones currently in use have built-in camera. In combination with corresponding software this allows with one key press to retrieve URL, which registers answer, from QR code and open it in web browser. However there are some technical limiting factors. There is multitude of QR code recognition software, browser software and mobile phone combinations available, thus organizers' are neither able to test for, nor consult participants in case of component integration problems. Additionally QR code recognition is rather dependant on lighting conditions, background colour and size e.g. this method is not well suited for a night event.

Automated result evaluation in real time allows usage of more complex algorithms, e.g., dynamic task value coefficient within event based on number of participants that completed it, and immediate result publishing irrespective of number of participants.

Result evaluation may include participant GPS track analysis. After finish of event participants themselves or organizers can upload their track in GPX format to ECS and its performs analysis of the following aspects:

- Maximum speed. There are road speed limits to be abided. If speed calculated by timestamp and coordinates from track exceeds predefined speed for predefined period, corresponding incident is registered for each occurrence and disciplinary action against offending participant may be taken. GPS tracks recorded on typical devices are approximated - there are less points on straight fragments and more in turns. Usually timestamps are given with second precision. This leads to several problems. On straight fragments slight curves are ignored, which reduces distance and slightly increases calculated speed. In turns points may be registered every second, which leads to undefined actual time span anywhere between 1 to 2 seconds and has significant impact on calculated speed. To accommodate these measurement errors and reduce false positives minimum time span to be analyzed is defined and rather conservative predefined speed and period parameters are used e.g.:

- Time span of at least 10 seconds, which should reduce timestamp error below 10%;
- Predefined speed of at least 100 km/h for road speed limit of 90 km/h;
- Predefined period of 60 seconds, which should indicate speeding, but exclude temporary acceleration e.g. during overtaking.

Besides extremely high speed indicates data integrity problem - coordinates of subsequent track points are too far away e.g. because of GPS receiver error or because of data tampering.

- Checkpoint authorization. Task may be considered completed if participant has visited area within predefined distance of checkpoint. An event design pattern that benefits from this approach are virtual checkpoints that are defined as coordinates and participants equipped with GPS receiver have to visit them. Speed within checkpoint area may be used as indicator for data tampering e.g. if checkpoint is reachable only by foot, it is reasonable to expect that participant speed in checkpoint area will not exceed 15 km/h.

Additionally GPS data may be used to provide dynamic visual representation of event. In case of ECS it means event replay with participant position display on Google Maps and intermediate results as shown in Figure 5. It is of interest for organizers to evaluate event design and it is of interest for participants to compare themselves with competitors.

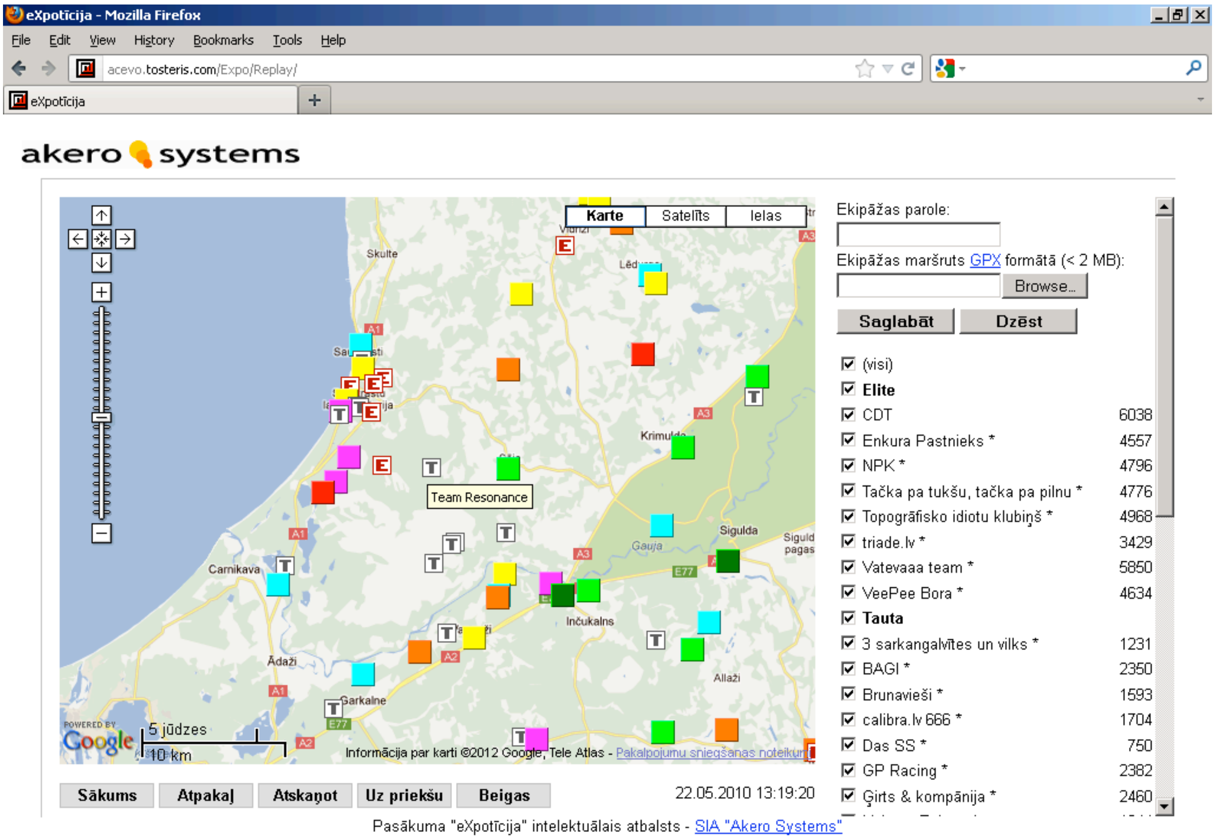

Fig. 5. **Event replay in ECS**

Communication support in ECS is implemented as messages with individual or group scope that are registered in system and delivered to addressees via e-mail to SMS text messages. In communication from organizers to participants this provides cost and time efficient means to distribute information to large audience. In communication from participants to organizers this provides means to allocate organizer resources for incident resolution on need-to basis without taking a risk of too few or too many appointed contacts.

3. Evaluation

Several prototypes of the tool kit were evaluated in various stages of research and development in multiple events organized by “Autoliste” and “CDT” over the course of last 5 years. Most of the events were designed based on search of the shortest path in predefined graph algorithm. Some parts of latest events were based on project ideas covered by Bekker et.al. (Bekker, 2010).

First deployment of ECS was in September 2007 in “Autoliste” organized event “AutoChase: Descent” and it was based only on SMS text message exchange between event participants and organizers for registration of checkpoints found and receiving next tasks to perform. Lesson learned from this deployment – you couldn’t rely on mobile operator’s SMS system with public SMS sending functionality because it has no correlation with any prioritization algorithms meaning that Message 2 can be received before Message 1.

After fiasco with SMS system authors decided to move ECS development to the system described in Section 2.2 resulting in multiple successful deployments starting from May 2008 and ending on September 2011.

In May 2011 “Autoliste” organized event for its 10-year anniversary during which some of participants were not playing fair game – before start of the event they visited some static checkpoints with the aim to get checkpoint codes therefore gaining advantage over others. That led to disqualification of respective participants and authors decided that system must be improved using embedded checkpoint device which should be small and therefore can be well hidden, turned on only after start of the event and turned off at the end – and all before-mentioned operations should be done autonomously. Also this approach would lead to more precise timing of registered events because prior to such improvement participants could gather multiple checkpoint codes and enter them into ECS all at once. That led to situations when organizers weren’t able to determine exact time when participants really visited each checkpoint. Due to usage of embedded checkpoint device’s transmitted dynamic code that is later entered into ECS it is possible to determine at least exact time frame when participant visited particular checkpoint leading to more flatten event time scale.

After 3 months of development device described in Section 2.1 was ready and tested by 20 participant teams in September 2011 during “Autoliste” organized event.

In total tool kit was evaluated in 17 events that included:

- 1 event with embedded checkpoint device (static checkpoints were used, too);
- 1 event with only QR code based checkpoints;
- 3 events where participant’s submitted GPS track were evaluated;
- 2 events where participants were required to track organizer car that was moving around event’s region, and after receiving new task from them find one or more static checkpoints;
- 10 events with only static checkpoints.

4. Related Work

While performing research and development of the tool kit authors evaluated related work in the field of outdoor and indoor game based learning and activities. Mostly all related work that covered ideas similar to ones described in this paper was intended for children learning experience while ones described in this paper are mainly evaluated in grown-ups environment. Conjunctive thread for related work found and most “Autoliste” and “CDT” organized events are that they (even before employing ECS system) are based on story that ties together various subtasks into one big task similar to ideas described by Verhaegh et.al. (Verhaegh, 2006) except – for grown-ups.

Facer et.al. (Facer, 2004) describes study that explores mobile technologies used in direct interaction with environment and other participants and it can be combined with principles of engagement and self-motivation amongst 11 and 12 years old children. Ideas covered in this work are similar to concept that uses “Autoliste” and “CDT” in their events – competition increases engagement and results in powerful and enjoyable learning and gaming experience.

In some works (Magielse, 2009; Bekker, 2010) were described specific devices created for gaming experience – due to their specialization for children all of them had an impact on audio-visual experience that correlates with Morse code employed in our system. One of recent papers that were found (de Vasconcelos Campos, 2011) described full computer system on participants to fully track, command and control them. For organizers it gives the opportunity to real-time tracking of participant and easily changing their tasks depending on situation.

As commercial analogue to embedded device with limited possibilities can be named Garmin Chirp (Garmin, 2010) – it works together only with Garmin GPS devices and can be used only to point user to exact location therefore requiring additional channel of information passing to participants.

Due to popularity of ECS system in “Autoliste” events, there already exists a clone of it – Xelluc SVS (Makatana, 2011) that currently has been used in 1 event – “Makatana”, but event organizers are planning to use it again. Main difference between systems is that in Xelluc SVS have been incorporated “The Settlers of Catan” rules and skipped a lot of features from original ECS system – except visual look.

5. Conclusions and Future Work

In this paper authors have presented their experiences from designing and developing of heterogeneous tool kit for education and entertainment that consists of software and hardware parts. They have developed and field-tested software system for event organization and management and developed prototype of a microcontroller based embedded checkpoint device with Real-Time Clock for precise timing and EEPROM memory for code storage.

The future work includes further development of software system to adopt it for wider use of event types. Also, embedded device must developed further into next stage where every participant instead of using their mobile phone will have their own device with built-in 2.4GHz transceiver module that will be able to communicate directly with checkpoint device and with other participants devices. That will enable organizers to adapt tasks more dynamically. Participant devices then can be used as data transportation devices or data mules to transport data gathered by checkpoints and from other participants to some predefined data harvesting point therefore eliminating need for intensive use of GSM network as data transport and saving on event organization costs. From research point of view participant interactions can be analyzed to get social patterns of participants. Due to recent decrease of electronics component pricing authors plan that participant devices will be equipped with GPS modules and data gathered from them can be analyzed to find participant behaviour, shortest path search in predefined environment performed by human and comparison with calculated results (Sadilek, 2010).

Acknowledgements

This work has been partly supported by the European Social Fund Project No. 2009/0219/1DP/1.1.1.2.0/-09/APIA/VIAA/020 "R&D Centre for Smart Sensors and Networked Embedded Systems", "19 points" SIA and "Akero Systems" SIA. Authors would like to thank "Autoliste" in person of Māra Niedra and "CDT" in person of Pēteris Daknis for testing their work during all development stages.

References

Arduino, 2008. Arduino Pro Schematics. Available at: http://arduino.cc/en/uploads/Main/Arduino-Pro-schematic.pdf, 15.01.2012.

Arduino Forum, 2007. Arduino + Wire = Morse Code AM Radio Beacon. Available at: http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1170901754, 15.01.2012.

Atmel Corporation, 2011. Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash ATmega328P. Available at: http://www.atmel.com/dyn/resources/prod_documents/doc8161.pdf, 15.01.2012.

Atmel Corporation, 2008. 2-Wire Serial EEPROM AT24C32C AT24C64C. Available at: http://www.atmel.com/dyn/resources/prod_documents/doc5298.pdf, 15.01.2012.

Bekker, T., Sturm, J. and Eggen B., 2010. Designing playful interactions for social interaction and physical play. In: *Personal and Ubiquitous Computing*, Springer, pp. 385-396.

de Vasconcelos Campos, D. and de Beauclair Seixas, R., 2011. Command and Control: A low cost framework to remotely monitor military training. In: *Proceedings of the 2011 Military Modelling and Simulation Symposium*, Society for Computer Simulation International, pp. 36-43.

Domburg, J., 2012. AVR-based FM-transmitter. Available at: http://spritesmods.com/?art=avrfmtx, 10.02.2012.

ERC, 2011. ERC RECOMMENDATION 70-03 (Troms 1997 and subsequent amendments), Relating to the use of short-range devices (SRD), Recommendation adopted by the Frequency Management, Regulatory Affairs and Spectrum Engineering Working Groups. Available at: http://www.erodocdb.dk/docs/doc98/official/pdf/rec7003e.pdf, 15.01.2012.

Facer, K., Joiner, R., Stanton, D., Reid, J., Hull, R. and Kirk, D., 2004. Savannah: mobile gaming and learning? In: *Journal of Computer Assisted Learning*, Wiley Online Library, pp. 399-409.

Garmin Ltd., 2010. Garmin Chirp Owner's Manual. Available at: http://static.garmincdn.com/pumac/Chirp_OM_EN.pdf, 15.01.2012.

Incepinar, B., 2001. Simple FM transmitter with a single transistor. Available at: http://tacashi.tripod.com/elctrncs/smplfmtr/smplfmtr.htm, 15.01.2012.

Magielse, R. and Markopoulos, P., 2009. HeartBeat: an outdoor pervasive game for children. In: *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, ACM, pp. 2181-2184.

Makatana, 2011. Autoorientēšanās pasākums "Makatana". Available at: http://makatana.xelluc.lv/?page_id=10, 15.01.2012.

MansOS, 2012. MansOS Operating System. Available at: http://mansos.edi.lv/, 10.02.2012.

Niigata Seimitsu Co., Ltd., 2006. NS73M-61LU-0101 Product Specification. Available at: http://www.sparkfun.com/datasheets/Wireless/General/NS73_Datasheet.pdf, 15.01.2012.

NXP, B.V., 2010. PCF8593 Low power clock and calendar Rev. 04. Available at: http://www.nxp.com/documents/data_sheet/PCF8593.pdf, 15.01.2012.

Sadilek, A. and Kautz. H., 2010. Recognizing Multi-Agent Activities from GPS Data. In: *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, AAAI.

Strazdins, G., Elsts, A. and Selavo, L., 2010. MansOS: Easy to Use, Portable and Resource Efficient Operating System for Networked Embedded Devices. In: *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, ACM, New York, NY, USA, pp. 427-428.

Verhaegh, J. and Soute, I. and Kessels, A. and Markopoulos, P., 2006. On the design of Camelot, an outdoor game for children. In: *Proceedings of the 2006 Conference on Interaction Design and Children*, ACM, pp. 9-16.