Network Enabled Capabilities – Grassroots Perspectives

Prof. Leo Motus, Dr. Merik Meriste, Jürgo-Sören Preden
Estonian Centre for Dependable Computing
6 Kohtu Street
Tallinn, 10130
Estonia

leo.motus@akadeemia.ee, merik.meriste@ut.ee, jyrgo@gt.ee

ABSTRACT

Majority of NNEC related studies focus on large scale operations, on concepts and methods of global access to operative information necessary to coordinate and control activities that involve large resources. The role of small autonomous squads, operating with limited resources, increases in the case of asymmetric warfare. This paper discusses problems related to providing an autonomously operating squad – that has only occasional access to global operative information -- with locally created information that can be synchronised with the global information system, whenever, and if, the opportunity occurs.

The situational awareness of the squad is to be synthesised by fusing information acquired from sensors and equipment installed on portable unmanned vehicles, from the dynamically deployable smart sensors, and from reconnaissance parties. The information is exchanged via mobile ad hoc network that connects the information sources and the information users within the squad; information processing (verification, fusion, and transformation to suitable form) is distributed between the nodes of the mobile ad hoc network. In order to check the semantic consistency of the information, together with its time-wise and position-wise validity, we need a novel model of computation. In this paper we discuss the on-going development of situation-aware model of interactive computation, its use in elaborating the multi-agent based interactive digital map provider, and also describe some experiments related to determining the position of smart sensors.

1.0 INTRODUCTION

Small autonomous squads have been inevitable throughout the warfare history for fulfilling special tasks, for intelligence surveillance and reconnaissance, etc. In today's asymmetric warfare, the small autonomous squads may often form the major fighting force, especially if their activities are well supported by network enabled capabilities. This has lead to "swarm" tactics, as explained in [1]: "one advantage of NNEC is that networked forces consist of smaller units that can travel faster and lighter; all units know the other's location and if one runs into trouble, other independent units nearby can quickly come to its aid, swarming to attack the enemy from all directions at once".

The "swarm" tactics becomes possible only if simultaneously with the fast distribution of information between echelons and structures, also the right for decision making, and capability to do so move down the echelon [2]. One of the major factors influencing the decision-making capability is availability of high-quality information that is dependable, coherent, and structured to match with the actual situation. The ideal availability and dependability of information and decisions can be guaranteed (theoretically and in principle) assuming the presence of ideal communication infrastructure, huge information processing

Motus, L.; Meriste, M.; Preden, J.-S. (2006) Network Enabled Capabilities – Grassroots Perspectives. In *Dynamic Communications Management* (pp. 16-1 – 16-14). Meeting Proceedings RTO-MP-IST-062, Paper 16. Neuilly-sur-Seine, France: RTO. Available from: http://www.rto.nato.int/abstracts.asp.

power for fusing and cross-checking of the acquired chunks of information, automatic matching of the ontology (or semantics) of the used information chunks, easy-to-use human interface, and reasonable time for decision-making.

In normal conditions and on the higher level of decision-making (that takes place in more or less stationary offices (e.g. headquarters), under normal traffic load of the communication infrastructure, etc.) we are not too far from, or at least moving towards satisfying those assumptions (see, for instance, [3]). However, when the crisis occurs, many of the above assumptions are not satisfied any more – for instance:

- traffic load increases approximately by an order of magnitude, respectively increases the communication delay; the actual increase of the delay depends upon the used protocols
- currently available information chunks often belong to inconsistent positions, or are inconsistent in time, i.e. the amount of coherent, dependable information reduces remarkably
- the human interface is (not so seldom) poorly designed and implemented, therefore the decision-maker has to manage with inconsistent in time and position, or contradicting information; hence the respective decisions are often incorrect; at the same time the stress level of the decision-maker rises because of the other potential consequences of the crisis.

Many examples that illustrate different aspects of the above statement can be found in the literature on embedded real-time systems and analysis of accidents – e.g. accidents in nuclear power stations, in chemical factories, avionics, and behavioural analysis of software-intensive applications (see, for instance, [4,5,6]).

Clearly, once the problem has been detected it can be (somewhat) relieved. The relief is easier to find in the stationary conditions, e.g. in the case of internet use alternative stationary communication media, use regularly updated copies of databases, use exchange operators instead of the stressed ones. Most of the listed methods are not applicable in autonomous squad that operates on its own, and with strictly limited resources. Also the user requirements, and respectively the priorities of many NEC aspects may differ pending on the level of decision-making – an autonomous squad with the limited material, communication, and time resources, operating deep in the hostile environment is not primarily interested in spreadsheets, the content of databases and web-pages.

In the most cases, the squad needs (in addition to general background information) detailed information about the operative situation in its vicinity -- preferably in the form integrated into a single source of essential information (e.g. a digital map). This information source should comprise the periodically updated reconnaissance results, collected online information from smart sensors, unmanned vehicles, headquarters, and other sources. The squad-specific information is presented by specially defined (potentially moving) icons on respective layers of the digital map. Each icon may have a shortcut to additional, more detailed information, or subjective comments of the squad members. We suggest that such a unified human-computer interface satisfies most of the needs of squad members.

The rest of this paper discusses some of the pragmatic and theoretical issues related to providing the information collecting, processing, verification, and distribution functionality within an autonomously operating squad. The discussion departs from hypothetical situation that mimics the goals, tasks, and potential information technology supplies of an actual squad. The details of the hypothetical situation are used to associate seemingly disparate topics in sections of this paper, such as:

- studies into heterogeneous mobile *ad hoc* networks, including methods for detecting the position of some of its nodes, routing of messages, and security of communication

- concepts and tools for building an unified and easy-to-use human interface for the squad members, including studies into situation-aware multi-agent systems, multi-agent based interactive digital map providing tools, and building a prototype human-interface
- search for the situation-aware multi-stream interaction machine as a theoretical model for behavioural analysis of the NNEC functionality and quality of service within a squad.

The following discussion proceeds in the context of computer-science and information technology, neglecting temporarily the military application aspects. Section 2 covers the IT infrastructure and is mostly based on widely accepted ideas and requirements. This is a section where our own contribution is rather modest. Section 3 describes the work done in Estonia on elaborating situation-aware agent-based software technology. A visible application of this technology is the tool-set (Kaardikratt) for building customised, interactive digital maps that can be distributed over the network with reasonable traffic requirements. For many applications such maps can serve as a unifying interdisciplinary user interface (military, police. border-guard, fire-brigade, emergency service, etc) that in addition to operative navigation information provides access to more specific databases. Section 4 provides a superficial survey of activities related to developing situation-aware model of interactive computation that forms the theoretical basis for behavioural analysis of real-time embedded systems, and systems of systems. This model is also crucial in verifying subtle consistency properties of the acquired information, and assessing the decisions deduced from this information. The theoretical model is essential for estimating the quality of service of the designed and built applications.

2.0 MOBILE *AD HOC* NETWORK

In this section we assume that the hypothetic squad is equipped with a number of wireless laptops, and has technical capability for intra-squad communication, communication with the other squads in this vicinity, and with the faraway headquarters. In addition, we assume the existence of communication links between:

- the squad's laptops, and
- the available control and information centre(s) for portable unmanned vehicles, and
- the gateway(s) to the dynamically deployable *ad hoc* sensor network(s).

The close networking of all those devices, plus strict verification and validation procedures of the acquired information, before its fusion and distribution to the squad members (and, if possible, also to the headquarters, and to the neighbouring squads), provides accurate situational awareness – such as, location of friendly forces, detecting threatening changes in the environment and landscape as compared to the information on the maps, tracking the movements of the hostile forces, etc.

By situational awareness we mean, in addition to detection and recording the events of interest, also precise positioning of those events in the two or three dimensional space, and in time. Unfortunately the conventional GPS is not quite reliable for positioning objects indoors or in subterranean environment, in dense urban environments, and in other areas of signal blocking. Consequently, object positioning in the field is not a trivial task. Similar difficulties are observed with the usage of UTC (Coordinated Universal Time) signals for time-labelling of the events and in calculating and assessing the validity time intervals for information chunks (see, for instance, [7]).

The information technology and communication (IT) devices of a squad form the following sub-networks:

- wireless laptops of the squad members form an almost conventional wireless local area network – the additional difficulty is that each node of this network is mobile (and the knowledge of precise current position of nodes is preferable, although not mandatory) – with a server in the communication centre

- IT devices working with each (type of) portable unmanned vehicle form a set of mobile sub-networks
- Smart sensors that are deployed in the landscape (building, subterranean area, etc) form an *ad hoc* sensor network, ready to communicate with the other sub-networks via its gateway(s); most of the nodes in a sensor network are immobile, although usually their positions are not precisely known *a priori.* For more details see, for instance, [8].

The network formed from the listed sub-networks is heterogeneous, with dynamically changing topology and mobile nodes; each sub-network is operating and moving autonomously, however their movements are loosely coordinated. Further in this paper we will call such networks mobile *ad hoc* networks. Many authors characterise mobile *ad hoc* networks by relatively low communication bandwidths, and dynamically changing connectivity (see, for instance, [9]). Agent-based systems and software engineering technologies (e.g. increased use of autonomous and proactive computing) have been used to relieve the impact of low communication bandwidth and not-quite predictable connectivity of the network to the quality of service.

In practice, the mobile *ad hoc* networks often have inbuilt empirical self-organising capability to compensate dynamic changes in the network topology, and many other similar empirical contrivances – that enable to build operational networks, but do not really support the analytical assessment of the quality of service provided, and do not apply the formal verification of the information that is stored and that moves in the network. The domain of mobile *ad hoc* networks is still in the rapidly evolving experimental stage. The theoretical foundations for computing in mobile *ad hoc* networks are still in the early development stage, are being actively studied, and the research ideas often stem from the studies in self-organising and adaptive systems and are closely related with the relatively recent studies in proactive systems (e.g. are biologically inspired).

Our team works on two research issues related to mobile *ad hoc* networks, in addition to experimental study of their capabilities:

- developing and experimenting with robust methods for node positioning (alternative to GPS); position of the information source at the time of acquiring the information is essential for ensuring situation awareness and formal verification of the consistency of acquired information,
- securing the information that moves in the network from being misused, and providing reasonable protection from the intruders.

2.1 Approximate node positioning in mobile *ad hoc* network

In this paper we discuss our approach to developing robust methods for positioning the moving or stationary objects without using GPS technology. A wider context of our work in agent-based digital map software as applied for position tracking, explaining the use of geospatial agents for collecting and processing of situation-aware information, the types of applied agents, the methods of generating those agents, and some typical tasks and appropriate pilot applications has been described in [10].

The time-labelled position information of nodes can be used by several logical layers of the mobile *ad hoc* network. For instance, by adaptive multi-hop routing algorithms, by information verification and fusion algorithms, by decision-making algorithms, and for reasoning about the semantics of information chunks. In the case of deployment of smart sensors or RFID tags, the positioning is often a one-off task since the sensors and ID tags are usually stationary. The case of the smart sensors and RFID tags mounted on a mobile object can be reduced to periodical detecting the position of that object.

In principle, we face the following alternatives for nodes positioning:

- some nodes are deployed at known (predetermined) locations
- some nodes can be positioned online using an existing infrastructure (e.g. Cricket-type beacons [11]), and/or nodes with known location
- some nodes (e.g. anchor nodes) are location-aware and update their position dynamically.

Conventionally, the positioning algorithms rely on distance estimation, or use bearing information, or use both distance and bearing. The positioning accuracy depends on the used hardware, on environmental conditions (reflections, interferences), and on the computational power available. We are interested in an approximate and unassuming positioning algorithm that needs minimum online measurements – such as algorithms based on estimated communication area of the nodes in this particular environment. Such algorithms define the location area of a node as an intersection of communication areas of the anchor nodes that are able to communicate with the given node. The estimated location area can be iteratively refined as new anchor nodes contact that node.

An experimental test-bed has been set up at Department of Computer Control, Tallinn University of Technology. An indoor positioning system consists of Cricket type beacons [11], MICA2 nodes and RFID tags deployed at known coordinates, and of a mobile node that moves around in the test-bed's territory collecting positioning data about detected immobile nodes. The mobile node is equipped with a Cricket receiver, a personal computer, a MICA2 mote with a serial adapter and a RFID tag receiver. The mobile node positions itself using the Cricket beacons.

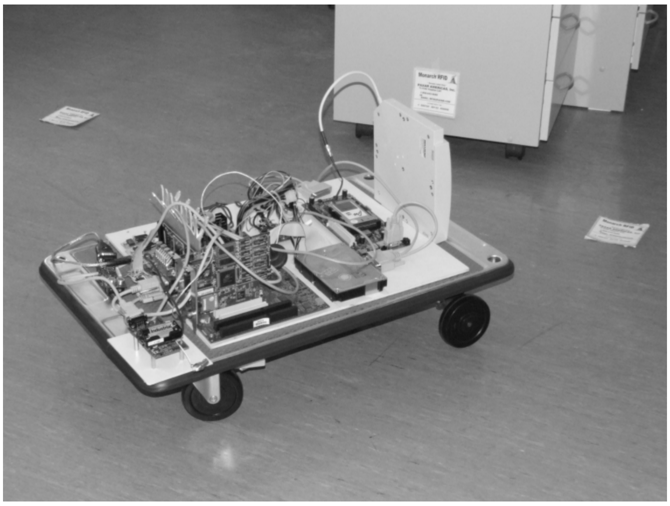

Fig.1 Mobile node in the test environment

In the experiments and simulation we assume that nodes have omni-directional antennas, hence in the homogeneous environment the communication area is circular. In principle, it is possible to extend the algorithm for use of directional aerials – as may be required in specific applications. The experimental results have been encouraging in a sense that the developed algorithm provides approximate positioning data for the network nodes with precision that is satisfactory for many applications in the field – for instance, for verifying the coherence of measurements acquired from different nodes, for detecting dynamic changes in the environment, for interpreting the pattern of changes, etc. Detailed description and analysis of the algorithm and experiments performed will be published in a separate publication.

2.2 Traffic security in mobile *ad hoc* network

The vulnerability of security in mobile *ad hoc* networks is in some aspects more serious problem than in conventional wired and wireless networks. For instance, network nodes can be easily captured, and the inbuilt or stored information (e.g. encryption keys, etc.) can be copied, or compromised. Sensor network can be disabled by just exhausting batteries in the nodes, e.g. by requesting additional computations, or excessive traffic. Cooperative nature of the nodes – e.g. multi-hop routing, that is an advantage of *ad hoc* networks – can be used by a single malicious node to degrade the throughput of the network by propagating false routing information. Besides, many nodes are too thin to handle sufficiently sophisticated conventional security measures. Therefore we cannot immediately transfer the existing security know-how to mobile *ad hoc* networks since this issue needs further theoretical and experimental research.

3.0 INTERACTIVE DIGITAL MAP

Success of applied computer systems depends heavily on the expediency of user interface. Warriors in the field are accustomed to discuss the situation, and plan actions by using maps that have been updated with the operative task related information. At the same time web pages, chat rooms, search engines, simulation tools, etc. are suitable extension to maps for high level decision-makers in the headquarters.

By crossing interactive digital map technology with agent-based software engineering we have built a prototype toolset that enables to customise interactive digital maps and to transfer those maps (or certain layers of those maps) around in a network with a reasonable additional load on the network traffic. A sample description of universal internet-based interactive digital map technologies is given in [12]. In order to improve performance of the toolset in a semi-autonomous mobile *ad hoc* network, the toolset needs preliminary tuning to:

- particular geographic area (selecting appropriate maps, pictures, blueprints for required constructions, historic notes, etc)
- prepare characteristic features of the application (icons and associated information that can be added into the map by the customers, separate icons for denoting moving objects, selecting separately transferable layers of the map, etc.)

Our group has developed, at Institute of Technology, University of Tartu, a prototype of an interactive digital map provider tool (*Kaardikratt*) that is suitable for operation in mobile *ad hoc* network. The map provider *Kaardikratt* is an offspring of a tool KRATT for building and studying generic situation-aware multi-agent systems. Various aspects of those tools have been discussed in [10,13,14]. The first experiments confirm that dynamic agent-based compilation of digital map frames can be done with reasonable processing speed, faster than by using web services of conventional geographic information systems.

3.1 Multi-agent map software

Agent-based digital map software can display areas from a variety of digital maps (raster or vector maps, maps with different colour schemes, with different packing methods, etc.). Reasonably short response time of the agent-based map software is achieved due to carefully designed network usage, combined with caching and parallel processing of source maps and databases in the source information servers.

Fig. 2 illustrates the use of agents for adding application specific layers, and/or specific icons with additional information, to the selected fragment of a base map. The autonomous and proactive agents are invoked by the user of the map, or by map server, the agents can reside on the nodes of the mobile *ad hoc* network, or if necessary (and feasible) on the external servers and (in principle) can have access to any server in the world. The main operational information and response to the queries from the node computer and the map server comes from the dedicated multi-agent system. All the information acquired by the agents is processed by the map server to update the local databases required to satisfy the requests from the node computers, or from the map server itself.

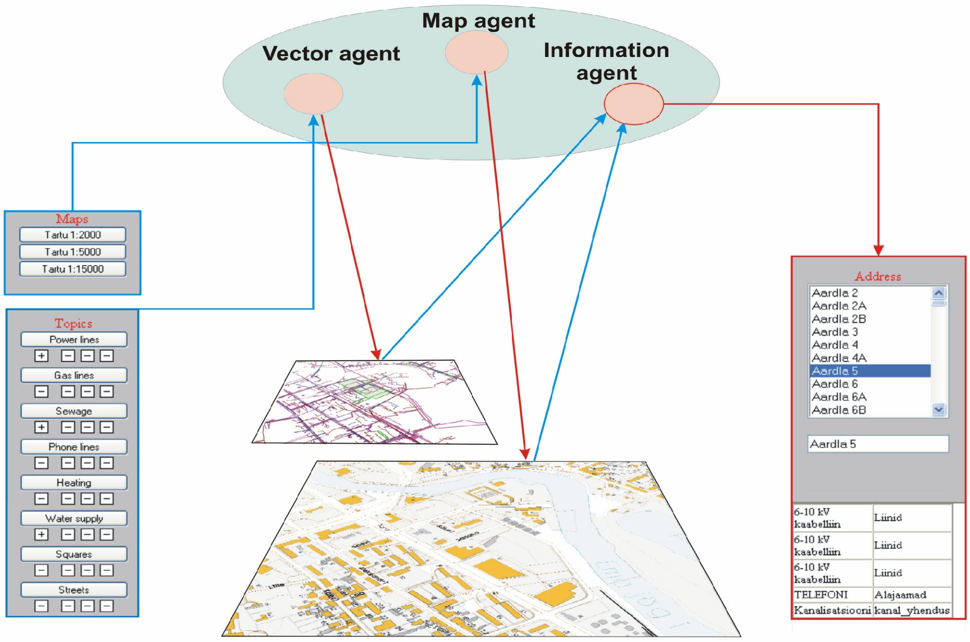

Fig.2 An application specific map is compiled by autonomous agents using information from several databases

Agent-based digital map software comprises (at least) the following types of autonomous agents:

- Internal information agents execute queries and manage databases that interact with external information agents
- External information agents manage, search, and forward the information– such as, recorded historic, environmental, or other data -- related to active map objects
- Map agents search, pre-process, and deliver fragments of the base map to the map server
- Spatial agents manage, search, process, and transmit the vector information for images (or situations) activated in the map object
- Location agents calculate and transmit periodically the position of active object to the other agents.

The set of agent types is extendable and depends on tasks requested by the user of the node computer, on types of base-maps used, on requirements of external databases used, and on other factors. Many of the agent types have generic features that facilitate the reuse of respective agents in many applications with minimal modifications.

A schematic implementation of the digital map software is illustrated by Fig.3. The laptops of the squad must have web browsers and sufficient memory space for automatically downloaded active map object and for some other background information. The map specific part of web browser's page is presented as

ActiveX component. The other parts of the page depend on the application requirements, are usually based on JavaScript, and cater for dynamic support of the page, and interactions between map object, the customer, the map server, and agents. Physically the map server can reside on any computer with sufficient computing and communication power, realistically it resides in one of the control and information centres of the squad. The prototype software has been implemented in C# and .NET platform.

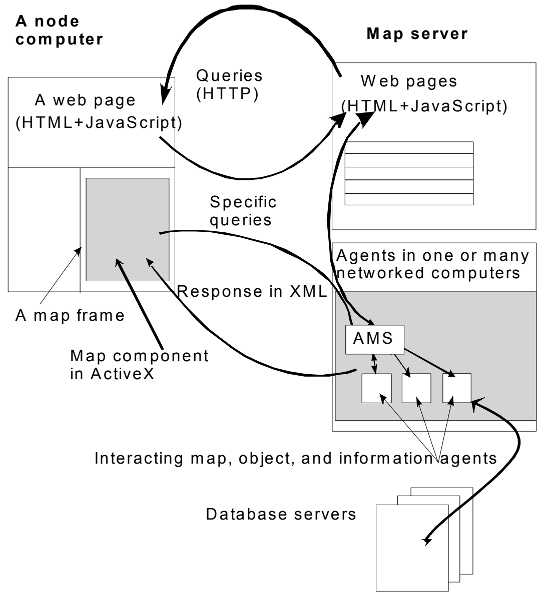

Fig.3 Implementation of multi-agent digital map software

3.2 The usage of the multi-agent map software

The network enabled capability of an autonomously operating squad may be harmed by occasional absence of communication with the headquarters. In such a case it is important to rely temporarily on the locally created network enabled capabilities that are built around the interactive digital map system, fine tuned to the squad needs. The digital map accumulates – on special layers -- the newly acquired static and dynamic information from internal and external sources, together with appropriate subjective comments added by squad members, or by the headquarters. The regularly updated digital map, or a collection of its layers, is electronically distributed among members of the squad and sent, if the connection is available, to the nearest headquarters. If necessary, the digital base-map can be substituted by a hand-drawn scheme or by a series of aerial photos (bound up with the landscape).

The actual usage of the multi-agent map software in the field proceeds in two modes:

- Positioning of the nodes should be repeated periodically during the exploitation of the network; however, after the first deployment of sensor network special efforts should be applied to determine positions of (at least some) sensors; for indoors, and/or subterranean, deployment of the sensors this mode also involves drawing or updating the area layout
- Updating the operative information (e.g. location of mobile objects, detected changes in the landscape) on the map takes place periodically, based on the information incoming from the local sources. In principle, each member of the squad may suggest updates (based, for instance, on

reconnaissance results), the information from the sensor network and from autonomous vehicles is monitored regularly and may be inserted onto the interactive map. New information from the headquarters is treated in the same way as the locally acquired information. The modifications of the map are to be verified and approved before redistribution of the map, at least the final approval procedures are external to the multi-agent digital map software.

4.0 SITUATION-AWARE MODEL OF INTERACTIVE COMPUTATION

Most of the advanced information systems that the today's technology enables to build are pretty useless, or even harmful, unless we can guarantee that the offered decisions, and information used for reasoning about those decisions are consistent and correspond to the actual situation at hand. Often we have only bona fide trust in high technology based decision support systems, since the existing theories cannot provide strict evidence in favour of the confidence level for fused information and for the deduced recommendations. In such cases the final responsibility is with the human decision-maker.

The mobile *ad hoc* networks, suggested for fostering network enabled capabilities on the grassroots level, form a system of systems, where many of the components are embedded real-time systems. Today it is a common knowledge that the conventional algorithm theory – based on Church-Turing thesis – is not capable to model the behaviour of embedded real-time systems with sufficient precision. Roughly speaking, the conventional algorithm theory can handle string processing, whereas embedded real-time systems process simultaneous interacting, situation-aware streams. Stream processing, in the general case, leads to interactive computation (see, for instance, [6, 15, 16]). The corresponding theory is still evolving and its methods are far from being matured and generally accepted, especially in the case of situation-aware interactive computation (e.g. embedded real-time systems).

The dominating practice of embedded systems verification is that the formal methods, developed for algorithm-based computation, are applied for verifying computation based on multiple simultaneous interacting streams. Such practice stretches the limits of the domain of definition of those methods and usually never checks the confidence level of the achieved approximation. Quite often the major assumptions, on which the verification methods are based, are unintentionally violated – e.g. in stream-based computation algorithms do have a memory of their previous executions, during simultaneous execution of interacting streams (for instance, forced, a.k.a. true concurrency [15,16]) "the principle on non-interference", that is the verification foundation of parallel programs, is violated, etc. Such practice is attested by a recent survey of validation and verification methods for embedded systems [17].

Methods for verifying subtle timing properties (e.g. validity intervals of events and variable values, non-transport delays of messages, etc.) assume the presence of timing information, and underlying sophisticated time model [16]. In many applications of a sensor network the sensor reading has no timing, nor location labels attached – this practically eliminates the possibility to verify consistency of this reading with the readings from the other sensors in the vicinity.

4.1 Towards situation-aware multi-stream interaction machine

The existence of an appropriate theoretical model of computation is not any more just a precondition of formal correctness analysis of the implemented software – i.e. it is not something used only for aftermath analysis of the software development process. Proportionally with maturing the model driven development methodology of systems and software, the model of computation has stronger impact on the non-functional requirements to the designed system, on data structures that are to be used in the designed system (e.g. sensor reading together with time instant of the reading, and with location coordinates of the sensor), and on the development life-cycle.

We suggest that in applications where embedded real-time systems are involved (e.g. the nucleus of grassroots level network enabled capability) the appropriate theoretical model is situation-aware multi-stream interaction machine [6] that is still under development. However, its prototype – the Q-model -- has been available since mid 1980-es [16]. The Q-model stems from the ideas of interaction-based computing and includes explicit time-awareness. Several experiments have been carried out on the Q-model, such as:

- linking the Q-model with the UML-based tools [18],
- transforming the Q-model into a weak second-order predicate calculus, in order to prove the non-contradiction of its axiom system [19],
- using the Q-model inference engine as the basis for real-time nucleus of a case-based remote diagnostic tool in a EU project BRIDGE [20].

The Q-model defines a system as a collection of loosely coupled components that interact via one-to-one connected and one-way channels. Typical interaction of components is time-selective – meaning that the consumer defines the age limits of the message it agrees to receive from the producer. Any component is defined as a mapping that may be repeated many times (up to the countable number of times):

$$p: \mathcal{T}(p) \text{ x } dom\ p \rightarrow val\ p, \text{ where}$$

$\mathcal{T}(p)$ is a well-ordered time-set that determines the time instants when the mapping is executed. The mapping can be defined across the borderline of different environments, covering the case when a part of a computing system interacts with a non-computer component in the artificial or natural environment. Such a mapping can function autonomously since each mapping has its own time-set. Such a mapping can also implement proactive behaviour by using the inner memory of the mapping about its previous executions.

The time-set may describe periodic, quasi-periodic, and spontaneous executions of the mapping – depending on the definition of the time-set elements. The time-set may also be used to describe dynamic reconfiguration of the computer network, and temporary or permanent disappearance of some components.

Such a mapping specifies a stream that permits additional flexibility, as compared to conventional stream processing. For instance, each stream in a system may have a different execution pattern (as defined by the time-set). All the streams need not be executed at regular intervals – that is essential for a system comprising several computing systems and non-computer components. The interaction of two streams – information provider (p_i) and information consumer (p_j) -- is illustrated in Figure 4.

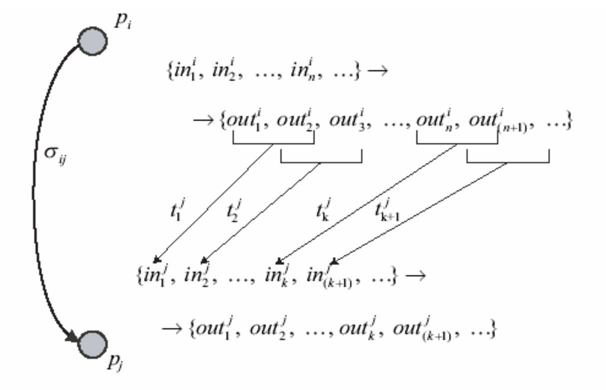

Fig. 4 Time selective interaction of nodes

Interaction of two streams generates a new stream -- stream of interactions, its formation is controlled by the information consumer (p_j). The mechanism that forms the elements of the stream of interactions is another mapping, called channel in the Q-model terminology:

$$\sigma_{ij} : val\, p_i \times \mathcal{T}(p_i) \times \mathcal{T}(p_j) \rightarrow proj_{val\, pi}\; dom\; p_j,$$

where the length of the message that passes the channel (i.e. the depth of the consumer required memory, or the length of an interaction stream element) is determined by a channel function:

$$K(\sigma_{ij}, t) \subset \mathcal{T}(p_i), \text{ and } t \in \mathcal{T}(p_j).$$

Please note that different time-sets may represent different time-counting systems (i.e. different metric times), and the channel function actually defines a timer on the producer's time-set $\mathcal{T}(p_i)$. The origin of this timer moves with the progress of the consumer -- i.e. is defined by the activation instant of the consumer $t \in \mathcal{T}(p_j)$ that generates the request for information and measures the age of the producer's information with respect to the instant t.

Depending on the relationship between the producer and consumer time-sets, one may need three basic types of channels:

- synchronous channels when $\mathcal{T}(p_i)$ and $\mathcal{T}(p_j)$ are identical
- semi-synchronous channels, when $\mathcal{T}(p_i)$ generates $\mathcal{T}(p_j)$, and
- asynchronous channels, when the time-sets are independent of each other.

The elements of the interaction stream are time-labelled by a channel type dependent relative time, so that the time constraints imposed upon the interaction of streams can be verifiably satisfied – even considering the potential uncertainty introduced when matching different metric times used by the producer and consumer streams.

We define situation-aware computing as a dependence of the system's behaviour on particular situation. Commonly, in embedded real-time systems we consider time constraints as the essential feature defining a situation. With the appearance of mobile systems, time constraints and location define the situation. In more sophisticated applications the characteristics of a situation can be extended by new features – e.g. stress level of the end users. Of those three features, location and stress level of the end user can easily be integrated into state variables that are used in models of computation, whereas time presents a harder problem (see for instance, [21]). This explains the explicit role of time in our model of computation.

Attempt to develop a situation-aware model of interactive computing belongs to the research domain of behavioural modelling of systems with emergent behaviour. The emergent behaviour is something that cannot be captured by the conventional algorithm theory. Consequently, this research remains today outside of the computer science mainstream, and has closer ties with the complexity theory – please make a distinction between complexity theory and computational complexity theory.

5.0 CONCLUSIONS

This paper is focused on issues related to supporting development of network enabled capabilities for autonomously operating squad that has only occasional communication with its far-away headquarters, but has its own ancillaries for acquiring and processing local information. Although there are still many unsolved practical and theoretical problems hindering from building truly dependable systems, the today's technology enables to build experimental systems in a sense that the quality of service of those systems is rather non-deterministic and the user of those systems must satisfy rather high educational expectations. The experimental systems are important for improving the underlying theory and the end users' human-machine interface.

The primary theoretical problems in this area are related to the mobile *ad hoc* network that comprises collection, pre-processing, fusion, and distribution of information from smart sensor networks, from portable autonomous vehicles, from the headquarters, and from reconnaissance. More specifically, the theoretical difficulty is related to situation-awareness of the collected information and to (automatic) formal verification of the fused situation-aware information.

This paper advocates the strategy of spiral progress of synthesis suggested in [22] -- in order to solve the set of primary problems we firstly have to solve some of the secondary problems related to the human-machine interface, attempting to minimise the possibility to ambiguously interpret the presented information, and to present the inconsistent information. Before we can proceed with resolving the problems with human-machine interface, we have to solve some problems with the situation-aware model of interactive computation. Having gained better understanding of the nature of computations in a mobile *ad hoc* network, we can start the following spiral by resolving some of the primary theoretical problems, etc. This paper reports the results obtained in one spiral of the ongoing work of developing network enabling capabilities for autonomous squad.

Acknowledgements. This research stems from projects partially financed by Estonian Science Foundation grant no. 4860 and by grants 0142509s03 and 0182565s03 from the Estonian Ministry of Education and Research.

6.0 REFERENCES

[1] K.W. Jonas (2005) "Network-centric operations: European capabilities", Report of the Defence Committee, the Inter-parliamentary European Security and Defence Assembly, Document A/1899.

[2] A.F. Cebrowski (2005) Introduction to a report "The implementation of Network-Centric Warfare", Office of Force Transformation, US DOD, available at http://www.access.gpo.gov/su_docs/sale.html

[3] L. Bubbers (2004) "Case Study on the NATO trial Immediate Reaction Task Force (Land)", 43 pp., Reinforce Business Innovation

[4] Fact sheet on the Accident at Three Mile Island, http://www.nrc.gov/reading-rm/doc-collections/fact-sheets/3mile-isle.html, last visited 17.07.2006

[5] N. Leveson (2004) "A Systems-Theoretic Approach to Safety in Software-Intensive Systems", IEEE Transactions on Dependable and Secure Computing, vol.1, issue 1, 66-86.

[6] L. Motus, M. Meriste, W. Dosch (2005) "Time-awareness and Proactivity in Models of Interactive Computation", Electronic Notes in Theoretical Computer Science, **141**, 69-95, www.elsevier.com/locate/entcs

[7] Coordinated Universal Time, http://en.wikipedia.org/wiki/Coordinated_Universal_Time, last visited 22.07.2006

[8] J. Hill, M. Horton, R. Kling, L. Krishnamurthy (2004) "The platforms Enabling Wireless Sensor Networks", Communications of the ACM, vol. 47, no.6, 41 – 46.

[9] S.M.T. Yau, H.V. Leong, A. Si (2003) "Distributed agent environment: application and performance", Information Sciences, **154**, 5-21, www.elsevier.com/locate/ins

[10] M. Meriste, L. Motus, T. Kelder, J. Helekivi, A. Marandi, J. Preden (2005) „Location awareness of information agents", Advances in Databases and Information Systems, Springer Verlag, LNCS, vol.3631, 199-208

[11] Nissanka B. P., A. Chakraborty, H. Balakrishnan (2000) "The Cricket location-support system", Proceedings of the 6th annual international conference on Mobile computing and networking, pp 32 - 43, August 2000

[12] Interactive digital map technologies applications in the internet, an example. http://ltgis.launchpad.onenw.org/best-practices/advanced/interactive-web-mapping, last visited 25.07.2006

[13] L. Motus, M. Meriste, T. Kelder, J. Helekivi, V. Kimlaychuk (2003) "A test-bed for time-sensitive agents – some involved problems", 9th IEEE International Conference on Emerging Technologies and Factory Automation, Portugal, vol. II, 645-651

[14] L. Motus, M. Meriste T. Kelder, J. Helekivi (2004) „Agent-based templates for Implementing Proactive Real-time Systems", Proc. International Conference on Computing, Communications, and Control Technologies, ISBN 980-6560-17-5, August 2004, Texas, USA, vol.1, 199-204

[15] P. Wegner (1998) „Interactive foundations of computing", Theoretical Computer Science, vol. 192, 315-351

[16] L. Motus and M. G. Rodd (1994) "Timing Analysis of Real-Time Software", Elsevier, 212 pp.

[17] RTO Technical report TR-IST-027 (2005) "Validation, Verification and Certification of Embedded Systems", Final report of NATO RTO Task Group IST-027/RTG-009, http://www.rta.nato.int, last visited 12.07.2006

[18] L. Motus, T. Naks (1998) "Formal timing analysis of OMT designs using LIMITS", Computer Systems Science and Engineering, vol.13, no.3, 161-170

[19] P. Lorents, L. Motus, J. Tekko (1986) "A language and a calculus for distributed computer control systems description and analysis", Proc. on Software for Computer Control, Pergamon/Elsevier, 159-166.

[20] T. Naks, L. Motus (2001) "Handling Timing in a Time-critical Reasoning System – a case study" Annual Reviews in Control, vol.25, 157-168.

[21] L. Motus (2003) "Modeling metric time", in B. Selic, L. Lavagno, G. Martin (Eds), UML for real: Design of Embedded Real-time Systems, Kluwer Academic Publ., Norwell, 205-220

[22] H.A. Simon (1996) "The Science of the Artificial", The MIT Press, Cambridge, MA, 3rd edition.