E-TENDER - AN APPROACH FOR ENSURING TRANSPARENCY IN DEFENCE BUDGET MANAGEMENT

Georgi PAVLOV and Veselina ALEKSANDROVA

Introduction

In accordance with the Law on Public Tenders (LPT) information on all current tender announcements and procedures is maintained in a database. A software product for electronic tendering called *e-Tender*, was developed to facilitate the implementation of LPT requirements. *E-Tender* may be used to enhance the transparency of the defence planning and budgeting processes seen here as a major prerequisite for transparency of a country's defence policy.[1]

The supply and demand of goods and services are the foundation of the market economy. The market is the place where the goods and services are exchanged and their prices are set. Prices depend on the degrees of supply and demand and, thus, free market is a good regulative mechanism for large-scale realisation of goods and services. Of course, for that mechanism to work effectively, it is necessary that supply is bigger than demand. This requirement, however, does not apply to some goods and services. In some cases supply is very small to satisfy the demand and does not provide for formation of market prices. For those goods and services the mechanism of an *auction* is worked out. Auction (or *tender*) is used to determine the best price that the buyers are willing to pay, as at the same time an equal chance is given to all participants. Tenders are implemented if the participants give more than is due.

Two types of tenders could be distinguished depending on the bidding procedure – overt and covert. Traditionally, overt tender is implemented as all participants meet at assigned time, in a hall, for bidding about chosen item by raising hand and saying price proposals. Everyone watches the bidding and can participate in it by offering higher price than all other participants. A tender ends when there is no more offers for bidding. That method has been proved during the years as sufficiently honourable and effective; new times, however, place new challenges. In the era of global economic developments, the trade does not recognise any geographic boundaries. People from different regions compete in a unified market, a market that has a global meaning. The conduct of traditional tenders involves travelling of a great number of participants from different geographic regions. Because of that, new forms for conducting public tenders are sought – forms suitable to overcome geographic boundaries.

Due to advances in information technology, many human activities have acquired new dimensions. Today, it is almost unimaginable to live without computers at all levels, as well as with no Internet in the sphere of global communications. There are some solutions for carrying out public tenders, as participants in several geographic points are connected through teleconference. Other solutions are based on building up a tender moment. Such solutions have many advantages, as well as some disadvantages. One drawback is the impossibility to assess the chances for success in bidding, because the number of participants is not known. That sort of tenders lasts a few days in order to allow maximum number of participants to take part in the bid. Another disadvantage is the ambiguity of the bidding process when a participant is not connected to the system virtual space by using Internet technologies, allowing users to log on for bidding in suitable.

This article offers a solution for realising overt tenders by using Internet technologies for instant messaging. Main privilege of the solution is the relative simplicity, which means that anybody can use the tool on his or her personal computer. Furthermore, a list of all participants in the bid is maintained, thus allowing everyone to estimate their chances for success in the tender. If the bidding has already started, new participants are not allowed to take part. Last but not the least, at any time every participant receives any price announced by other participants.

E-Tender allows effective tender implementation in a few hours. The solution is directed to organisations that have wide geographic audience with similar interests.

Analysis of contemporary information technologies and their application

XML technology for representing documents

The rapid advance of Internet technologies, connecting large variety of computers in a global network, brought to fruition new methods for representing and processing information, which should be universal enough for using on different type of computer systems. The World Wide Web Consortium (W3C), an organization determining directions of WWW development, makes an important contribution. For example, a standard Hyper Text Markup Language (HTML) for representation of correlated information got a great popularity. It was developed in the beginning of the 1990’s and continues to evolve. Despite the considerable achievements, HTML has its own restrictions. Because of this, its main application is to represent logically related text documents. Predicting the necessity of ways for universal representation of other types of documents, W3C initiated the development of technology similar to HTML, but much more universal. The first version of the *Extensible Markup Language (XML)* was published in February 1998.[2] Version 4.01 was launched in 1999.

Main directions of the elaboration are following:

1. Easy usage of XML in the Internet environment. Any user should be able to view a XML document as easy as HTML one. Practically, that could be achieved when XML documents’ browsers become as accessible and widespread as HTML file browsers are at the moment.
2. XML should maintain a large range of applications. The initial focus will be on exchange of structured documents in Internet environment, and it should not restrict the application.
3. XML should be compatible with the *Standard Generalized Markup Language (SGML)*, a standard introduced in late 1960’s. Many organizations have large archives in that format. For this reason XML shall be created pragmatically to provide for interoperability with existing standards while, concurrently solving the challenge of sharing structured information via Internet.
4. It should be easy to develop programmes that process XML documents. For example, a computer science graduate should be able to compose a program for processing XML documents approximately in two weeks.
5. Also, the number of optional XML elements should be restricted to the maximum extent possible. Such elements surely cause problems of the compatibility when users start to exchange documents and sometimes even cause serious obstructions and misunderstanding.
6. XML documents should be understandable and precise enough. In this way, if a user does not have an appropriate XML browser and receives XML document, he should be able to view it, in a chosen text editor, in order to obtain an idea about its content.
7. XML specification design should be created as soon as possible. The standardization process is well known with its continuity.
8. It should be easy to create XML documents. Indisputably, improved tools for creation of XML content will appear, but no immediate availability is to be

expected. Therefore, in the initial period, it should be possible to create XML documents by already known tools such as text editors, simple commands of PERL interpreter, etc.

9. XML record conciseness does not have a particular value. There are SGML attributes, which allow writing of SGML documents to be minimised. XML technology will not maintain this feature. The maintenance of these attributes would make difficult the realisation of XML programme-analyser (or person who creates it). Many recent text editors give an opportunity for defining short key combinations about defined phrases, in order to be used in the writing.

So, everybody could compose his or her own XML document representing random information using uniquely requested elements. The question is how to explain to users what each element we have used means, in order to give them the possibility to read or write document with the same structure. Another approach is to create the application that deals with documents, following a defined structure. But how that application could recognize necessary elements if each author uses his own names? For that purpose SGML is designed as a language for declaring element types in the document - Document Type Declaration (DTD). The types declared by this language (DTD) are elements, element attributes and objects.

After creating the declaration of XML document, we can disseminate it, mentioning that this XML document has a structure described in already created declaration. So, all recipients of the document will know about the structure used and, in their turn, will be able to create their own documents. In that sense, the DTD declaration facilitates the practical application of XML in data representation. Moreover, the DTD has an additional advantage - using it, each author or recipient of document could check its validity. There are XML analysers for verifying the document validity based on the DTD description. Implementing XML analysers the user could be sure that the documents, which he or she communicates, adhere to the previously defined structure.

Peer-to-peer communication

Main function of the communication network is to deliver information from a transmitter to a receiver. To that purpose, all network devices have unique identifiers and the necessary communication protocols. The communication will be fully effective when the sender knows what information any recipient is interested in, and also when the recipient tells it to the sender. In such model of communication the source of information generally is called SERVER and the receiver - CLIENT. Therefore, that sort of communication is called *client-server*. Characteristic of the model is that the server has a certain number of resources, which are ready to be given to its clients. Clients in turn initiate the communication by sending requests to the server for specific information. The server waits for the client's requests and sends the requested information, if it is available, or sends a message describing where the requested information eventually could be found. In this mode, the client who sends a request initiates the communication. It is received by the server, which composes a response and sends it back to the client. In this way the communication cycle ends. The process is depicted on Figure 1.)

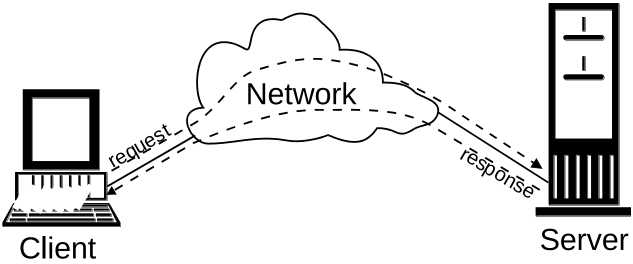

Figure 1: Client-server information exchange.

Serving as a transmitter, the station simultaneously sends and receives information. Or there could be a transmitter that may send information that has not been explicitly requested. In this case the communication is equal in value, or the so-called *peer-to-peer communication*. In this mode any transmitter sends information to defined receivers, without the need to have a defined request. It is sufficient that receivers should notify the transmitter that they want to receive defined information when it is accessible. At the same time, any receiver could be a transmitter to other receivers. In this mode of communication any receiver establishes initial connection with some transmitters. After connecting, the two sides could exchange information independently of each other, not needing an additional agreement. Simultaneously, both stations play roles of transmitter and receiver. Both transmitter and receiver could terminate the connection (Figure 2). Such scheme of communication is used in systems with distributed structure. Examples for such systems are *distributed.net* and some applications for file exchange such as *Napster* and *Gnutella*.

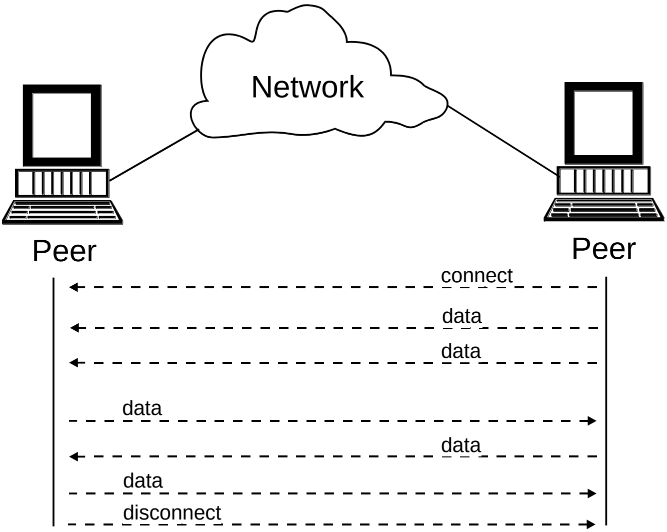

Figure 2: Peer-to-peer communication.

RSA encoding and e-signature

The encoding is an important way for keeping confidentiality of the information in its transportation by using unprotected channels for data transferring. Many approaches for encoding exist, for instance the method of RSA Security, Inc. [3] has achieved considerable popularity. The main advantage of this method is the use of pair of keys (public and private) for encoding and decoding.[4] There is a difference between this method, where the private key could be used for decoding and other methods using only one key for both operations.[5] Using public key only specific information could be encoded. In this case, any recipient could disseminate his own public key, but he must keep reliably the private one. Nevertheless, this is easier than delivering a key from the sender to the receiver without compromising it.

The public key is expressed by the two numbers (e, n), the private one – by (d, n), where e, d and n are positive whole numbers. The message, which will be encoded, is expressed as a sequence of whole positive numbers less than n and marked with m_i. The encoding the message results in a sequence of numbers c_i less than n:

$$c_i = m_i^e \ \mathbf{mod}\ n$$

that represents the encoded massage. Obviously, in the encoding process only the public key (numbers *e* and *n*) is used. After receiving the encoded message we it is decoded as follows:

$$m_i = c_i^d \mathbf{\ mod\ } n$$

Decoding can be performed if the numbers *d и n*, which represent the private key of the receiver, are known. Thus, only the owner of the private key can decode. The algorithm for composing private and public keys is of considerable importance for the security of information exchange. In order to calculate *n,* two sufficiently large prime numbers, *p* and *q,* are needed. After such numbers are found, *n* is calculated as follows:

$$n = p \times q$$

The components of the private key *d* are defined through the following procedure. A whole number, which is mutually prime with the multiplication *(p-1)* × *(q-1),* is chosen so that

$$\textit{The greatest common divisor (GCD) of } [d, (p\text{-}1) \times (q\text{-}1)] = 1$$

The public key component *e* is chosen using a whole number that is multipliable conversely of *d* by module *(p-1)* × *(q-1).*

$$(e \times d) \mathbf{\ mod\ } ((p\text{-}1) \times (q\text{-}1)) = 1$$

Despite that *n* is publicly known, it is not easy to find out *p и q* and, therefore, the private key component *d.* It is difficult to find *d* because *n* is a very high number and can not be expanded into prime factors. This fact, along with the availability of two independent keys, provides for good protection of this algorithm against compromising the encoded information.

E-signature becomes more widespread in electronic communication means by analogy with a personal signature. Any formal document, which is sent electronically, needs an e-signature to prove its authenticity. E-signature requires only one person to put it – the author of the signature, and all concerned persons should be able to perform identification and verification.

Two ways for electronic signing are commonly accepted. In the first approach, the author “*signs*” a notice, encoding it by his private key implementing the scheme described above. Afterwards, the encoded notice is sent to respective recipients, who in their turn decode it by using the sender's public key. If decoding is successful, the recipient can be sure that the notice really is from the already mentioned sender. However, if the recipient does not possess the sender's public key, he will not be able to read it.

In the second approach the notice, without any changes, is delivered to all recipients, and the signature is added at its end. Generally, a control sum for the original notice is calculated applying an algorithm set in advance. A feature of these algorithms is that the control sum changes when any change in the notice occurs. Thus, it is guaranteed that the control sum will be different if there is an attempt to modify the notice. As a result, according to the described algorithm, the control sum is encoded applying the author's private key and after that it is added to the original notice. Alternatively, it is possible to send the encoded control sum independently of the notice. This is the so-called "*detached signature*."

The control of such signature is performed in a similar manner. The control sum of the document is decoded by the author's public key, and it is calculated again by using the same algorithm. Finally, both control sums are compared, and if they coincide we could be sure that the document is in the form signed by the author. Advantage of this approach is that the information in the notice is accessible to the readers, even if they are not able to verify the sender's signature. It is also interesting to note, that e-signature does not provide protection of transmitted information, due to the use of author's public key for decoding, which is widely accessible by definition. Therefore, in the most cases the notice and signature of the definite recipient could be encoded by his own public key.

Jabber technology for instant messaging

The main application of Jabber technology is an information exchange about presence and instant messaging. The Jabber instant messaging (IM) system is distinguished from other similar systems by the following key features:

- Jabber is designed on the basis of XML;
- Jabber uses distributed network of servers;
- The Jabber protocol specification and code are opened;
- The Jabber architecture is modular and extensible.

Jabber has been designed as a model of the widely popular messaging system on the Internet, namely *e-mail*. Similar to e-mail systems, Jabber is built up on distributed network of servers that use common communication protocol, to which specialized clients connect to receive messages as well as to send messages. Unlike traditional e-mail, Jabber delivers messages in real time because the Jabber server knows when a particular user is online. This knowledge of availability is called "*presence*" and it has key meaning for the development of instant messaging.

In contrast to other instant messaging solutions, Jabber communication always passes through the server. All messages and data from one client to another must go through the server. Despite the important role of the server, the communication model remains peer-to-peer. Any client is free to communicate directly to another client, but that is not recommended because it depends on (and introduces) specifics of the realization.

In the Jabber architecture model,[6] every user has a local server, which receives information directed at him or her, and different servers transfer messages and presence information among themselves. So the client's application is simplified and the server takes care of transferring and processing information. Each server functions independently of the others and maintains its own user list and the necessary information. Also any Jabber server can talk to any other Jabber server that is accessible via the Internet. These features are shown graphically on Figure 3.

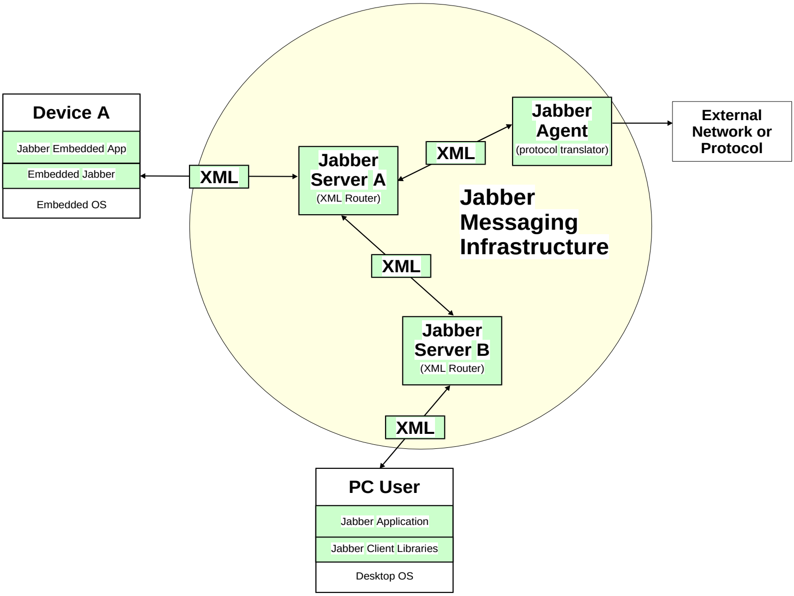

Figure 3: Jabber Architecture Model

The Jabber server is the main component in the instant messaging architecture. It performs two primary roles:

- Watching for client requests and communicating directly with client applications;
- Communicating with other Jabber servers.

The Jabber server has a modular architecture. It consists of several main modules providing respectively the following functionality: user authentication; data storage (user information, offline messages etc.); delivering messages, etc. Additional functionality can be easily added to the server. Thus, the so called "*transports*" or *protocol translators* could be added easily. With their help, a Jabber server communicates with any non-Jabber messaging system (see the Jabber Agent on Figure 3).

The XML technology takes main place in Jabber's realisation. It supports architecture design that is fundamentally extensible and able to use almost any structured data. The communication among architecture's components is realised by the help of *XML Streams*. Any user program creates a XML stream. Through it, the program sends and receives information in XML message form. In a similar way, the communication between two Jabber servers is realised. When the work ends, XML stream does the same .

The communication among Jabber users is realised in a way similar to the Internet e-mail. The user software sends a message to the Jabber server, which examines the request. If the recipient is a user of the same server, then the message is sent directly to this user. Otherwise, the message is sent to the *Etherx* component for processing and transmitting.

Etherx takes care of the communication among Jabber servers. In this case, the component will directly connect to the *Etherx* component of the remote server, whose user should receive the message. So the received message from the remote Jabber server is sent directly to the recipient, if he connects to the server at this moment. The message is saved for further notification, if the customer is offline at the moment.

Any Jabber user has own identifier (JID). This identifier is very similar to the e-mail address of the user, i.e., john@jabber.org. The identifier includes the user name (john) and the name of the server, which serves this user (jabber.org). JID, so constructed, allows any Jabber server to find the respective server of the distant user, realising effectively distributed system architecture.

In addition to these components, JID comprises the so-called "*resource*" of the user. The resource is used for identification of different sessions of a user, allowing him or her to connect to the system simultaneously from several different applications. The resource is added to the identifier, as it is written after the server name, separated by a slash for example:

john@jabber.org/Work

john@jabber.org/Home

Another important architecture feature is the maintenance of "*presence.*" This is information that shows when a user of a server is in a condition to receive instant messages. Any Jabber client is "at the disposal" when connected to the system and "*absent*" in the rest of the time. The main idea of the presence is to inform any customer if his interlocutors are also online and could communicate. This feature is realised as any user "*subscribes*" for other clients' presence. This system for subscription allows any user to permit the system to inform some users about his presence and to forbid this for other users. The Jabber server maintains a list with these subscriptions for any user. This list is called *roster*. It is saved on the server automatically, so the information about interlocutors is always available for users independently of the computer and user software. The Jabber server maintains the presence automatically when other users allow or forbid enquiries for a subscription. Additional information about user distribution in groups and their nicknames is saved in the *roster*. The user provides this information, and the roster serves for facilitation.

Architecture of the e-tendering system

The basis of any Internet application is the operational system that provides for communication in this environment. The general view of the e-Tender system's architecture is represented on Figure 4. Obviously, it does not cover the issues of the centralized server and the number of user computers. The components, building the business logic and database, are realized on a workstation. The architecture includes Jabber server and a database.

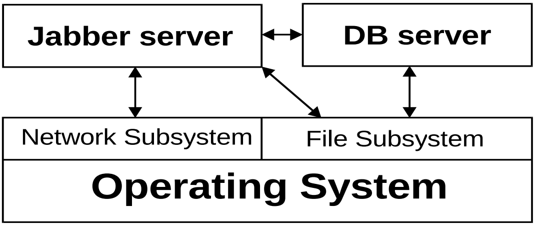

Figure 4: General of e-Tender system architecture.

The Jabber component accomplishes following functions:

1. Watches for incoming user requests, that are received from clients by the network subsystem;
2. Verifies JID and password, presented by the client, and permits or forbids the access of the user to the system;
3. Generates messages about presence of users and sends them to the network subsystem for transfer to clients;
4. Receives messages from users and processes them in accordance with the business logic, and generates needed messages about results;
5. Exchanges operational information with the database;
6. Processes saved messages about offline users, operating on the file subsystem;
7. Processes lists of friends and user preferences, operating on the file subsystem.

The database component accomplishes the following functions:

1. Operating on the file subsystem, it saves and indexes operational information on a permanent storage device;
2. Executes orders for access to and changing of operational information, received from the Jabber component.

The client software runs on user computers. The main role of the client software is to provide the user interface to the system. The system architecture includes user interface component and communication component, represented on Figure 5.

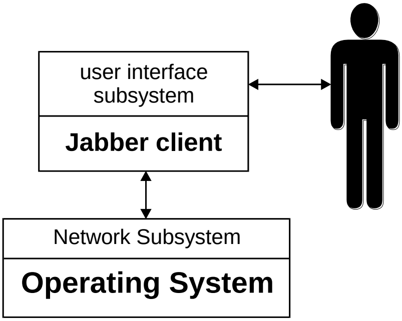

Figure 5: Interface and communication features.

The user interface must allow convenient work. The accomplishment of this requirement is quite difficult in most of the cases, and it necessitates thorough discussion with users after creating the prototype of the product. Therefore, the interface implementation needs to be separated as a component, thus allowing easy interface modification without a program breach. The communication component connects to the server, encodes and sends messages, already received from the interface, and decodes and verifies replies, which are received from the server. The exchanged messages contain information about the user presence, messages between users (chat), announcements of offered price, as well as office messages (*start* and *end* of tender, access to the list of user friends, etc.).

Data base architecture

To this aim, the application has to keep operational information about users, tender goods and announcements. The tender implementation is connected to the bidding of different users for any article. Therefore, the announcements must contain information about the buyer of the article and additional requirements in accordance with the Law on Public Tenders (LPT). The intention is to use relational model for data storage allowing designers to achieve the best results in presenting of this type of information.

Business logic definition

The business logic is defined to a great degree by the necessity to realise a public tender where sellers participate from different geographic regions. Traditional variants of public tender implementation are not applicable to this requirement. Therefore, integrating advanced Internet technologies, we propose an alternative that preserves to a large extent the conventional model of tender while allowing for participation of remotely situated subjects, e.g., companies.

Economically, the implementation allows large savings from rent of proper premises and for publicity. In the electronic variant, publicity is realized through an *electronic portal*, thus saving from printing and distribution of advertising materials. The expenditure for computer equipment is also smaller than spending to rent an auction auditorium, proper for implementing auction activities. Last but not least, buyers are able to participate without the necessity to leave their work places or homes. Of course, such solution is directed at clients that have the knowledge for using computers and the Internet. The organizers must know the technologies used in order to respond adequately to potential problems and inquiries of consumers.

The traditional model of the public tender is used as foundation of the business logic. Information about required goods is published in advance on the virtual site of the organizer (called also *host*). Data about the server of the organizer, date and place of the tender are also published on it. All participants, that are interested, are invited to send needed documents in order to create their user profiles in the system. The organizer verifies all received applications and creates respective user profiles. All approved users receive messages, where their personal data for registration in the system (user name and password) is documented. The users in turn obtain software necessary to connect to the system at the announced time. The host (or user with delegated rights) will greet all clients and will give them explanations, if that is necessary. This is realized by sending instant messages to tender participants.

When the host decides that all of interested clients are online, he or she starts the tender session. The system turns into "acceptance" mode for proposals and, at the same time, suspends new user registrations. Any participant receives a list of all client names, included in the tender session, and instant information about announcements of the rest of the participants. In this regard the electronic variant aspires to maximum resemblance with the traditional one. If there is no sufficient interest on the part of clients, the host may decide to suspend the bidding. So the client, that has announced the best price, receives the right of buying. The system automatically names the winner and his announced price. With this, the tender session about this article ends and the system turns again into "*registration*" module for users. Thus other users are given an opportunity to be online for the next tender session. At this time the host chooses the next article, which is offered on the tender and fixes a time in order to answer new questions. After of all tender sessions are closed, users that have received rights to buy are invited to connect to the organizers by phone or e-mail in order to clarify means of payment and delivery.

The system will maintain functions of instant communication among users, exchange of data and opinions. The aim is to make the tender a means for informal meeting of clients, as in the classic variant of the auction.

Conclusion

The successful realization of a public web site for electronic submission of documents facilitates the achievement of the following objectives:

- public access and transparency of tenders;
- confidentiality of any company information provided and proposals made by the prospective contractors;
- possibility to ask for the opinion of "*independent*" experts and, thus, to expand the database for analysis and selection of a company that is appropriate for the realisation of a particular project.

Another main advantage is the possibility to create a database of all companies that "*successfully*" execute orders and those, where project implementation is problematic. The software realization of such e-tender would also create excellent opportunities for accelerating the accomplishment of "*express*" requests, for example, procurement of perishable products. In addition, all those interested would be able to check what are the criteria for choosing a contractor, as well as whether criteria and procedure are in line with the Law on Public Tenders. As a whole, this problem is related to the recently widely discussed problem of the e-Government.

Notes:

[1] For detailed description of requirements, achievements and challenges refer to the compendium *Transparency in Defence Policy, Military Budgeting and Procurement*, ed. Todor Tagarev (Sofia: Geneva Centre for the Democratic Control of Armed Forces and George C. Marshall – Bulgaria, 2002).

[2] Walsh Norman, *What is XML?* < http://www.xml.com/pub/a/98/10/guide1.html> (28 May 2002).

[3] *CryptoBytes Newsletter* (RSA Data Security Inc., 1997), <http://www.rsa.com> (12 March 2003).

[4] Bruce Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edition (New York, NY: John Wiley & Sons, 1996).

[5] *Advanced Encryption Standard* FIPS-197 (Computer Security Resource Center, National Institute for Standards and Technology) <http://www.nist.gov/aes/> (14 June 2003).

[6] Saint-Andre Peter, *Jabber Technology Overview*, <http://docs.jabber.org/general/mhtml/overview.html> (03 June 2002).

GEORGI PAVLOV – see p. 147.

VESELINA ALEKSANDROVA is Assistant professor at the Interoperability Department of the "G.S. Rakovski" Defence and Staff College in Sofia, Bulgaria. She holds a M.Sc. degree in Computer Science from the Technical University of Sofia (1990), graduating in addition a course on Applied Mathematics and Informatics, Institute of Applied Mathematics and Informatics, Sofia (1992), and a Course for additional specialty of International Economic Relations at the University of National and World Economics in Sofia (1989). Her main research interests are in analysis and design of information systems, database systems, and networks. *E-mail*: alexandv@md.government.bg