A PROPOSAL
FOR
TACTICAL MESSAGING AND USAGE OF
EXTENSIBLE MARKUP LANGUAGE MESSAGE TEXT FORMATS
IN THE TACTICAL COMMAND CONTROL AND INFORMATION SYSTEMS

A THESIS SUBMITTED TO

INFORMATICS INSTITUTE

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

HÜSEYİN SAYIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN THE DEPARTMENT OF INFORMATION SYSTEMS

NOVEMBER 2003

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I alse declare that, as required by these rules and conduct, I fully cited and referenced all material and results that are not original to this work.

Hüseyin SAYIN
Author

Approval of the Graduate School of Informatics

Prof.Dr. Neşe YALABIK

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Assoc.Prof.Dr. Onur DEMİRÖRS

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science Science in Information Systems.

Assoc.Prof.Dr. Ahmet ÇOŞAR

Supervisor

Examining Committee Members

Prof.Dr.Semih BİLGEN

Assoc.Prof.Dr. Ahmet Coşar

Assoc.Prof.Dr. Ali DOĞRU

Assoc.Prof.Dr. Halit OĞUZTÜZÜN

Assoc.Prof.Dr. Onur DEMİRÖRS

ABSTRACT

A PROPOSAL
FOR
TACTICAL MESSAGING AND USAGE OF
EXTENSIBLE MARKUP LANGUAGE MESSAGE TEXT FORMATS
IN THE TACTICAL COMMAND CONTROL AND
INFORMATION SYSTEMS

Sayın, Hüseyin

M.S. Department of Information Systems

Supervisor: Assoc.Prof.Dr. Ahmet COŞAR

November 2003, 146 pages

This thesis investigates the usage area of Extensible Markup Language Message Text Format (XML-MTF) in Tactical Command and Control Information System (TC2IS). It examines the used tactical message types and their application area in Turkish Army and what the XML and XML-MTFs are explained. Finally; MTF traffic of a brigade is simulated to verify that XML technology can be used effectively in Turkish Land Forces Tactical Command Control and Information System Projects (TLF TC2IS).

Keywords: XML, MTF, XML-MTF

ÖZET

TAKTİK MESAJLAŞMA VE
TAKTİK KOMUTA KONTROL BİLGİ SİSTEMLERİNDE
GENİŞLEYEBİLİR İŞARETLEME DİLİ MESAJ TEKST FORMATI
(XML-MTF)NIN KULLANILMASI

SAYIN, Hüseyin

Yüksek Lisans, Bilişim Sistemleri Bölümü

Tez Yoneticisi: Doç. Dr. Ahmet COŞAR

Kasım 2003, 146 sayfa

Bu tez Taktik Komuta ve Kontrol Sistemlerinde (TKKBS) XML'le Tekst Tabanlı Mesaj Formatlarının kullanılması konusunu analiz etmektedir. Taktik mesajlaşma çeşitleri ve Türk Silahlı Kuvvetlerinde uygulama alanlarını incelenmektedir. XML ve XML-MTF'in ne olduğunu açıklanmıştır. En sonunda da, Kara Kuvvetleri Komutanlığı Taktik Komuta Kontrol ve Bilgi Sistemeleri projelerinde XML teknolojilerinin etkin olarak kullanılabilirliğini doğrulamak için bir tugayın mesaj trafiği simüle edilmiştir.

Anahtar Kelimeler: : XML, MTF, XML-MTF

ACKNOWLEDGEMENTS

I express sincere appreciation to my advisor, Assoc.Prof.Dr. Ahmet COŞAR for his guidance and patience in reading my long preparations during the thesis work and completing simulation program. I especially thank Assoc.Prof.Dr. Ahmet COŞAR for his support and his help during my long illness period. I also thank Prof.Dr. Semih BİLGEN for his guidance and good proposal about message traffic simulation of a brigade which forms my thesis basics. Assoc.Prof.Dr. Halit OGUZTÜZÜN's guidance and endurance about preparation of concepcual model becomes an obligatory duty for me to thank. I also appreciate my wife Zeynep and all other friends, for their understanding my frequent absences. To my wife, Zeynep, I offer sincere thanks for great faith in me and her willingness to endure with me. To my child, Barış Can, I thank him for understanding my frequent absences.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ACRONYMS

ADatP-3	: Allied Data Publication Number 3
ADOP-2000	: Fire Support Automation Project 2000 (Ateş Destek Otomasyon Projesi 2000)
BLOS	: Beyond Line of Sight
bps	: bits per second
BWT	: Burrows-Wheeler Transform
C2	: Command and Control
C2IS	: Command and Control Information System
C4I	: Command, Control, Communications, Computers, Intelligence
C4ISR	: Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CIS	: Communication and Information Systems
CNR	: Combat Net Radio
COTS	: Commercial-Off-The-Shelf
CSS	: Cascading Style Sheets
DCF GML	: Document Composition Facility Generalized Markup Language
DTD	: Document Type Definition
DSSSL	: Document Style Semantics and Specification Language
ECM	: Electronic Counter Measure
FMF	: Fixed Message Format
GML	: Generalized Markup Language
HERİKS	: Air Early Warning System (Hava Erken İkaz Sistemi)
HTML	: Hypertext Markup Language
IERs	: Information Exchange Requirements
ISO/IEC	: International Standard Organization / International Electrotechnical Commission
IT	: Information Technology
kbps	: Kilobit per seconds
KEİS	: Land Integrated Information System (Kara Entegre İstihbarat Sistemi)
KWOC	: Keyword-Out-of-Context
LBS	: Logistics Information System (Lojistik Bilgi Sistemi)

LZ	: Lempel-Ziv (compression)
MA3	: Milli Açık Anahtar Altyapısı
MEFORS	: The Turkish Army Message Formatting System (Mesaj Formatlama Sistemi)
MKS	: Maneuver Control System (Manevra Kontrol Sistemi)
MS	: Microsoft
MTF	: Message Text Format
NADGE	: NATO's Air Defense Ground Environment
NATO	: North Atlantic Treaty Organisation
NATO FORMETS	: NATO Formatted Message Text System
NILE	: NATO Improved Link Eleven
RFC	: Request for Comments
RLE	: Run-Length Encoding
RTF	: Rich Text Format
SAM	: Surface to Air Missile
SGML	: Standard Generalized Markup Language
SHORAD	: Short Range Air Defense System
SSSBs	: Ship Shore Ship Buffers
TADIL	: Tactical Digital Information Link
TAFICS	: Turkish Armed Forces Integrated Communication System
TASMUS	: Tactical Area Communication Systems/Taktik Saha Muhabere Sistemi
TDLs	: Tactical Data Links
TDM	: Time Division Multiplexing
TDMA	: Time Division Multiple Access
TAF	: Turkish Armed Forces
TLF	: Turkish Land Forces
TLF C2IS	: Turkish Land Forces Command Control and Information System Projects
TLF IEDM/ KKBDVM	: Turkish Land Forces Information Exchange Data Model / Kara Kuvvetleri Bilgi Değişim Veri Modeli
UHF	: Ultra High Frequency
URL	: Uniform Resource Locator
USMTF	: United States Message Text Format
VMF	: Variable Message Format
www	: World Wide Web (W3)
W3C	: World Wide Web Consortium
XSL	: Extensible Style Language
XLink	: XML Linking Language
XLL	: Extensible Linking Language

XML	: Extensible Markup Language
XML-MTF	: eXtensible Markup Language Message Text Format
XPointer	: eXtensible Markup Language (XML) Pointer Language
XSL	: eXtensible Style Language

CHAPTER 1

INTRODUCTION

"**Digital information is rapidly becoming integrated into all aspects of military activities. Operations are becoming increasingly fast-paced and diverse as the management of information moves closer to the center of military power. To provide commanders with the right information is required to make decisions in this environment, a greatly enhanced command and control (C2) concept for information gathering, dissemination and visualization is needed, which is based on revolutionary new information-age concepts and technologies.**"[1] The battlespace infosphere must allow for innovation and rapid acquisition of new mission. It also should support the commander with the means to find and retrieve information, to combine and manipulate it. An "**information-centric**" approach is needed for systems to work together effectively. Common information objects, such as military message text formats (MTFs), enhances interoperability.

The fundamental requirement addressed by all Turkish Land Forces Command Control and Information Systems Projects (**TLF C2IS** - MKS, KEİS, LBS, ADOP, HERIKS) is "*to provide the right information at the right time, disseminated and displayed in the right way, so that decision makers can do the right things at the right time in the right way*". [2]. The concept of building and interconnecting systems around the information that they manage and exchange is the future vision for **TLF C2IS**. These projects require extensive and constant information exchange among them, between Turkish Armed Forces Units and other governmental organizations.

A great deal of time and resources have been invested to address Command Control and Information Systems (C2IS) interoperability problem. Today, MTF program governs a significant portion of data exchanged to support the full spectrum of military operations. MTF is one of the primary means for exchanging information within Turkish Army. Information technology will act as a force-multiplier to enhance operational effectiveness at every level of C2 systems. We will be able to find, retrieve, process and exchange large volumes of information by using current industry standards. Custom formats are needed to exchange information between different organizations. These acquisition efforts have aggressively sought commercial off-the-shelf software that can be adopted to meet unique military requirements.

Extensible Markup Language (XML) is a metalanguage that has a universal format and provides a common platform for exchanging information. Its format is recognizable by both humans and machines. XML allows developers to easily describe and deliver rich and structured data from any application in a consistent and standard way. A role is being the bridge between structured data and the unstructured real world. [3]

XML will increasingly become the tool of choice for exchanging information. XML may be used to enhance interoperability and information exchange among Command, Control, Communications, Computers, and Intelligence, (C4I) systems. XML is industry standard, inexpensive, powerful, and rapidly evolving, being driven by the dynamics of the commercial marketplace. The simplicity, flexibility and low development cost of XML will make it attractive in today's environment. It does fill a gap in the existing web architecture. Its flexibility will enable integration with existing technologies as a data exchange format. [3]

For decades, Turkish Army has analyzed, and is still going ahead with Message Formatting System (MEFORS) project, and has documented its information exchange requirements in great detail. Structured message text semantics and formats implementing those requirements were agreed upon a long time ago and are stil in use. Turkish Land Forces have an ongoing corporate data modeling effort (**TLF C2IS), and** will have a data administration organization. This provides an excellent data model to which all XML tags and Document Type Definitions (DTDs) must comply and refer. It also has an emerging high level architecture in which the place and role of XML could be clearly identified. What XML can contribute are modern information encodings for Turkish Land Forces, and Turkish Army. By using the defense information exchange semantics and the benefits of strong software industry, ***XML can give defence systems of Turkish Land Forces better interoperability and greater capabilities at lower costs.***

The overlap in requirements of military information systems and commercial information systems is steadily growing. XML technology and With XML, it still seems possible to influence its further development to include essential defence requirements, as these standards have not been developed specifically for defense applications in mind. XML can be used to modernize military information standards through commercial technologies. Current army wireless technology has limited transmission bandwidth. Messaging is vital for all millitary organizations. XML cannot be thought as stand-alone. Extensible Markup Language Message Text Format (XML-MTF) will become the tool of choice for Turkish Land Forces. But, ***XML is verbose.*** There is a lot of "**redundant**" data in an XML document. The structure of messages, the optimal network structure with consideration of organization's hierarchical (and communication) structure needs to be considered and then planned. In this thesis, I investigated the feasibility of using XML for exchanging military messages using currently available communication media. The simulation results given in this thesis can be used to decide whether available bandwidth would be sufficient to handle the traffic load generated by XML formatted messages. For this reason, an imaginary brigade is formed, and its message trafic is analyzed for Turkish Land Forces. A detailed structure of the brigade (the organisational hierarchy) is provided in Figure 1.1.

XML-MTF endeavor is likely to be successful. It is a low-cost, enhanced alternative to current methods of information exchange. *If all these efforts succeed (and there is every rason to believe that)* ***XML can give defence systems of Turkish Land Forces better interoperability and greater capabilities.***

Organization of the Thesis

This thesis consists of two main parts. First part starts with an overview of effective Command and Control Information Systems (C2IS) and tactical area digitization. Then it presents Tactical Information Exchage Standards. The aim of the first part is to make reader familiar with digitization challenges about effective C2IS data exchange and current situation of military data exchange mechanism with messages, overview of XML technology. This part is also a base for the current Army Tactical Information Exchange Model.

Second part of the thesis deals with XML MTF and requirements to develop a military message text format architecture for Turkish Land Forces. A simulation approach is used to analyze a model because of the organizational outlook of a brigade. Starting with the general outlook of the brigade, the main maneuvre units of an army, and the lower level formations, battalion and company level nodes, and the connections between them are modeled. Then the interactions of the formations with one another are developed.

Chapter 2 starts with an overview of data exchange concepts. Different messaging technologies are analyzed. This chapter also covers tactical information exchange standards, that are fixed size and bit oriented data exchange with MTF, and database-to-database message exchange. The aim is to explain the details of MTF in order to help the reader to understand the details of MTF based data exchange.

Chapter 3 introduces markup languages, explains what XML is, and emphasizes XML basics.

Chapter 4 introduces XML MTF and summarizes the current situation of XML-MTF applications. Current MTF which is used by TLF is compared with XML-MTF.

Chapter 5 describes MTF based C2 messaging simulation tool and flow of an imaginary brigade’s message trafic is analyzed for Turkish Land Forces Tactical Command Control Information Projects (**TLF TC2IS**).

Chapter 6 summarizes conclusions and discusses the possible related future work.

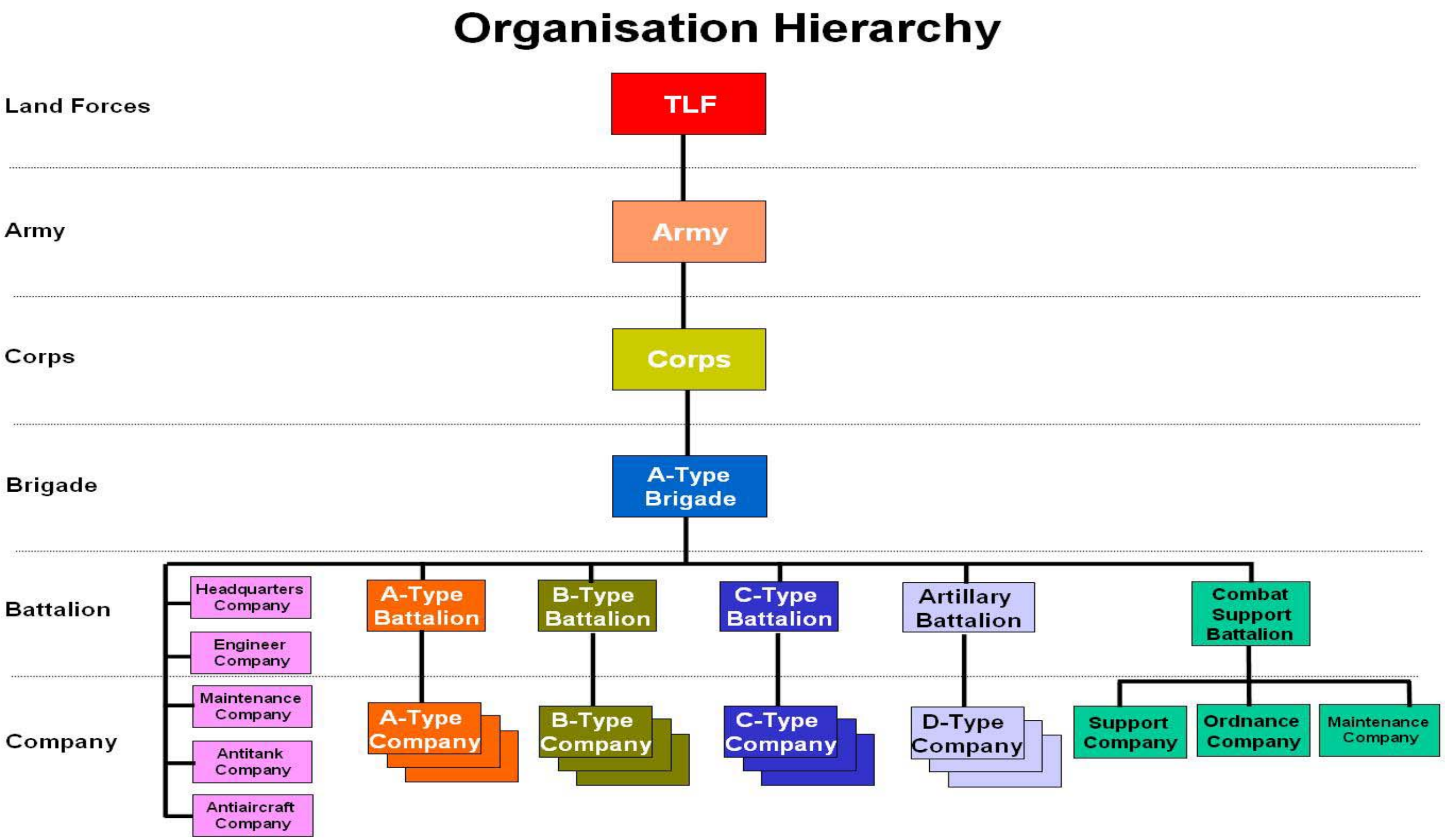


Figure 1.1 : Hierarchy of the modeled military units.

CHAPTER 2

DATA EXCHANGE WITH MESSAGE TEXT FORMAT

2.1. Overview of Data Exchange

Information has been at the core of military operations through the ages. Throughout history, military leaders have recognized the key role of information as a contributor for victory on the battlefield. Commanders have always been sought a decisive information advantage over their adversaries. It is sometimes gained. The writings of both Sun Tzu and Clausewitz reflect the key role of information in warfare. Sun Tzu, writing 2,500 years ago, emphasized the importance of knowledge in war. "*The general unreliability of all information presents a special problem: all action takes place, so to speak, in a kind of twilight, like fog. War is the realm of uncertainty; three quarters of the factors on which action in war is based are wrapped in a fog of greater or lessor uncertainty. The commander must work in a medium which his eyes cannot see, which his best deductive powers cannot always fathom; and which, because of constant changes, he can rarely be familiar.* "[4].

In the modern world, ***timely***, ***consistent*** and ***understandable*** information exchange between headquarters and operational units (goverment agencies, defence establisments) make C2IS effective. Interoperability must also provide unambiguous information exchange. The definition and implementation of a standard messaging protocol is important for military applications. Different protocols have been preferred for different families of weapon systems, that is based on the needs and limitations of their individual context. A standard messaging protocol, that has been used by all nodes, was needed to obtain the desired interoperability.

Future military operations will be more complex and unpredictable than those of the past. ***Digitization must be able to decrease decision-making time. Optimizing the flow of information delivers some operational benefits.*** Information can be sent and received by a wide range of systems. In an ideal world, every systems are similar to each other and have the same database, the same operating system, the same computing platform, and so on. Thus it makes information exchange much easier. However, that is not the case. It would be only the most optimistic scenario. [4][5]

2.2. Tactical Information Exchange Standards

The cost of replacing legacy systems with improved database and data interchange technology must be considered. Solutions which meet current operational requirements are needed for existing systems. Formatted messages are the primary means for information exchange between heterogeneous systems in the existing C4I world. Basic tactical information exchange standards are:

1. Bit-oriented data exchange,
2. Data exchange with MTF messages,
3. Database-to-database exchange,

2.2.1. Bit-Oriented Data Exchange

Bitwise data exchange in military is generally based on **T**actical **D**ata **L**inks (TDLs). TDLs provide the specialized information infrastructure that underpins situational awareness and C2 in the **maritime** and **air** domains. They enable the combat systems of aircrafts, ships, C2 units, ground based air defence batteries and others to exchange digital information rapidly to engage the tactical battle. [6]

The family of bit-oriented data exchange formats will be presented in the following sections and consists of

1. Tactical Links
    a. Link1
    b. Link-4/4A
    c. Link-11/11B
    d. Link-16
    e. Link-22
2. Fixed Message Format (FMF)
3. Variable Message Format (VMF)

2.2.1.1. Tactical Links

Link-1 Architecture: Link 1 is a non-secure data link. It is used for exchange of air surveillance data between control centers and combined airsector operation centers. Within Turkish Air Forces, Link-1 is used by "air defence ground systems" and exchanging radar scope images between radar units. It is used for early warning purposes most North Atlantic Treaty Organisation (NATO) nations. Message standards are defined in STANAG 5501[1] [7] while standard operating procedures are laid down in AdatP-31[2] [8]. [9]

Link-4 (4A/4C) Architecture: Link-4 is a non-secure data link which is used for providing data to fighters. Link-4A uses a command and response protocol. It provides simultaneous channels from a given frequency spectrum by using the principle of Time Division Multiplexing (TDM). *Link 4C is a fighter to fighter data link.* It complements Link-4A; but they do not communicate directly with each other. Message standards are defined in STANAG 5504[3] [10]. Its standard operating procedures are laid down in AdatP-4[4] [11]. [12]

Link-11/11B Architecture: Link-11 is based on 1960s technology. It is a relatively slow link. It normally operates in a polling mode in a network. It has a network control station that is polling each participant in turn for their data.

Link-11B (*Tactical Digital Information Link-B (TADIL-B)*) is used for data exchange with a dedicated, point-to-point, fully automatic, phase-continuous, full-

---

[1] This document is not publicly available. It is classified as "Mission Secret".
[2] This document is not publicly available. It is classified as "Mission Secret".
[3] This document is not publicly available. It is classified as "Mission Secret".
[4] This document is not publicly available. It is classified as "Mission Secret".

duplex digital data link. The exchange of air, surface and subsurface tracks, electronic warfare data and limited command data among C2 units is supported by Link-11. Ground Based Surface to Air Missile (SAM) Systems are (or will be) equipped with Link-11. Link 11B is used for Ground Based SAM Command, Control and Fire Distribution Centers and Air Defence Ground Environment integration. Message standards for both Link 11 and Link 11B are defined in STANAG 5511[5][13] while standard operating procedures are laid down in AdatP-11[6][14]. [15]

Within Turkish Air Forces, Link-11 is used by systems to exchange tactical air support data and radar scope image between Navy Surface Radar Units and Air Force Radar Units. Within Turkish Navy, it is used for exchanging tactical data (radar images) between Navy Ships and between Navy Land Radars (It is called Ship Shore Ship Buffer). NATO primarily uses Link-11 as a Maritime Data Link. However, Link-11 will be adapted to cater for theatre missile defence information exchange requirements.

Link 16 Architecture: The general purpose of Link-16 is the same as that of Link-4A and Link-11. It has been developed to meet the information exchange requirements (*real-time tactical data)* of all tactical units. It supports the exchange of surveillance data, electronic warfare data, mission tasking, weapons assignments and control data like Link-11. Link-16 transmits broadband data across a variety of air, naval and ground-based platforms. [16]

Link-16 distributes tactical data and digital voice information at high rates among land, surface and airborne units with extensive jam resistance. Primary Link-16 functions include the exchange of friendly unit position and status data, the dissemination of tactical surveillance tracking data, and control/management of surface, subsurface and air engagement. The resulting multiple data link battle groups will be able to more efficiently conduct operations throughout the worldwide environment. [17]

---

[5] This document is not publicly available. It is classified as “Mission Secret”.
[6] This document is not publicly available. It is classified as “Mission Secret”.

The US and NATO selected Link-16 as the main tactical data link for theatre missile defence. Message standards for Link-16 is defined in STANAG 5516[7][18] while standard operating procedures are laid down in AdatP-16(A), Ch 1[8][19].

Link-22 Architecture : Link-22 is a "Link-16 family member". Initially it was evolved as NILE (NATO Improved Link Eleven). Link 22 is something of a hybrid between Link-16 and Link-11. It uses Time Division Multiple Access (TDMA) or Dynamic TDMA (DTDMA) architecture. [20]

Link-22 is defined in STANAG 5522 [21] while standard operating procedures are laid down in AdatP-22. [22]

2.2.1.2. Fixed Message Format (FMF)

FMF is fixed length messages format which is ***suitable for near real time data exchange in a bandwidth constrained combat environment.*** It has been developed to meet the information exchange requirements (*real-time tactical data)* of air defence units, supporting the exchange of surveillance data, mission tasking, weapons assignments and control data. There must be an agreement on byte ordering of data which is sent on the wire, regardless of format of data. It is not man-readable. In theory, this could be pushed all below the socket layer, but in practice certain fields in the socket address structure must be maintained in network byte order.

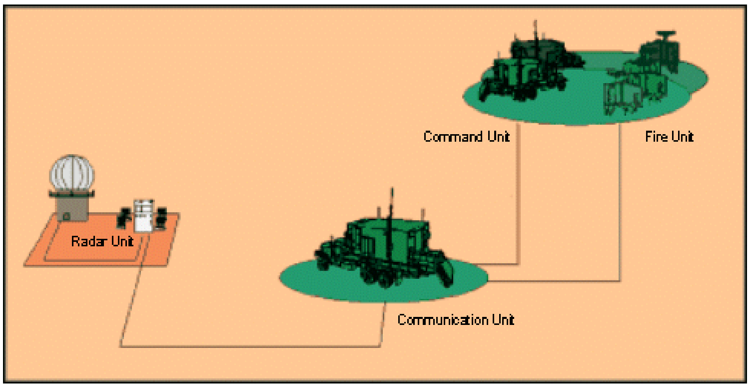


Figure 2.1: TLF HERIKS FMF communication.

---

[7] This document is not publicly available. It is classified as "Mission Secret".
[8] This document is not publicly available. It is classified as "Mission Secret".

FMF is used in TLF HERIKS (Early Warning Ground System/Hava Erken İkaz Sistemi) for internal communication, from a ground radar to a fire support unit which is shown in Figure 2.1.

2.2.1.3. Variable Message Format (VMF)

VMF is another "Link 16 family protocol". It is **suitable for near real time data exchange in a bandwidth constrained combat environment** (e.g., over Combat Net Radio (CNR) and other tactical digital entry devices.). It provides a common means of exchanging digital data between combat units at varied echelons. VMF provides an interoperability standard which includes message, data element, and protocol standards. Link-16 data elements are used to create variable length messages. It is not man-readable. **The sender only transmits fields that contain data.** It provides flexibility to select/omit particular fields to be transmitted in a message. It is the option to transmit fixed-format messages. The textual content of the message is seperated from the communications or network functions. [23]

It is an extremely flexible message standard. Information only required at send time. Both of the message fields and message fields length is variable, so the message length is variable. Implementation of the protocol is complex and difficult. The amounts of data and detail of information. varies. Thus, variable format minimizes the bandwidth needed for transmission by reducing the input/output load (the overall bit count). VMF is the army solution to support land combat operations of a brigade and below. It solves ***battlefield digitization interoperability and bandwidth problems,*** VMF accommodates existing character-oriented MTFs and bit-oriented TADILs message standards.

The information and addressing portions can be selectively adapted to suit the situation. Data fields can be selected or omitted from a message as required. "Null" or “zero” filled fields are not required to prevent unavailable or redundant information transmission. VMF is intended to be filling the gap between Link-16 and some other TDLs and MTF do.

2.2.2. Message Text Format (MTF) Messages

Before digital computers and communications were readily available, voice-based procedures were defined to facilitate information exchange and understanding between human operators. With the advent of computers, text was preferred over voice. Thus, the voice communication moved to digital text based procedures. That is implemented via sets of messages.

For messages, a natural language in a normal free text mode can be used. However, this procedure has serious disadvantages. The free text messages often make rapid processing imposible. Sentences often contain words that can be omitted without loss of information. Words frequently have more than one meaning. These words can usually be represented either by abbreviations without loss of mnemonic value, or by codes already in existence that are accepted through general use. These disadvantages become even more obvious when computer assistance to message processing is needed.

To achieve a better solution for messages, two separate actions can be taken. First, an artificial language can be created with a vocabulary restricted to words (including abbreviations and codes) for which unambiguous meanings have been agreed by all participants. This is especially desirable in a community that uses multiple natural languages. Second, structures for this artificial language can be created so that as much information as possible is conveyed purely by the position of words within predetermined formats to complement the information provided by the words themselves.

The requirement for an artificial language for messages is not dependent upon the existence of computer assistance to message processing. However, a computer can be more efficiently programmed to process information presented in an artificial language, that uses only precisely defined words and predetermined formats, than to process information in a natural language. This is an important and significant additional benefit.

The exchange of information is an essential activity in any defence organisation. The information that is transferred should be **concise**, **accurate**, **up to date** and **readily understandable**. A standardized information exchange language must possess these characteristics. MTF supports the full spectrum of military operations. MTF was developed over a 30-year period. It is one of the primary means for exchanging military information within all armies. Information exchange requirements and military exchange formats have been agreed and codified. The implementation of these formats has been too expensive to allow general use in all defense systems. The required software had to be custom-built, exclusively for the military market.

Formatted messages are ***unambiguous**, **more clearly understood.*** They can ***help to overcome language differences.*** The*y* offer significant advantages over free-text messages. The structure, sequence and allowable contents of a formatted message is the difference between a formatted message and a free text message. Computer systems can automatically process formatted messages.

A formatted message is exchanged information with a formal language defined by MTF rules. There are different MTF standards in use, with variations in rules and basic structures. The various internationally agreed MTFs, for example are:

- MEFORS (Turkish Army Message Formatting System)
- NATO FORMETS (NATO Message Formatting Systems),
- USMTF (United States Message Text Format),
- US VMF(United States Variable Message Format),
- OTH-T (Over The Horizon Targeting) Gold,
- ADFORM (Australian Defence Formatted Message),

These standards ***describe the syntax*** *and* ***semantics of structured alphanumerical messages.*** Most of them were originally designed for teletypewriters or telex. They encode the agreements on information exchange for everything from logistics to intelligence reports.

Turkish Army Message Formatting System (MEFORS): The purpose of MEFORS is to improve interoperability among Turkish Army Units. It provides the rules, constructions and vocabulary for standardized Character-Oriented MTFs. That can be used in both manual and computer-assisted operational environments. It contains the reports used by Turkish Army (General Staff, Land, Navy and Air Force) which uses NATO Allied Data Publication Number 3

(ADatP-3) [24] rules. It is integrated with Turkish Army Message Distribution System (MEDAS).

NATO MTF System (FORMETS): The purpose of FORMETS is to improve interoperability between different national and NATO defense systems. FORMETS is a suitable artificial language that provides the rules, constructions and vocabulary for standardized character-oriented MTFs. It can be used in both manual and computer-assisted operational environments. The rules comprise the agreed conceptual definitions and arrangement of these within predetermined formats.

The syntax of FORMETS specifies the structures which can be defined through set, field and message formats. This language can be used by automated systems during the message preparation. Strict compliance with field format specification rules is used to validate the structure of a received message to protect against contamination of an operational database.

FORMETS contributes to and draws from other interoperability standards, such as agreed NATO terminology, to the optimum extent achievable. It is guided by the NATO Interoperability Management Plan.FORMETS has been used by the majority of NATO nations. The agreement between NATO nations to use this standard is recorded in Allied Data Publication Number 3 (ADatP-3) [24] and STANAG 5500[9][25].

United States Message Text Format (USMTF) : *USMTF* is an artificial language comprising rules, structures, and vocabulary to ensure the construction of standard character-oriented MTFs. It is limited to the teletype character set. It is the medium for information exchange between United States of America (USA) and international C2IS. USMTF is designed primarily for non-real-time exchange. It is the standard for man-readable and machine-processable information exchange. It reduces time and effort required for information exchange. USMTF standardizes the use of equipment and methods for exchanging information between all theater command and control elements. The standard is documented in MIL-STD 6040 [26] and Tactical Command and Control Planning Guidance for

[9] This document is not publicly available. It is classified as “Mission Secret”.

Joint Operations. The Joint Interface Operational Procedures (JIOP). [27] Both of them are used together.

2.2.3. Database-to-Database Exchange

A coherent and interoperable defence communication and information system capability is vital to military operations. In addition to information exchange by structured messages, database to database information exchange between systems is being used today.

Message standard, which is defined above, provides a means to define a message format (ie., transfer syntax), which describes the definition of the data elements. Other techniques provide direct database to database exchange of data as a service. They are invisible to the user. Main purpose is to exchange information directly between information systems without the constraint of formatted messages. "*The data elements, which are currently defined in various message standards, are not mutually consistent or based on any process or data models either within a message system or across message systems.*" [28] Achieving integrated and seamless interoperable systems need data oriented methodologies and technologies that work in harmony. Common database is a simple way that handles these requirements. All the systems use the same data. The identification of information and information interchange requirements can be provided by a single data model. It integrates all the tactical systems into a single database. The figurative explanation is shown in Figure 2.2[10].[29]

[10] It is taken from http://www.jta.itsi.disa.mil/jta/jta-v1.0/sect4_JTA_Section_4_Information_Modeling.htm [29] with minor changes. (Last visited September 18, 2002)

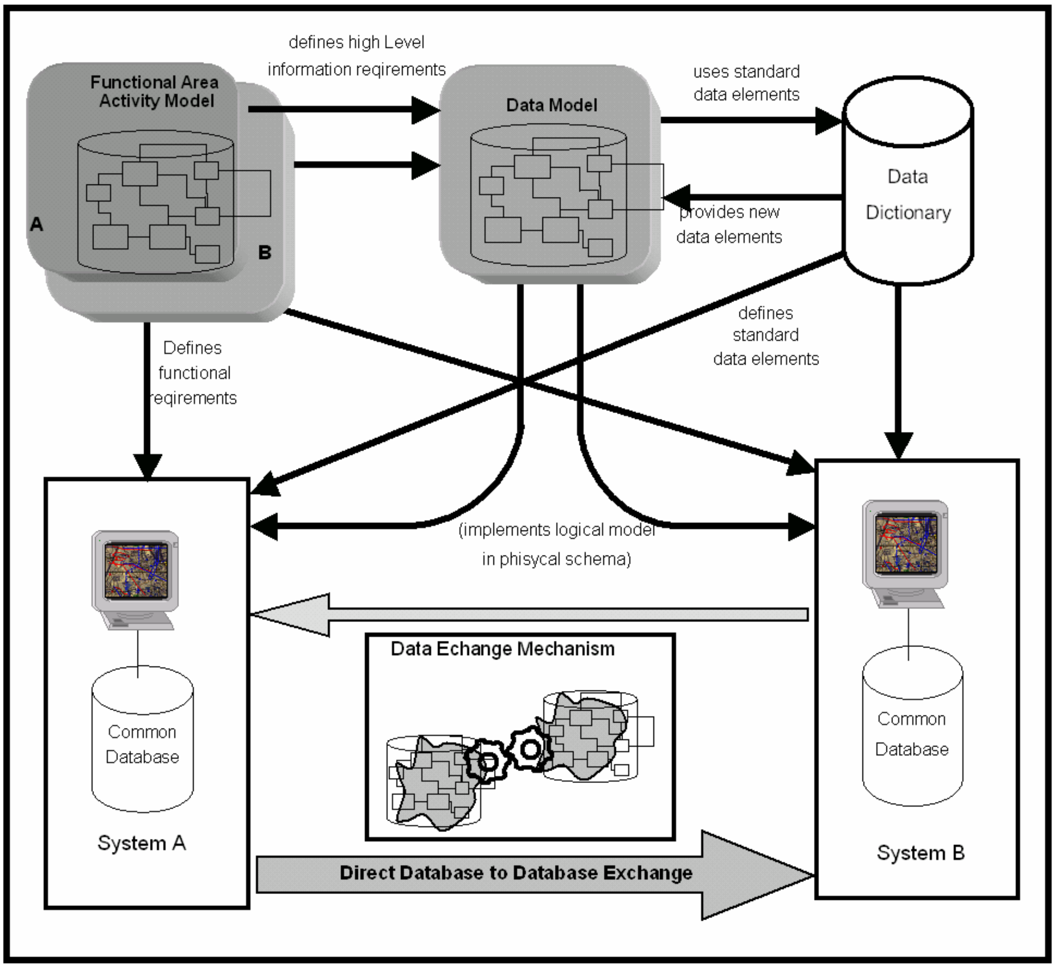


Figure 2.2 : Database to database exchange model [29]

Integration, interoperability, flexibility, and efficiency is important for information management. The development of a common, multi-purpose, standards-based technical infrastructure is the means to achieve these. These improve the effectiveness of functional operations and use of technology throughout national armies.

The concept of building and interconnecting systems around the information that they manage and exchange is the vision of Turkish Land Forces Command Control and Information Systems Projects (***TLF C2IS*** - *MKS, KEİS, LBS, ADOP, HERIKS*). These projects require extensive and constant information exchange among TLF units, and with other army units and governmental organizations.

Structure of the information is expressed in a data model. Current Information Exchange Requirements (IERs) of military units change over time. A flexible generic model for exchanging information is needed. Turkish Land Forces has been developing a common database architecture for all ***TLF C2IS*** projects.

TLF C2IS specifications consist of two main components: a ***data model*** and a ***replication mechanism***. Data model, which is called ***TLF IEDM/KKBDVM*** [30]***,*** is based on NATO Land C2 Information Exchange Data Model (LC2IEDM). It is a product of the analysis of a wide spectrum of ***TLF C2IS*** projects' information exchange requirements. It models the information that land component commanders need to exchange (both vertically and horizontally) of essential battlespace information. It forms a common framework for C2IS projects to exchange data between them. It will also serve as a basis, if ***TLF*** desired, for new systems. Replication mechanism, Turkish Land Forces Information Exchange Model **(*****TLF IEM/KKBDM***) [31]***,*** is complementary to ***TLF IEDM/KKBDVM*** data model. When a C2 application changes the state of information that it holds, and which is recognised by ***TLF C2IS*** specification, this information is automatically replicated to all other co-operating systems that have subscribed to receive this information. The meaning and context of the information is preserved and requires no additional processing on receipt to make it useful.

CHAPTER 3

MARKUP LANGUAGES AND XML

3.1. Overview of Markup Languages

Markup stays at the core of the print-based world. The term ***markup*** specifically refers to tagging electronic documents for one of two purposes:

- *to modify the look and formatting of text or*
- *to establish the structure and meaning of a document for output to some medium, such as a printer or the World Wide Web (www).*

Editing programs, such as Microsoft (MS) Word and MS Frontpage, use markup to accomplish formatting. In addition to formatting text, ***markup can work to determine the structure and meaning of textual elements.***

Markup is commonly used to change the look of text by adding formatting, such as bold or italic fonts, text indents, font sizes, and font weights. Markup tags typically work by turning these attributes on when they are needed and off when they are not. Two types of markup languages are in use today:

1. Specific markup languages,
2. Generalized markup languages.

3.1.1. Specific Markup Languages

Specific markup is used to generate code that is specific to a particular application or device. These markup languages are often built to serve a particular need like formatting documents for the web or text formatting and vice versa.

Hypertext Markup Language (HTML) [32] and Rich Text Format (RTF) [33] is most common specific purpose language that is used for the web. However, the markup code for HTML and RTF look different, even though both were created for similar purposes. The two languages, and others, are not interchangeable. A processor that understands RTF will not understand HTML, and vice versa.

Many markup languages are created for the specific purpose of **describing text formatting** and **layout** but not for any other purposes, Such specificity results in several limitations which is common to specific markup languages:

"Authors are limited to a particular *set of tags.*" ***"****A document* ***might not be portable to other applications."*** [34]. Because the data is **not self-describing.** It does not provide data about data (also called ***metadata***). So that the data in the documents can stand apart from the formatting. It cannot be used for any other purpose than that for which it was intended.

"*The Rich Text Format (RTF) Specification is a method of encoding formatted text and graphics for easy transfer between applications*." [35] "*An RTF file consists of unformatted text, control words, control symbols, and groups*." [36]. Figure 3.1 is a short memo document formatted in RTF using Wordpad:

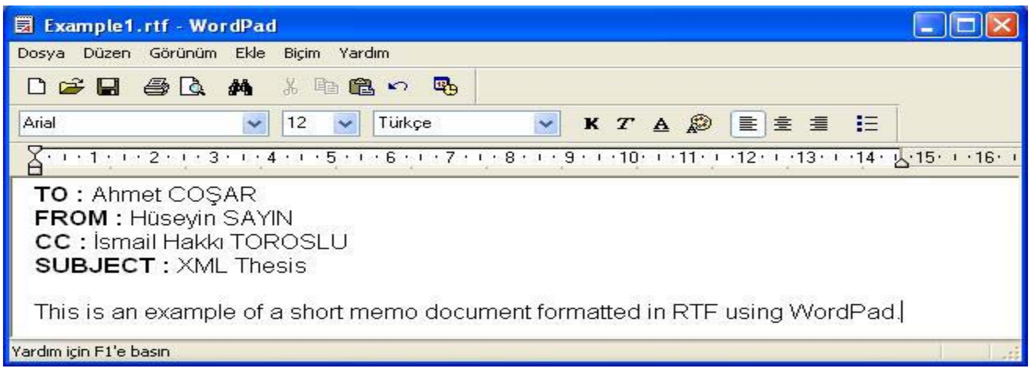


Figure 3.1 : A short memo example of RTF.

The text in Figure 3.1 looks like any other text when a word processor or desktop publishing program is used; or even while a Web page is viewed. This text has been formatted with RTF and saved as an RTF file. Here is how the tagged text, or code, for this document looks:

```
{\rtf1\ansi\ansicpg1252\deff0\deftab720{\fonttbl
    {\f0\fswiss Arial;}
    {\f1\froman\fcharset2 Symbol;}
    {\f2\froman Arial;}}
{\colortbl\red0\green0\blue0;}
\deflang1033\pard\plain\f2\fs20\b To: \plain\f2\fs20 Ahmet COŞAR
\par \plain\f2\fs20\b From: \plain\f2\fs20 Hüseyin SAYIN
\par \plain\f2\fs20\b Cc: \plain\f2\fs20 İsmail Hakkı TOROSLU
\par \plain\f2\fs20\b Subject: \plain\f2\fs20 XML Thesis
\par
\par This is an example of  a short memo document formatted in RTF using
WordPad.
\par }
```

Figure 3.2 : RTF short memo example tagged text code.

The tagged text looks much different than the displayed text, which is given example Figure 3.2. This text code tells the application everything about the application needs to know. Every tag describes how the text should be formatted. But it tells nothing about the meaning of text data included in the document. RTF code is actually quite readable as it is seen in above figure. However, this is not always true with other markup languages. If we look at a document that is saved in MS Word document format will not be readable at all. The memo document example saved as a MS Word document is shown as a unformatted file in Figure 3.3.

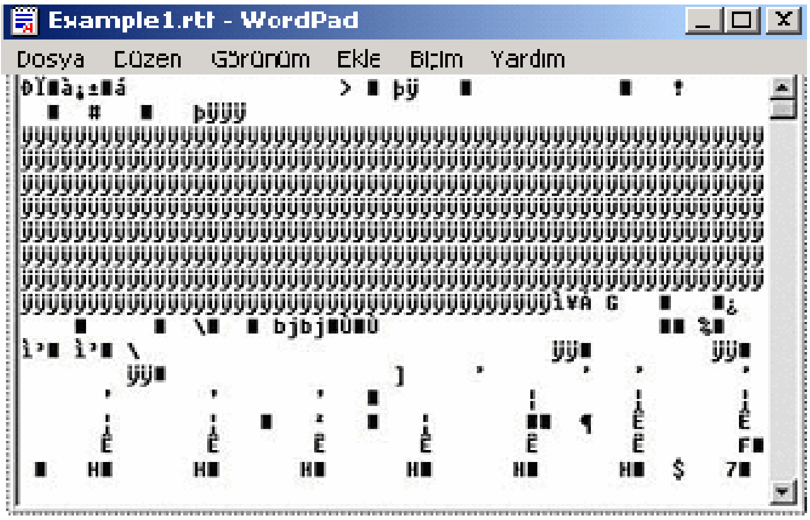


Figure 3.3 : Plain text format view of the memo document example.

MS Word adds some extra "stuff" to RTF markup tags and saves the entire document as a binary file, which is not readable as text. MS Word document format is also closed. It is not publicly available. Since Notepad cannot interpret RTF tags. It simply displays everything in the document without applying any formatting to the text. So human eyes can easily read all the code in the document. This illustrates an important notion about markup. A processor read the markup codes, interpret how they affect the text, and display the results. It is impossible to derive the ***structure*** of the document simply from the code. The author of the document can place words and formatting anywhere in the document and in any order. But it makes it difficult for human readers to interpret the markup code of a complex document. Processing a specific section of the formatted text will cause the structure information lost. It also makes it almost impossible for someone else to author a document of exactly the same type. [37]

The above document is not easily portable to other platforms or devices. Since no rules exist for how this document is structured. The document would certainly not be extensible beyond the markup already coded into it. In other words, ***without the ability to set rules for a document, it is impossible to create other types of reusable document structures from the original document***.

HTML is ***an electronic-publishing language, which supports and shares information on the internet***. It is superficial and describes how a web browser should arrange text, images and others on a page. It **concerns with appearances.** HTML was defined as ***an application of SGML. HTML documents*** are supposed to ***conform to a specific set of rules that identify exactly how a document should be put together.*** HTML is very ***simple.*** It makes web cruising so simple. The screen formatting that is built into HTML is likewise very simple. *"HTML's limited formatting options make web publishing even easier."* [38] It primarily indicates only the appearance of a document's element. *"HTML's flexible and powerful hypertext linking is* ***easy to set up****, but it has limitations that complicate large-scale implementations."* [38] HTML has significant barriers for applications beyond simple browsing, such as reuse, interchange, and automation**.** ***It lacks the necessary expressiveness. It is limited to a simple, small and fixed set of presentation-oriented tags. It lacks the absolute consistency***. *"Many markup languages have served quite well as document formatting tools for printing or for the Web. However, they do not perform as well at describing the data they contain or at providing contextual information for the data"* [39]. HTML ***provides nothing to denote the data within a document***, which cripples attempts to achieve reuse.

3.1.2. Generalized markup languages

"*Generalized markup languages* ***describe the structure*** *and* ***meaning of the text*** *in a document, but* ***it does not define how the text should be used***." [40] The language itself is not made for any specific application. It is generic enough to be used in many different applications. Documents described with a generalized markup language are usually portable to more applications than those which are described with a specific markup language. Document Composition Facility Generalized Markup Language (DCF GML, or GML) [41] was the precursor to Standard Generalized Markup Language (SGML[11]) that was adopted as a standard by the International Organization for Standardization[12] (ISO) in 1986. It provides two basic parts:

- ***The markup should describe the structure*** of a document (not its formatting or style characteristics).

[11] See www.w3c.org/MarkUp/SGML/ for details (Last visited December 29, 2002)
[12] See. http://www.iso.ch/welcome.html for details (Last visited December 29, 2002)

- ***The syntax of the markup should be strictly enforced***.

SGML is a metalanguage which prescribes the rules for creating a specific markup language such as HTML. "*SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the Web.*" [42] It is defined by ISO 8879-1996. [43]

The benefits of SGML are:

- *"Infinite possibilities for expressing information (infinite tag set),*
- *Write once, reuse many times,*
- *Future-proof, platform-proof,*
- *Validation for completeness and correctnes.," [44]*

SGML has some limitations for web delivery. ***SGML only standardizes structure.*** It is too general. There is no widely accepted standard stylesheet format for expressing SGML information. The primary problem is that it has **no mainstream browser support.** It has little or no vendor support.

These similarities exist for HTML; because ***HTML is an application of SGML. HTML was created by using SGML standard.*** Because of this, many of the details of SGML are carried through to HTML, but not all details. The biggest difference between SGML and HTML is that ***nothing in the SGML document indicates how the data should look.*** However, the markup identify the structure of the document. ***SGML is extensible.*** It means that an author can define a particular structure by defining the tags that fit that structure. However ***HTML is not extensible***.

3.2. Extensible Markup Language (XML)

XML is a markup language. It has a mechanism to identify structures in a document for documents containing structured information. It is general-purpose information exchange standard which is defined by the World Wide Web Consortium (W3C)[13] in 1996. It is approved as a standard**[14]**, XML specification[45],

[13] See http://www.w3c.org/ for details (Last visited December 29, 2002)
[14] See XML 1.0 http://www.w3c.org/TR/1998/REC-xml-19980210 for details (Last visited December 29, 2002)

in 1998. *XML is a set of rules for designing text formats for data. It produces files that are easy to generate and read.* W3C has called *XML* "*a common syntax for expressing structure in data*". [45] It is designed to improve the functionality of the web. It provides more flexible and adaptable information identification. A document (or a piece of data) can be separated into three parts: ***structure*****,** ***content*****,** and ***style*****.** ***Content, presentation format, and interface rules are not integrated in an XML statement.*** It is a feature of quality; not a deficiency. ***XML separates data*** and ***the presentation of data.*** The XML document can be displayed in a number of ways depending without changing the structure of the data. Content is defined by the user. All documents have some structure. XML documents contain character data and markup**. XML data is in text form.** The distinction between data and markup, ***style,*** is separated more in XML than it is in many other systems. It is portable across a number of environments and platforms. The original source document can be displayed on a computer monitor, within a cellular phone display or translated into voice. "*XML allows developers to easily describe and deliver rich, structured data from any application in a standard, consistent way. XML lets you exchange information across platforms, not to be confused with writing cross-platform applications.*" [46] Technology is needed to allow software systems to share information. *"Web usage is growing up by wireless phone, personel digital assistant (PDA), and vs. Web must be reconstructed because of the display, bandwidth, memory and central processing unit (CPU) constraint on these devices. For example, a product data is formatted by using XML, which is shown Figure 3.4. This product data can be presented by an XML processor in different environment, which is shown below Figure-3.5*[15]*."* [47]

```
<product>
        <id> 12345678-Q </id>
        <description> Thinkpad 2000D </description>
        <price> $999.99 </price>
</product>
```

Figure 3.4 : XML tagged text code example.

[15] Figure 3.4 and 3.5 is taken from Assist.Prof.Selim AKYOKUŞ's presentation in [47] .

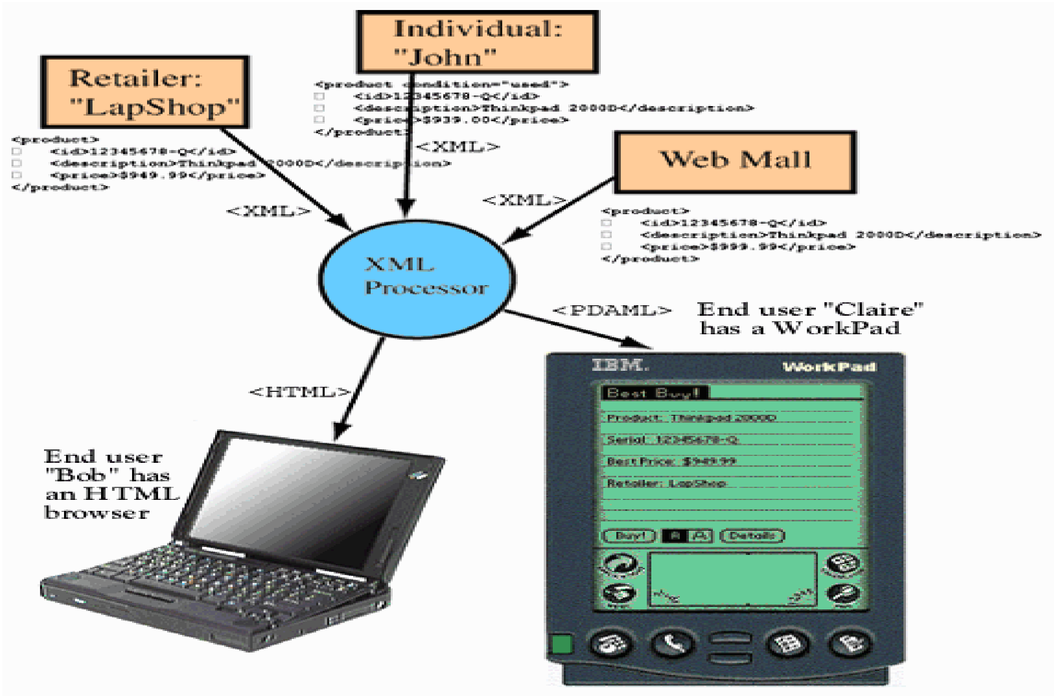


Figure 3.5 : XML processor in different environment.

SGML is ***father to HTML*** and ***brother to XML***. SGML has never reached widespread public usage because of its complexity. XML was invented to overcome SGML's complexity. XML = SGML with minor changes. Applications that can read SGML can also read XML, however the opposite is not true XML and HTML are related. HTML is used for creating web pages that can be viewed all around the world. They have similar characteristics. HTML is an application format which is based on SGML. HTML cannot be used to define new applications as easily as XML can. ***XML is a complementary format to HTML***. ***XML is a highly functional subset of SGML.*** But it has only a small part of its complexity. XML is a meta language like SGML. That means new markup languages can be defined. Markup tags built the structure of XML as in HTML. However, there is a very important difference between tags in HTML and tags in XML. ***XML tags have no predefined meaning*** unlike HTML tags**.** XML allows the creation of an indefinitely large number of new (custom-defined) or tags. Therefore it enables the generation of new specific markup languages for diverse fields such as mathematics, vector graphics, music and technical documentation. [48] The comparison of XML, SGML and HTLM can be easily seen at table 3.1

Table 3.1: Comparison Table of Three Basic Markup Languages

S.No	SGML	HTML	XML
1.	Extensible	Not extensible (Limited structure-limitations can be traced to its fixed set of tags)	Extensible (tags can be defined by the user)
2.	Flexible	Inflexible	Flexible
3.	Complex	Very simple (HTML makes Web cruising so simple)	Simple but a little complex than HTML, and simpler than SGML
4.	Learning and using is hard because of its complexitiy	Easy to learn and use	Easy to learn and use
5.	SGML provides the capability for any designed set of tags infinite possibilities for expressing information (infinite	HTML is a single set of tags	XML provides the capability for any designed set of tags infinite possibilities for expressing information (infinite tag set)
6.	work can be shared with others	limited work sharing with others by display capability	Work can be shared with others
7.	Write once, reuse many times	Limited reuse (Many organizations publish the same information in multiple forms) (provides nothing to denote the data within a document, which cripples attempts to achieve reuse)	Write once, reuse many times

Table 3.1: (Continued)

S.No	SGML	HTML	XML
8.	Validation for completeness and correctness	No validation for completeness and correctness	Validation for completeness and correctness
9.	Future-proof, platform-proof	Platform-proofed	Future-proof, platform-proof
10.	No mainstream browser support	Internet Explorer and Netscape (common and most used) browsers support	
11.	Links are multi-directional (supports one source and many destination)	Links are uni-directional (always one source and one destination)	Links are multi-directional (supports one source and many destination)
12.	Go back lists, etc. are maintained by links	Go back lists, etc. are maintained by the browser	Go back lists, etc. are maintained by links

XML is not just for Web pages. It can be used to store any kind of structured information. It simplifies and lowers the cost of data interchange and publishing in a Web environment. ***Both computer and humans can easily read XML syntax.*** *XML is the universal license-free, platform and vendor-independent, industry-wide and well-supported format for data on the Web.* XML allows documents to be written once and published in multiple media formats and devices. It also supports media-independent publishing.

XML is a family of technologies :

1. **D**ocument **T**ype **D**efinition (**DTD**)
2. XML Schema,
3. XML Namespaces,
4. **D**ocument **O**bject **M**odel (**DOM**),
5. Style Sheet Languages
   a. **C**ascading **S**tyle **S**heets (CSS),
   b. e**X**tensible **S**tylesheet **L**anguage (XSL),
6. eXtensible Linking Language (XLL),
   a. **Extensible Linking Language (XLink),**
   b. **XML Pointer Language (XPointer),**

3.2.1. Document Type Description (DTD)

"The function of the markup in an XML document is to describe its logical structure and storage layout and to associate attribute-value pairs with its logical structure. XML provides a mechanism, ***DTD, to define constraints on the logical structure and to support the use of predefined storage units".*** [49]

"The document type declaration must appear before the first element in the document" *[46].* DTD gives information about the rest of the document structure. XML documents can be associated with a DTD. ***DTD defines what markup tags*** (what names are used for the different types of element) ***is used in the document. It determines appearance order of the tags and so on***. This allows applications to validate import data for conformation to DTD. DTD is an authors' rule book to create new documents of the same type. It can be used as a base document. That is briefly to say, **DTD is XML's grammar**. [50]

"*A markup declaration is an element type declaration, an attribute list declaration, an entity declaration, or a notation declaration. An element type declaration constrains the element's content. An attribute-list declaration specifies the name, data type, and default value (if any) of each attribute associated with a given element type. Entity declarations define storage units. Notation declarations provide names for the notations, which identify by name the format of unparsed entities*" [50]

A DTD can be ***external*** or ***internal*** (embedded). An external DTD is defined outside the content of the document. It is prefered when a common DTD is used. An internal DTD is part of an XML document. A document can contain one or both types of DTDs. If a document contains both, ***the internal subset is processed first and takes precedence over any external subset.***

3.2.2. XML Schema

"*The purpose of a schema is* ***to define and describe a class of XML documents,*** *just like a DTD,* ***by using these constructs to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content, attributes and their values, entities and their contents and notations.*** *Schema constructs may also provide for the specification of implicit information such as default values. Schemas document their own meaning, usage, and function.*" [51] XML Schema was originally proposed by Microsoft, but became an official W3C recommendation[16] [51].

XML DTDs were intended originally for document management. XML schema is an alternative to DTD. An XML schema describes the structure of an XML document. XML schema language is also referred to as XML Schema Definition[17] (XSD). W3C XSD Language is an XML language for describing and constraining the content of XML documents. It is more expressive than XML

[16] See http://www.w3.org/XML/Schema for details (Last visited December 29, 2002)
[17] See http://www.w3schools.com/schema/schema_intro.asp for details (Last visited December 29, 2002)

DTDs. It is used by a wide variety of applications. It is self-describing, and be optimized for interoperability;

3.2.3. XML Namespaces

The term namespace means ***a set of names in which no duplicates exist.*** *"An XML namespace is a collection of names, that is identified by Universal Resource Identifier (URI), which are used in XML documents as element types and attribute names"* [52] Namespaces in XML is a methodology for creating universally unique names. It is a specification (collection of names) that describes how a URI is associated with every single tag and attribute, types and names, in an XML document. XML namespaces are based on the use of qualifed names. It consists of a namespace prefix and selects a namespace. The combination of the univesally managed URI namespace and the local name produces names that guarantee universal uniqueness.

A namespace declarations must be located after XML declaration (if any), and before the DTD (if any). It can be included directly within an element that is part of the namespace. The scope of namespace prefixes is global to the whole document, including the DTD. "***A namespace prefix may not be declared more than once; it must be unique within an XML document*** " [46]**.** The namespace prefix ***xml*** is reserved. No other prefix beginning with this three-letter and any case combination is allowed.

*"**Writing schemas (or DTDs) from scratch is hard,** so it will be beneficial if parts from existing, well-designed schemas can be re-used. There are many occasions when an XML document needs to use markups defined in multiple schemas, which may have been developed independently. Both considerations require that document constructs (tags and attributes) have universally unique names. XML namespaces provides the mechansim to accomplish this*". [46]

Generally, an application will try to discern the associated XML tag. XML does not ensure data compatibility with a given application. **Standardization of XML tag elements and attributes can be provided by namespace registry.** Namespace registry helps interoperability standardization by preventing identical custom-named tags in XML documents, standardization of XML tag elements and attributes.

3.2.4. Document Object Model (DOM)

DOM, the W3C DOM[18] [53], is a standard object application-programming interface that gives developers programmatic control of XML document (manipulating XML, and HTML files) content, structure, formats**.** DOM is not an object model like the Component Object Model (COM). The COM, like CORBA, is a language-independent way to specify interfaces and objects. ***DOM manages HTML and XML documents by using a set of interfaces and objects.***

"*XML on its own is really this stream of internationalized characters that follows certain rules. But you cannot actually do anything with this stream of characters with pointy brackets and question marks in it. But, for example, if you have used XML to define items in a catalog, you need some way of getting ordering information out of the catalog and inside an order form, inside that commerce application That's the DOM.***"** [46] XML-related work will be represented in terms of tree and graph abstractions by hiding details. In particular, ***DOM provides a tree-based application programming interface*** to XML**.** It provides methods for traversing the tree. DOM protocol converts an XML document into a collection of objects, that is represented in tree in a program. Then the object model is manipulated in any way. Any part of the data can be visited at any time, that is call "random access", then modify the data, remove it, or insert new data.

W3C DOM consists of different levels (DOM Level 1, 2, 3). Each new level is based on the previous and extends the API to make it more usable. These levels together are usually referred to as DOM Specification.

3.2.5. Style Sheet Languages

"*One of the most important aspects of **XML** is that it **separates style from data.** This enables different users to define their own views of how they want the data to be presented."* [46] Information presentation of an XML documents is stored separately in a stylesheet document. Style sheets offer precise control over the presentation of web pages. ***A stylesheet is a set of stylistic rules that describe how web documents are presented to users.*** Using stylesheets can specify like size, color, spacing and placement of text, images on the page, and

[18] See http://www.w3.org/DOM/ for details (Last visited December 29, 2002)

more. ***Stylesheets tell the processor how to display the document.*** Two evolving stylesheets languages to use with XML are;

1. Cascading Style Sheets (**CSS**) language
2. Extensible Stylesheet Language (**XSL**).

3.2.5.1. Cascading Style Sheets (CSS)

Cascade is a method of defining importance of individual styling rules. It allowes conflicting rules to be sorted out and such rules can be applied to the same selector. ***Style Sheets are templates.*** It is similar to templates in desktop publishing applications.

Cascading Style Sheets[19] were introduced in late 1996. It enables much more sophisticated page design (typography and layout) than web developers had been used to. They helped manage the complex tasks of developing and maintaining web sites, and keeping them up to date. In the case with the W3C DOM, the CSS specification also consists of different levels [54][55]

3.2.5.2. eXtensible Style Language (XSL)

XSL is a style language for XML. It allows to specify rules that indicate how to transform an XML document. "*XSL stems from two standards, **D**ocument **S**tyle **S**emantics and **S**pecification **L**anguage (**DSSSL**)* [56] *and **C**ascading **S**tyle **S**heets (**CSS**) language. DSSSL is an ISO standard for specifying document transformation and formatting in a platform and vendor-neutral manner*" [46] DSSSL is used to specify the presentation of documents markup by using SGML. DSSSL consists of two main components: **transformation language** and **style language**.

"The transformation language is used to specify structural transformations on SGML source files. It can also be used to specify the merging of two or more documents, the generation of indexes and tables of contents, and other operations." [46] Style language provides a standardized, powerful language for describing the format of SGML and XML documents. The presentation of data is

[19] Seehttp://www.w3.org/Style/CSS/ for details (Last visited December 29, 2002)

handled separately with either **CSS** or **XSL**[20]**.** [57] Both of them are also W3C standards. "*The fact that W3C has started developing XSL in addition to CSS has some confusion.* ***Why develop a second style sheet language when implementers haven't even finished the first one?*** *The answer can be found in the table 3.2.*" [46]

Table 3.2[21]: CSS and XSL usage with HTML and XML [46]

	CSS	**XSL**
Can be used with HTML?	Yes	No
Can be used with XML?	Yes	Yes
Transformation Language	No	Yes
Syntax	CSS	XSL

CSS styles documents *but* ***XSL transforms documents.*** It is particularly useful for high quality print production. XSL defines a set of formatting objects and XML namespace. It describes the desired format of a document. Structure of an XSL document can be entirely different from the source of XML document. XSL Transformation Language (XSL-T) [58] is used for transformation from the source to the target XML. It also used for simple XML to XML transformation. This transformation could be a presentation format (e.g., HTML or PDF), or to an alternate representation of the content (e.g. an XML document with a different DTD). As a result, content of a document can be managed independently of from its presentation. An alternative view of that content can be produced by using different XSL stylesheets.

XSL is the advanced language for expressing style sheets. An XSL processor uses XSL stylesheet to parse an XML source document and construct usable output for presentation of the stored data. Figure 3.6 [46] shows How an HTML output can be provided from an XML document and XSL. In theory, no limitations on the output is set such as sound files, RTF, raw text, etc. In theory, In the future, web browsers will support XSL directly, but currently there is no such support.

---

20 See http://www.w3.org/Style/XSL/ for details (Last visited December 29, 2002)

21 Table 3.2 is taken from [46] , pp.20

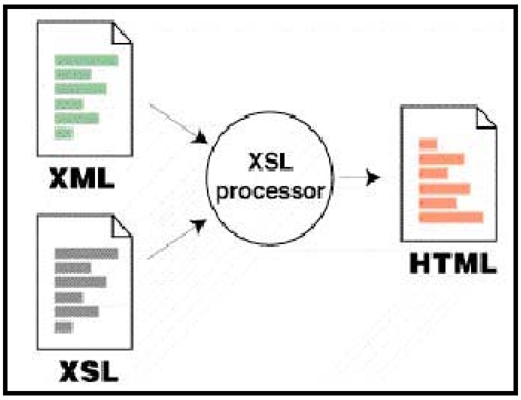


Figure 3.6[22]: XML, XSL stylesheet and XSL processor relationship to construct HTML output [46]

3.2.6. eXtensible Linking Language (XLL)

XLL is a broad term for XML hyperlinking (linking and addressing). It is based on the ISO/IEC 10744-1992 [59] standard for hypermedia. It is an extension of SGML. It defines inline and out-of-line link structures and some semantic features including traversal control and presentation of objects. XML linking and pointing/addressing mechanisms formerly named as "Extensible Linking Language[23] " [60] at March 1998. It were provisionally renamed ***Xlink***. Then, XML linking and addressing mechanisms are described in two W3C specifications:

1. XML Linking Language (XLink),
2. XPointer.

A link can be defined as an explicit relationship between two or more data objects or portions of data objects. Links can be managed externally. They associate metadata with a link, and express links that reside in a location separate from the linked resources. *XML syntax is used to create link structures.* XML *allows* not only simple one-way hypertext links, but also ***multi-directional links.*** Different destinations can be reachable from a single link. Content of a document is directly placed and being viewed without user intervention by XLL. XLL also provides for replacing content inline with updated content from another document.[61]

---

[22] It is taken from [46], pp.24

[23] See http://www.w3.org/XML/Linking for details (Last visited December 29, 2002)

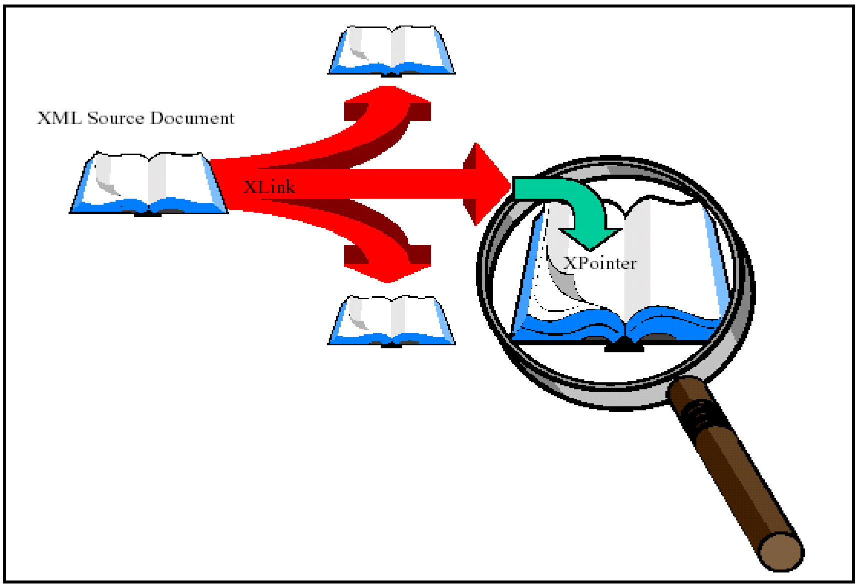


Figure 3.7 : XML document, the XLink and Xpointer relationship [46]

Figure 3.7[24]. shows how Xlink and Xpointer work together in XLL. Xlink is used to reach a document and Xpointer points the location out within that document. [46] ***Xpointer operates on the tree representation of XML documents.*** It supports specification of locations by nodes and node lists. Xpointer builds upon XPath addressing into the internal structures of XML documents. *XPointer* is expected to be used with XLink. It is defined in W3C Working Draft [62].

XPointer useful for finding relative locations. It provides better locations specifications than HTML. It points to specific places *inside* of documents (***Pointers without anchors*****).** It makes locations human-readable and writable, rather than mere hash.

3.3. XML Compression

XML provides a universal way of exchanging data across system and vendor boundaries. XML uses tags to identify data items. These tags allow users to search, sort, identify and extract data as desired. XML format makes the use and interchange of data easier and more user configurable. But XML substantially increases the size of these files over the size when the same data is represented

[24] It is taken from [46], pg.25

in its raw format. ***XML is verbose*** like other textual markup language***.*** There is a lot of **redundant data** in an XML document. XML documents are a prime candidate for compression. XML's verbosity is not its most significant "flaw". Although, the **simplicity**, **generality**, **versatility** and **exchangeability** of XML make it worthwhile. Large XML files consume bandwidth and large amounts of storage (both the client and server machines).

XML is still an evolving technology. "***XML's future could depend on efficiency.*** *It is likely that XML will continue to have strong support over the next decade in terms of tools, standards, and users*" [63]. "*The benefits of compression rate and compression time are* ***minimising storage space requirements, transmission bandwidth requirements and main memory requirements for processing."*** [64] Some compression formats of XML may be faster to parse making, reading and writing of the compressed files faster than non-compressed XML.

The right choice of compression tools and libraries for a particular application depends on the characteristics of the data and application in question: ***streaming versus file; expected patterns and regularities in the data; relative importance of CPU usage, memory usage, channel demands and storage requirements; and other factors.***

XML is stored in plain text files. Common text compressors can be used. Plain-text XML compression is easy, fairly fast, and relies on existing, robust compressors. Because XML is not yet mature. Markup documents (mostly markup) tends to be more redundant and compressible than text. Documents which is mixing text and structure tend to fall in between these extremes. This distinction between textual and structural data is a recurring theme. Off-line compression is undesirable for XML because it forces a long wait before document parsing and processing can begin.

Just as choosing the right algorithm can often create orders of magnitude improvements over even heavily optimized wrong algorithms. Choosing the right data representation is often even more important than compression methods. A lot of problems genuinely require significant memory, bandwidth, storage and CPU resources. In many of those cases compression techniques can help ease, or shift, those burdens.

Many-megabyte XML documents are often impractical to spend the memory, disk space, and CPU overhead to manipulate such huge documents. It would be nice if the entire multi-megabyte source could not be taken, but only parsed blocks, read serially, and apply similar techniques.

CHAPTER 4

XML AND XML FORMATTED MESSAGES

4.1. Overview of XML Message Text Formats

Digitisation of battlespace is guaranteed to deliver operational benefits. As Blair says :

> *Paradoxically, the capacity of modern communications and information systems may constrain the commander by overwhelming him with the volume of data. The growth of information is attributed to the increasing complexity of modern war. The quantity of information used to make battlefield decisions is very small; that operation orders should be very short; that very few staff are required to produce those orders; that there is significant advantage in making and disseminating decisions much faster than at present; Recently, a concept known as **"Information-Centric Warfare"** has made claims of being able to speed up command decision making and improve tactical command and control. The commander's greater* comprehension of events is an opportunity. The task of leading an information-rich formation is a challenge. The possibility of information transparency is a threat. [65]

Interoperability of military information systems is enhanced through the use of common information objects in the Army. One of the most widely used objects is ***Message Text Format (MTF)***. Message Text Format is human readable and character/text-based. MTF is one of the primary means for exchanging information within the Turkish Army. MTF is a military proprietary message standard for sharing structured, well documented collection of Information Exchange Requirement(IER)s. It is platform independent, man and machine-readable. It also supports full spectrum of military operations (intelligence, air, surface, sub-surface, fire-support, logistics, medical, etc.). The aim of using formatted messages in Command and Control (C2) systems is to establish a common standard of data interchange. Using a standard message format improves accuracy and consistency in the exchange of information. It also reduces time and effort required for that is drafting, reading and processing of messages.

Electronic data interchange is an appropriate area of XML development. XML is a metalanguage that provides a universal format for data. It provides a common platform for the exchange of information that is recognizable by both humans and machines. Developers can easily describe and deliver rich, structured data from any application in a standard, consistent way using XML. It forms a bridge between structured data and the unstructured real world. It does fill a gap in the existing web architecture. Its flexibility will enable integration with existing technologies as a data exchange format.

XML is an initiative to modernize military information standards through commercial technologies. Turkish Land Forces recognizes the potential of XML to enhance interoperability and information exchange among C4I systems. XML will become the tool of choice for exchanging information between Turkish Land Forces, other military units (Navy, Airforce) and governmental organisations. TLF will try to adapt XML to meet military information requirements. The simplicity, low development costs, and flexibility will make it attractive to in today's environment. The concept of building and interconnecting military systems around the information that they manage and exchange is a vision for the future of **T**urkish **L**and **F**orces **C**ommand **C**ontrol and **I**nformation **S**ystems Projects (**TLF C2IS**). These projects require extensive and constant information exchange between them, such as Command Posts of the Turkish Armed Forces and other governmental organizations. In military applications, XML-MTF endeavor is likely to be successful. While Turkish Land Forces expects improvement in interoperability. Data Management Group should focus on data content, rather than format. Systems will have to be developed which support XML-MTF standards. **TLF C2IS** will have defined ***XML-Message Text Format (XML-MTF) as a standard for Turkish Land Forces in near future.*** Current MTF system in Turkish Army, and NATO, uses slash characters as delimiters to parse messages into applications. A simple formatted message example is shown in Figure 4.1. The process is not user-friendly and expensive government-developed software is required to process the data. XML is an artificial language for C4I systems.

NATO FORMETS provides the rules, constructions and vocabulary for standardized Character-Oriented MTFs that can be used in both manual and computer-assisted operational environments. In the terminology of the standards, messages are made up of Fields, Sets (groups of fields), and Segments (groups of sets), as illustrated in the Figure 4.1. MEFORS is the messages, that are sometimes called reports, used by Turkish Army (General Staff, Land, Navy, Air Force) which uses NATO Allied Data Publication Number 3 (ADatP-3) rules.

The following fragment that is shown in Figure 4.1[25] is from a Tactical Report, or TACREP, which contains “Alarm Announcement” information in ADatP-3 MTF format:

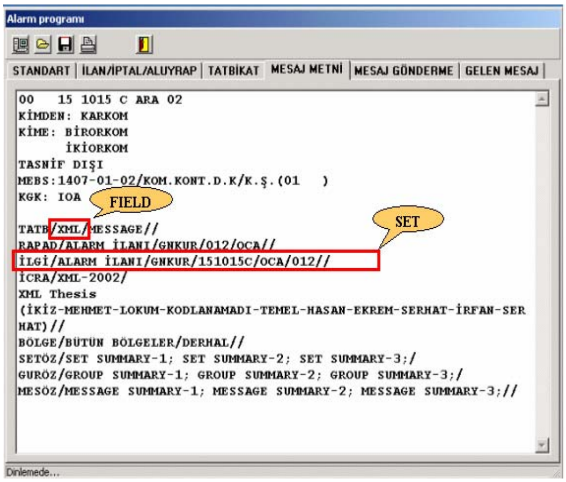


Figure 4.1 : Tactical report which contains “Alarm Announcement” ınformation in ADatP-3 format

The given example in Figure 4.1 that can be rewritten by XML is shown in Figure 4.2 :

[25] It is taken from Turkish Land Forces Alarm Announcement Reporting program.

```
<?xml version="1.0" encoding="windows-1254" ?>
<!DOCTYPE MEMO SYSTEM "alarm_ilan_iptal.dtd">
<!-- This is ALARM_ILAN_IPTAL.xml example with XSL for thesis -->

<ALARM_ILAN_IPTAL>
 <BASLIK_BILGILERI>
   <TARIH_SAAT_GURUBU>
        <GUN> 15 </GUN>
        <AY> ARALIK </AY>
        <YIL> 2002 </YIL>
        <SAAT> 10:15 </SAAT>
   </TARIH_SAAT_GURUBU>
   <KIMDEN> KARKOM </KIMDEN>
   <KIME> BİRORKOM </KIME>
   <KIME> İKİORKOM </KIME>
   <GIZLILIK_DERECESI> TASNİF DIŞI          </GIZLILIK_DERECESI>
   <EVRAK_NO> MEBS:1407-01-02/KOM.KONT.D.K/K.Ş.(01)
</EVRAK_NO>
   <KOD_NO> KGK:10A </KOD_NO>
 </BASLIK_BILGILERI>
 <MESAJ_METIN>
   <HAREKAT_TATBIKAT>
      <HRK_TATBIKAT> TATBİKAT </HRK_TATBIKAT>
      <ADI> XML </ADI>
      <ILAVE_TANITICI> MESSAGE </ILAVE_TANITICI>
   </HAREKAT_TATBIKAT>
   <RAPOR_ADI>
     <MESAJ_TIPI> ALARM İLANI </MESAJ_TIPI>
     <GONDEREN_MAKAM> GNKUR </GONDEREN_MAKAM>
     <MESAJ_SIRA_NUMARASI> 012 </MESAJ_SIRA_NUMARASI>
     <AY> OCAK </AY>
   </RAPOR_ADI>
   <ILGI>
     <MESAJ_TIPI> ALARM İLANI </MESAJ_TIPI>
     <GONDEREN_MAKAM> GNKUR </GONDEREN_MAKAM>
     <TSG>
        <TARIH> 15.10.2002 </TARIH>
        <SAAT> 10:15 </SAAT>
        <GRUP> C </GRUP>
     </TSG>
     <AY> OCAK </AY>
     <MESAJ_SIRA_NUMARASI> 012 </MESAJ_SIRA_NUMARASI>
   </ILGI>
   <ICRA_KODU>
     <KOD> XML-2002 </KOD>
```

Figure 4.2 : An XML formatted “Alarm İlan-İptal” message example.

```
        <ALARM_RUMUZU> XML Thesis </ALARM_RUMUZU>
        <RUMUZ_KODLAMASI> (İKİZ-MEHMET-LOKUM-KODLANMADI-
TEMEL-HASAN-EKREM-SERHAT-İRFAN-SERHAT)
</RUMUZ_KODLAMASI>
     </ICRA_KODU>
     <BOLGE>
       <KAPSADIGI_BOLGE> BÜTÜN BÖLGELER </KAPSADIGI_BOLGE>
       <IVEDILIK> DERHAL </IVEDILIK>
     </BOLGE>
     <SERBEST_METIN_SETLERI>
        <SET_OZETI> SET SUMMARY-1; SET SUMMARY-2; SET
SUMMARY-3; </SET_OZETI>
        <GRUP_OZETI> GRUP  SUMMARY-1; GRUP SUMMARY-2; GRUP
SUMMARY-3; </GRUP_OZETI>
        <MESAJ_OZETI> MESSAGE SUMMARY-1; MESSAGE SUMMARY-
2; MESSAGE SUMMARY-3;  </MESAJ_OZETI>
     </SERBEST_METIN_SETLERI>
  </MESAJ_METIN>
</ALARM_ILAN_IPTAL>
```

Figure 4.2 : (Continued)

XML file of Alarm Announcement /Alarm İlan-İptal" document which is given in Figure 4.2 might appear in the browser like in Figure 4.3 :

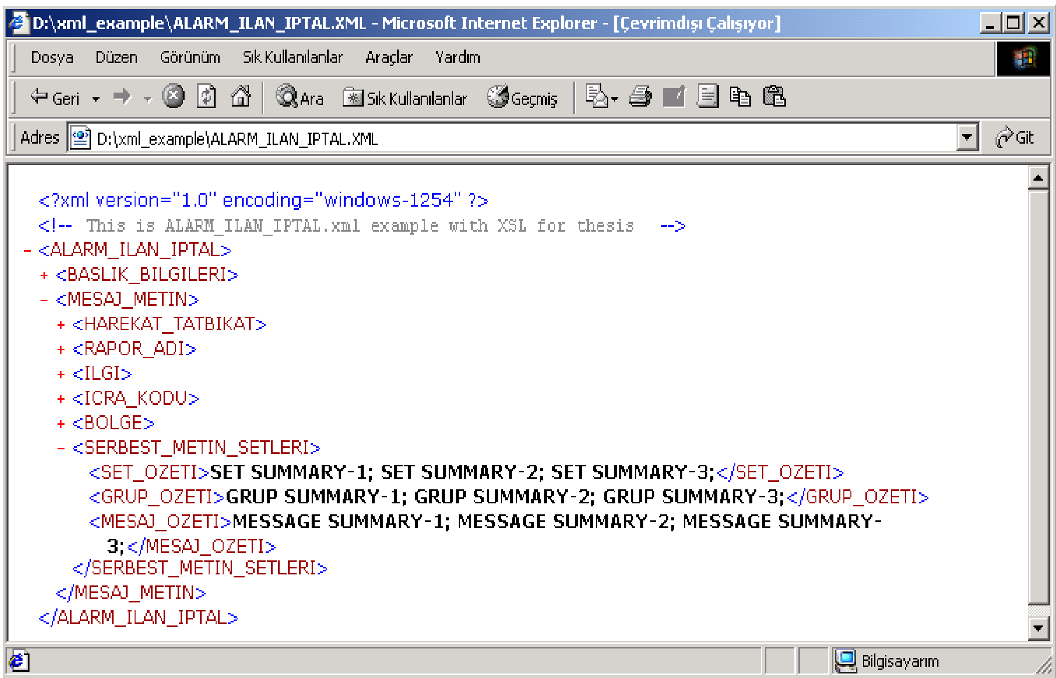


Figure 4.3 : An XML formatted "Alarm İlan-İptal Message" example browser view

Both human readibility and fast machine processibility of XML is critical in an environment where the decision maker has an extremely short amount of time to absorb information. XML is particularly well suited for web environments. For users with legacy systems that do not recognize XML, a short program can be written that translates XML-MTFs into the format recognized by legacy systems. Shells for message types can be defined with XML-MTF, which are stored in a database.

Allied **D**ata **P**ublication **N**umber **3**[26] ***(ADatP-3)*** [24] defines an artificial language which is suitable for the exchange of defense information. The system comprises the rules governing representation of agreed conceptual definitions and arrangement of these representations within predetermined formats.

Almost all forms of formatted messages have a heading, a text and an ending. The heading and ending parts of a message is dependent upon protocols for the communications system that is being used. Procedural guidance for these parts is specified in the appropriate communication publications. It is not affected by message text formatting rules**.** Text composition of a message is divided into three portions (***Introductory/Header*** Text**,** ***Main Message/Body*** Text**,** ***Closing*** Text) to account for additional communications, security or other requirements that must be met. An example is shown in Figure 4.4.

[26] This document is not publicly available. It is classified as “Mission Secret”. This part is summarized from Allied Data Publication Number 3 (ADatP-3)

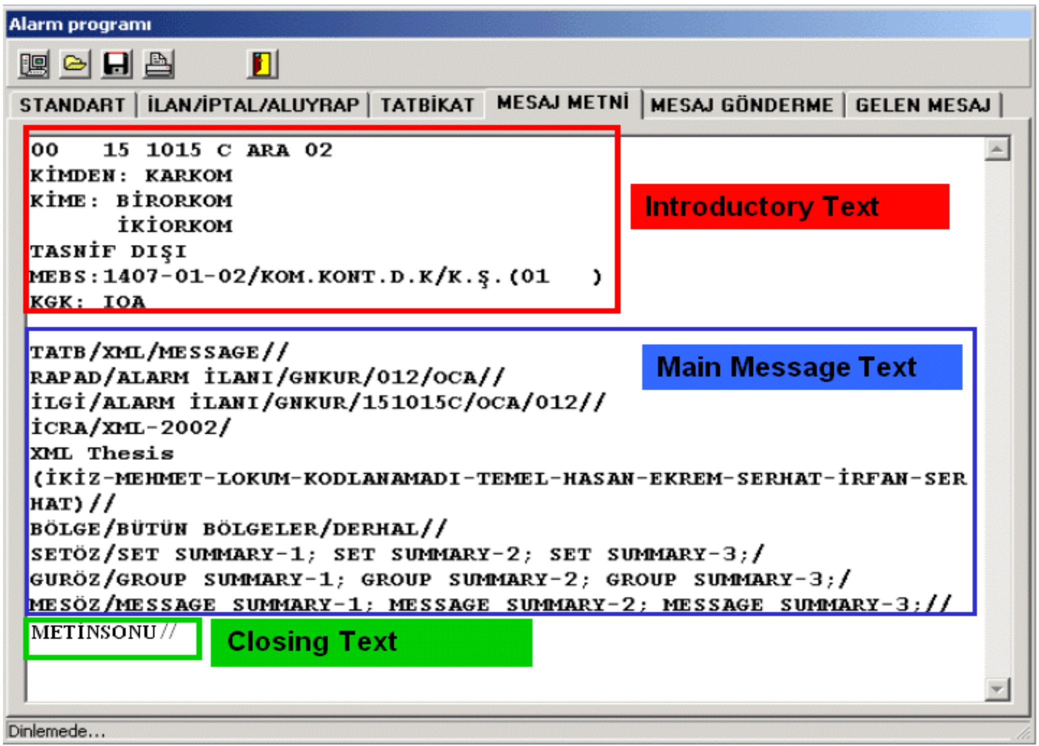


Figure 4.4 : The Alarm announcement tactical report message text format

The introductory text is generally reserved for information required by procedures established for the communications system in use. The ***main message text*** contains all the information concerning the subject that is addressed by the formatted message. It is structured in accordance with a message text formatting rules. The ***closing text*** provides information that is related to, but is not part of, the message text.

4.2. Military Message Standards Compared with XML

The emergence of the XML has created significant interest in the software industry. Use of XML-enabled software in different organisations simplifies electronic information exchange, thereby supporting interoperability between the organisations. Ease of integration and interoperability are desired qualities in most computer systems and certainly in military C4I systems. An XML enabled C4I system would most likely be easier to integrate with XML-enabled Commercial-Off-The-Shelf (COTS) products, than a C4I system using a proprietary format. Indeed, C4I systems could probably be constructed by XML-enabled COTS products alone. To facilitate interoperability between C4I systems in different organisations (especially between military organisations and non-governmental organisation), XML based messages could be used in the future to exchange information between the systems. [66]

A number of standards have been developed to facilitate unambiguous, electronic exchange of military messages e.g. ADatP-3, MEFORS, USMTF, ADFORMS (information is given in section 2.2.2), and VMF (information is given in section 2.2.1.3.). A catalogue of formatted messages are defined, which covers everything from logistics, to mission planning and intelligence reports.

XML only defines the general rules to structure information. It does not provide any catalogue of messages, nor does it provide any means for expressing constraints on the message contents.

The standards whose name is given above, all contain general rules describing how to represent fields, sets, and segments of a message. They also specify how to group fields to make up a particular set, and how to group sets to make up a specific formatted message. To allow unambiguous interpretation, the meaning of formatted messages is also described. They also contain constraint specifications. As the Figure 4.5[27] illustrates, XML specification is minimal compared with ADatP-3.

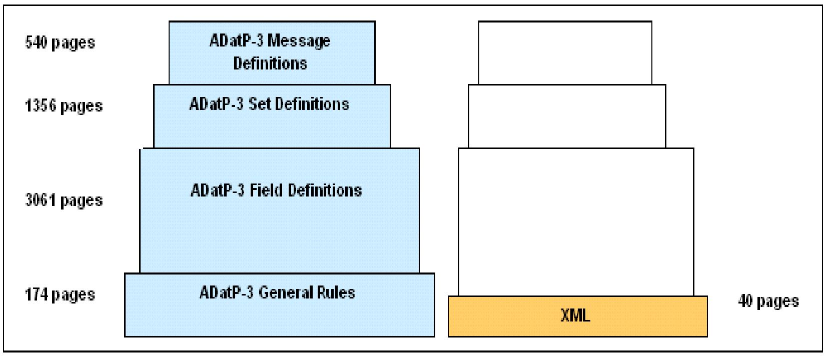


Figure 4.5 : The comparison of AdatP-3 and XML specification rules book total page number[28]

---

[27] Figure 4.5 and Figure 4.6 are taken from [67] pp.5,6 with minor changes.

[28] XML compared with Adatp-3, the numbers how many pages contained in each part of the printed standards (AdatP-3 Baseline 1.0 and RFC XML 19980210, respectively).

"*To continue the analogy to natural languages, one could say that XML is to artificial languages, what the Latin alphabet and the convention of writing left to right, separated by spaces and finishing with punctuation marks, is to Western languages.*" [67] It means that a person can detect words and sentences written in another language, but message cannot be interpreted correctly, if he/she does not know that language. Similarly, computer systems cannot interpret the messages correctly without special knowledge about the structure of messages written in XML. This means that XML in itself cannot be used as a replacement for the existing military message standards. XML can be used with those standards which maintain the catalogue of messages, as illustrated in the Figure 4.6.

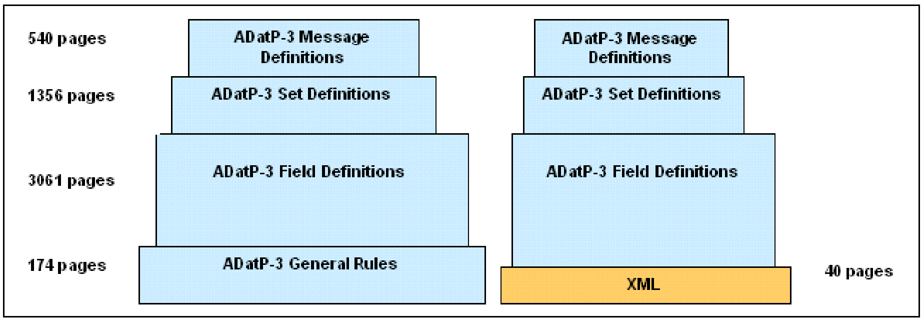


Figure 4.6 : AdatP-3 to XML mapping framework

XML can provide a common framework on top of which more complex standards can be constructed. XML versions of the military standards are relatively easy to produce. In the future, ADatP-3 to XML mapping will have been become part of the ADatP-3 standard. Current NATO and TLF C4I systems are based on current versions of ADatP-3. NATO and TLF's systems will be redesigned to handle XML messages that are formatted to the military standards. XML format makes the use and interchange of data easier and more user configurable. XML substantially increases the size of these files when the same data is represented in its raw format. Compared with the ADatP-3 version, XML representation requires 2-10 times as many characters in order to represent the same amount of information which is illustrated in example given in Figure 4.7. Like any textual markup language, ***XML is verbose.*** Therefore, there will be a need for software products capable of handling AdatP-3 and XML messages.

FORMATED MESSAGE

……
HAVAHRK/020200Z/6/TU/F16/İN:401/
/YÜKSEKLİK:12000FT//
……

XML-MTF MESSAGE

```
<hava_harekatı_veri>
     <tarih-saat> 020200Z </ tarih-saat >
     <miktar> 6 </miktar>
     <ülke> TU </ülke>
     <ucak_tipi> F16 </ucak_tipi>
     <iz_no> 401 </iz_no>
     <yükseklik> 12000 ft </yükseklik>
...
</ hava_harekatı_veri >
```

Figure 4.7 : Formatted message and XML-MTF example

There is a lot of redundant data in an XML document including white spaces, elements and attribute names, that are shown in Figure 4.7. XML documents are therefore a prime candidate for compression. Data compression is widely used in a variety of programming contexts. The characteristics of the data is important to choose the right compression tools and libraries for a particular application. Some experimental test results, which compares AdatP-3 message file and XML message file sizes (in kilobits) and their compressed forms (in kilobits) with commonly available compression programs, are shown in Table 4.1. File-A, the first row of the table, is a Tactical Report which contains "Alarm Announcement" information. The first column represents the size of AdatP-3 formatted MTF, the second column of the table, is an XML formatted version of the same message. File-B is inflated by appending 4 of the same type of messages together. (4 dinstinct Alarm Announcement Tactical Report Messages). File-C is inflated by using large text in SETÖZ GURÖZ and MESÖZ in Alarm Announcement Tactical Report Message. File-D is inflated by appending 30 messages of the same type together (30 dinstinct Alarm Announcement Tactical Report Messages that is used large text in SETÖZ, GURÖZ and MESÖZ).

Table 4.1 : Comparison of Adatp-3 and XML Formatted Files Sizes and Compressed File Sizes That is Compressed by Common Compression Program

Compression Type ==>	Original File Size		7Z		BH		BZ		BZ2		BZIP2		CAB		GZIP		Jar		LHA		RAR		ZIP		ZLIB	
Files	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML	P-3	XML
File A.txt	1	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
File B.txt	4	13	1	2	1	1	1	2	1	2	1	2	1	1	1	1	1	2	1	1	1	1	1	1	1	1
File C.txt	35	38	5	6	5	6	6	7	6	7	6	7	5	6	5	6	5	6	8	9	5	5	5	6	5	6
File D.txt (30xFile A)	35	215	3	2	3	3	4	2	4	2	4	2	3	2	3	3	3	3	5	3	3	2	3	3	3	3

* **AdatP-3 File Size** : P-3
* **XML File Size** : XML

These experimental results show that, if the file size is small, there is not a big difference between AdatP-3 formated file and XML formated file sizes. Applying compression can be taken into consideration if text part is much bigger than tags and the file size is large. Current army wireless technology has limited transmission bandwidth. Standard army radios have been improved to transmit voice and data, as in TASMUS. The transmission of graphics and messages when their sizes reach into the hundreds of megabytes form a significant bottleneck. Many-megabyte XML documents are often impractical to spend the memory, disk space, and CPU overhead to manipulate such huge documents. It would be nice if the entire multi-megabyte source could not be taken. Just as choosing the right algorithm can often create orders of magnitude improvements over even heavily optimized algorithms when it is chosen wrongly. Choosing the right compression methods is sometimes even less important than the right data representation. Significant memory, bandwidth, storage and CPU resources are wasted and this creates a lot of problems. In many of those cases compression techniques can help ease, or shift, those burdens.

At land tactical theatre, military units, like infantry and tank units, genarally do not use real time data (seconds or minutes can be negligible). Because of negligibility of small time intervals, compression time is not as important as limited transmission bandwidth.

CHAPTER 5

A NETWORK SIMULATOR
FOR BRIGADE LEVEL MESSAGING SYSTEM

5.1. Conceptual Overview

In the future, military operations will be more complex and unpredictable than those of the past. The commander's comprehension of events will grow up. To meet the challenge and counter the threat, ***"brave thinking will be required"*** in order to seize the opportunity. Technology will never substitute for good generalship and human qualities and moral issues will be increasingly decisive in this environment. "*Commanders will need to be able to focus on critical issues in a wealth of data. Roles and responsibilities should change to emphasise leadership over management. Creativity will be rewarded and predictability heavily punished. The next major war will teach some hard lessons. Being ready to learn fast is more important than having the answers in advance."* [68] Developers made decisions about how to define, organize, manipulate, store and transport information based on what was optimal for the system under development. There is much more information that can be fed around the today's systems than a decision-maker can assimilate.

Digitization must be able to decrease commenders' decision-making time by optimizing the flow of information. Using and timely delivery of information in tactical theatre needs information management, retrieval, delivery technologies and processes. This is especially important for decision makers who have to respond to dynamic situations. In order to achieve automated information exchange within TLF, a common specification of the information is required. Legacy systems seems to be working fine. Current information exchange requirements (IERs) change over time, and for that reason there was a need to design a flexible generic model that could adapt over time to changing information needs as well as serve as a basis, if ***TLF*** desired, for new systems. To realize a new vision of messaging system for TLF C2IS projects, XML will be considered as an initiative to modernize military information standards, that provides a universal format for data and provides a common platform for exchange of information, through commercial technologies. XML will become the tool of choice for exchanging information among Turkish Land Forces sub-units, other military units and govermental organisations. Past, today and future (if accepted) MTF implementation strategy is shown in Figure 5.1 apparently.

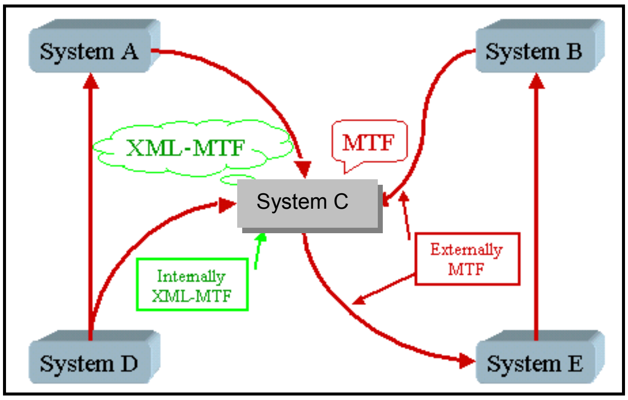


Figure 5.1 : Data exchange with MTF externally and with XML-MTF internally

Messaging is vital for all millitary organizations. No messaging system can be thought of as a stand-alone. Organizations need to analyze and develop the **optimal network structures** with consideration of their hierarchical structures. Their needs are to be analyzed as well. The structure of messages and the technology of the network needs to be considered and then planned. For this reason the general appearance of the structure of a brigade has been formalized. Then, a picture of the lines of communications and messages coming in and going out of a brigade has been drawn.

Different messaging technologies are analyzed, which is appropriate for TLF interoperability itself and with NATO, which are explained in chapter-2 and 4, so that it can be applied them to the new messaging system of the Brigade. As a result, there will be connection(s) between units within the Brigade. After generating a final picture of messaging system, the resulting network and messages are analyzed by using simulation tool. The purpose of this chapter is to design and develop a messaging structure for a simulated Brigade level military organization. The structure of a Brigade is analyzed and different messaging and communication combinations for different levels of hierarchical structure are investigated for the usage of *XML-MTF as a standard for Turkish Land Forces messaging system in near future.*

5.2. Conceptual Model

5.2.1. Requirements

System will be used for messaging of tactical level military organisation. ***XML is verbose.*** It substantially increases the size of files when the same data is represented in its raw format. XML documents are a prime candidate for compression. Data compression can be used in a variety of programming contexts. Current army wireless technology has limited transmission bandwidth. How much bandwidth of a link is used with ADatP-3 formatted version of a message and XML representation of the same file, and their compressed form? Critical waiting times of the messages in queue must be determined or can be compared with each other.

5.2.2. Entities and Objects

Units, communication links and messages are main entities of the tool for simulation.

Unit types are;

- Number of personel, weapons, and tasks are changed with their service type for all level military organisation.
- Platoon : is the lowest level military organisation,

- Company : is the lowest level main manoeuvring unit of a military organisation. It is the superior unit of a platoon,
- Battalion : is the superior unit of a company. It is the lowest level military organisation which acts independently,
- Brigade : is the main operational unit which fully supports itself for an operation. It is the superior unit of a battalion, but it has service support units (generally they are company level units), and fire support units (they can be battalion or company level units)

Units are defined by attributes;

- unit number : is a unique number which identifies each military unit,
- unit name : each unit has a short and long name,
- type of unit : identifies the type of military unit such as battalion, company, etc.,
- unit class : is the main functional area of unit or briefly military service type such as border defence unit, tank, infantry, etc.,
- superior unit: the unit of which this unit is a part.

A communication link object is modeled with;

- its bandwidth in bits per second,
- the two military units at two end points of the link,
- necessary explanation text about the line.

Messages are defined with the following attributes;

- message name: each message has a short and long name.
- message size (in bits) : is the total of message header size and message body size,
- message header size (in bits) (that is shown in Figure 4.4),
- message body size (in bits) (that is shown in Figure 4.4),
- minimum, average and maximum size (in bits) of that message,

- XML conversion factors: are used to estimate the size of the message when it is coded in XML. For this purpose message size is multiplied by the XML conversion factor for that message.

- size of DTD (if it exist) for the message (in bits),

- size of CSS (if it exist) for the message (in bits),

- compressed size (in bits) : indicates the average sized file after compression,

- XML format compressed size (in bits) : indicates the average sized XML file after compression,

- compression time of the average sized file (in miliseconds) : is used during transmission time calculation for compressed file,

- compression time of the average XML file (in miliseconds) : is used during transmission time calculation for compressed file,

- message overhead time : is used for message transmission time calculation. It includes message conversion into XML, saving the file on disk and reading the message file for transmission,

- message priority : determines which message is transmitted first. It is used to determine the highest priority message(s) in the queue. If there are two or more such messages with the same priority, the one with the earliest timestamp is choosen,

Other defined attributes;

- the begining (the lower time limit) and ending time (the upper time limit) for messaging : is system (link) free time between these time limits,

- minimum and maximum busy time intervals (limits) : is used for communication line busy time (time used by other system) generation between these two limits,

- total busy time in a day : the total amount of time in minutes the link is not available for messaging (line is used by other system); that is to say total busy time for a link between the begining and ending time of messaging,

- minimum time approach : the minimum time between the two busy time interval to prevent the coverage of the same time interval,
- simulation run : denotes one pass for gathering one simulation data for each type of message.
- simulation time : is the time which is used instead of system time.
- sending time of a message: is hour, minute and second of the message for transmission.

5.2.3. Functional and Behavioral Capabilities

Types of military units, units, communication links among these units will be defined for simulation. Units, communication links' capacities and messages which are used by units must be definite or identifiable. These objects' attributes must be detailed. Units are defined to use simulation and possible message types, and the types of messages that are sent by units are determined. These are main objects in tactical theatre for simulation. The message arrival times and communication line's busy times are generated randomly by using information about message types and user given parameters. Message waiting times are calculated and saved to database. After completing the simulation of messages between units, various graphichs are produced for message traffic enabling comparison and analysis of message formating alternatives on each links.

5.2.4. Logic and Algorithms

Line Busy Time Generation : Communication media is not always free to transmit a message. Radio silence, enemy electronic jamming and other systems make system busy. The time betweeen the begining time and the ending time for messaging (system is free between these time limits) can be used for data communication; other times radio silence, enemy electronic jamming is carried out (this means system cannot be used from the upper level time limit to message lower time limit) or other systems use the communication/data link. Communication line busy time is critical for wireless communication media. Four parameters are defined, that is minimum busy time limit, maximum busy time limit, total busy time in a day, and minimum time approach to generate random busy time intervals.

First, usable time in a day is calculated between the message lower and upper level time limits. Total busy time in a day cannot exceed the usable time in a day. Then, maxsimum free time (possible maximum time between two busy time interval) is calculated and line’s free time is randomly generated. Calculation of maximum free time and line’s free time between two time interval formula is shown in Figure 5.2. Then, line busy time interval is generated randomly by using maximum busy time. Line busy starting hour, minute and busy time ending hour, minute are calculated. If generated time does not cover busy time which is not recorded before, line number, busy time starting hour, minute and busy time ending hour, minute is saved to the database table. This process is repeated until overall busy time is greater than Usable Time in a Day parameter or generated busy hour, minute is grater than Time Upper Limit.

Usable Time = (Hour Upper Limit - (Hour Lower Limit+1)*60+(60-Minute Lower Limit)+Minute
Upper Limit Usable Time = Usable Time - Total Busy Time
Busy Time Number= ROUND(Total Busy Time in a Day / (Max.Busy Time Interval+Min.Time Aproach)
Max.Free Time Parameter = ROUND(Usable Time In a Day/ Busy Time Number)

Figure 5.2 : Calculation formula of maximum line free time

Activity diagram and pseudo code of this part is shown in Figure 5.3 and 5.4.

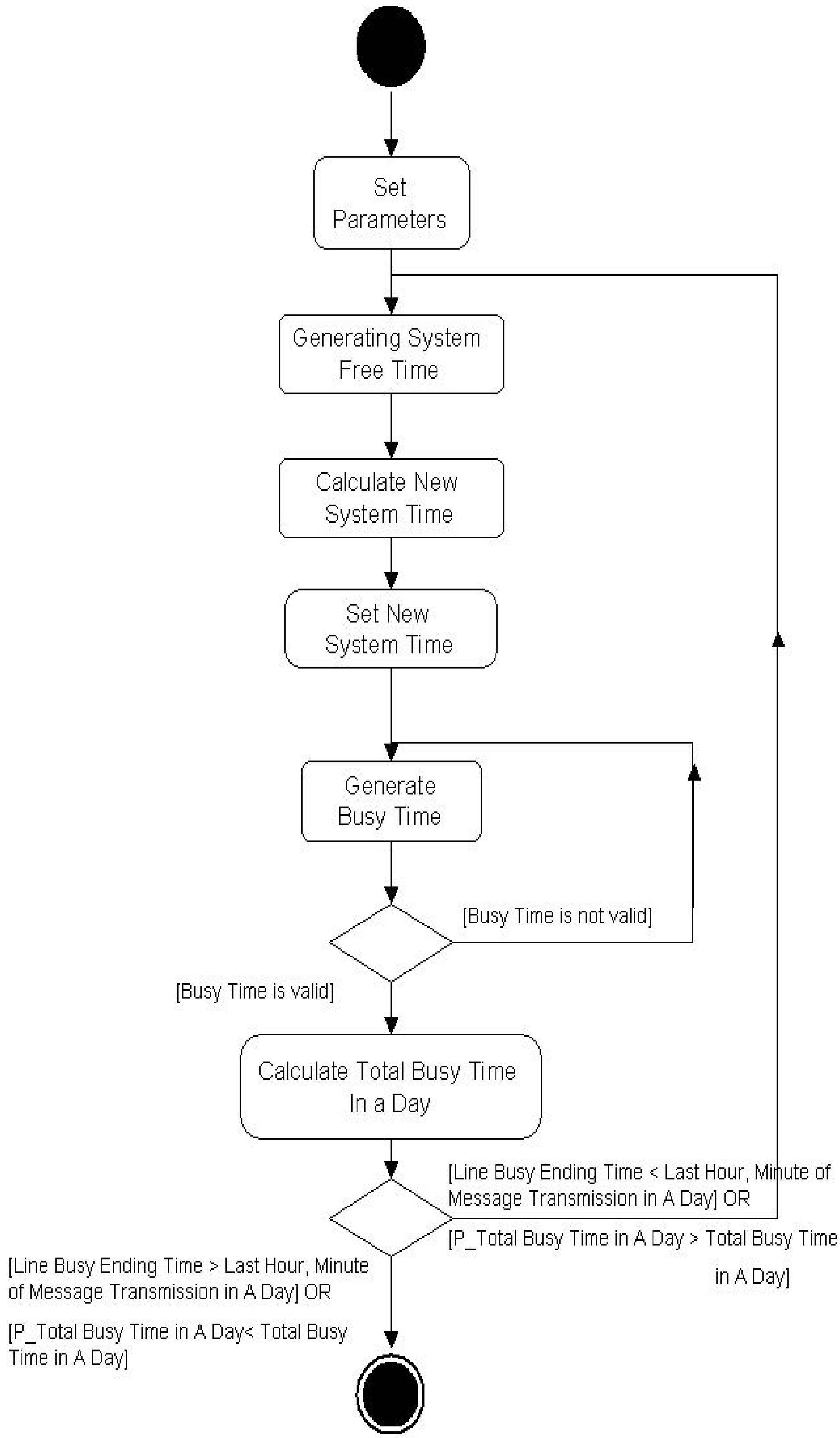


Figure 5.3 : Activity diagram of line busy time generation

```
Set All Parameters( minimum busy time interval, maximum busy time interval,
                    total busy time in a day, minimum time between two busy intervals;
                    Line Busy Starting Hour, Minute = Starting Hour, Minute of Message Transmission In a Day;
                    Line Busy Ending  Hour, Minute = Starting Hour, Minute of Message Transmission In a Day;
                    Total Busy Time In A Day       = 0;
                  )
Delete All Records in “T_HAT_DOLULUK” Table;
Go to First Record of the “T_HATLAR” Table;
REPEAT // Repeat-1
     Calculate Maximum Line’s Free Time;
     REPEAT  // Repeat-2
          Free Time   = Generate System Free Time(Randomize;
                                                   Free Time  = Random (Maximum Free Time Parameter of the Line));
          System Time       = Last Line Busy Starting Hour, Minute + Free Time;
          Busy Time   = Generate System Busy Time (Random (Maximum Busy Time Interval));
          IF (Busy Time >= Minimum Busy Time Interval) THEN
             BEGIN
                IF Busy Time = OK THEN
                     Total Busy Time In a Day      = Total Busy Time In a Day + Busy Time;
                     Busy Hour, Minute of the System = System Time + Busy Time;
                     SAVE Record( Busy Hour, Minute of the System) TO “T_HAT_DOLULUK” Table
                ELSE
                     Generate New Busy Time;
             END;
     UNTIL (Line Busy Ending Time > Last Hour, Minute of Message Transmission In a Day) // Repeat-2
      “T_HATLAR” Next;
UNTIL (End of “T_HATLAR” Table’s Last Record)  // Repeat-1
```

Figur

Figure 5.4 : Pseudo code of the communication line busy time creation

Sending time of a message; hour, minute and second, is generated randomly between the message lower and upper time limits.

Message Transmission Process : While one message is transtmitted, other messsages are stored in queue, that is a table. After finishing a message transmission, another (next) message can be sent. If more than one messages are waiting in the queue the message which has highest priority will be sent first.

The normal message size is calculated by **adding message header size** and **message body size**. The size of XML message files are computed by multiplying message header and text size with XML conversion factors.

"Second" part of time parameters is represented as milisecond for sensitive time calculation. Queues are database tables. Line busy times are queued by increasing order (they are, tables data, ordered with an SQL statement). First of all, line busy time is sorted by increasing order. The messages are queued, then prioritized (they are ordered by message's hour, minute, second and priority) and transmitted. Simulation time (prior message time) denotes last transmitted message time. Simulation time and prior message hour, minute, second parameters are set to zero at the begining of simulation. First of all, line busy time and message time are read. If line is not busy, the current message time is compared with simulation time. If simulation time is less than the current message time, message is marked as "transmitted". If not (simulation time is greater than the current message time), message is pushed into waiting message queue. Current message parameters and waiting time is saved. Simulation time is set to sent message time (line occupation time, message compression time, if it is compressed file, and message overhead time is added to current message time) and next message sending time is read. This process is repeated until the simulation time is less than current message time, then all messages are transmitted from queue, waiting time is calculated and saved. If line is busy, message time is pushed into the waiting message queue, simulation time is set to ending hour, minute of line busy time, then next message time is taken and comparison process is repeated until current message time is greater than simulation time (ending hour, minute of line busy time). After that, new line busy time is read, and waiting message queue is checked. If it is full, message transmission time and waiting time is calculated for each message, marked as transmitted and new simulation time is set to last sent message time (line occupation time, message compression time, if it is compressed file, and message overhead time is added to the last transmitted message time from queue) until queue is empty. Waiting message parameters are saved for later analysis. All process which is explained above is repeated until all of the messages which belong to that line finish. This process is repeated for all types of messages (such as normal, minimum, average, maximum, their XML counterparts and compressed forms of all) for one simulation run.

Pseudo code of this part is shown in Figure 5.5 and Figure 5.6.

```
PROCEDURE  TRANSMIT WAITING MESSAGES
    SET flag := True;
    SET Prior Message Parameter to line busy time ending hour, minute;
    WHILE NOT flag DO
    BEGIN
         READ First Record From Waiting Message Queue Table
         IF Line is Busy THEN
            Save Record in "T_MSJ_BEKLEME";
            flag := False;
         ELSE
             SET Simulation Time/Prior Waiting Message Parameters
    END-WHILE
```

Figure 5.5 : Pseudo code of transmission of waiting messages

```
PROCEDURE  SEND MESSAGES;
BEGIN
    READ First Busy Time;
    GOTO First Message;
    WHILE NOT QUERY Line's Messages END of FILE DO
        READ Message Time; (PROCEDURE)
        IF Line is Busy THEN
            IF There is/are message(s) in waiting message Queue
                PUSH Message in Queue (Sending Time< Line Busy
                                       Time Starting Hour)
            SET waiting reason ="HAT_DOLULUK"
            PUSH Message in Waiting Message Queue
            SET New Simulation Time as Line Busy Ending Time
        ELSE
            IF There is waiting messages THEN
               TRANSMIT Waiting Messages
            ELSE IF Message Time > Busy Time THEN
                    SET Current Message as "Transmitted"
                    SET New Simulation Time
        END-ELSE-IF;
        GO TO Next Message;
    END-WHILE
END;
```

Figure 5.6 : Pseudo code of messages transmission

5.2.5. Relationships

Army units have hierarchical relationship each other. Brigade is main maneouvring unit in TLF. The organizational outlook of a brigade and the lower level formations, battalion and company level nodes, and the connections between them are modeled. Two or more battalions (as a maneouvring unit), fire support battalion (artillary units, v.s.), combat service support battalion, and the support and the command support companies (headquarter company, engineering company, maintenance company, antitank company and antiaircraft company) are generally subordinate units of a brigade. Battalions generally have four or more companies (headquarter and service/combat support company, and three maneuvring companies; such as tank company, mechanised infantry company).

They have an hierarchical command link from top (superior unit) to bottom (subordinate unit).

The main means of communication is wireless radio including Frequency Modulation (FM) and High Frequency (HF) type radio devices, telephony, written messages via teletypewriters, and hand typed written messages delivered by courier. The current means of communications is provided by a signals company (all of these are provided by different sub-units of the signals company). A communication infrastructure, basic communication lines, is being built to take into consideration hierarchical command relationship. Separate data link is allocated for each command link. A unit can only sent a message to its superior, subordinate or neigbour units. Assumed organisational hierarchy and communication media is shown in Figure 5.7.

There is a wireless backbone within the brigade for networking purposes. Some of the used devices are hand-held and some of them are mounted on vehicles. Usage of TASMUS backbone for data communication is assumed as the main link between a brigade and its sub-units. TAFICS and TASMUS, together, have been used together for larger army units. As a result of the network infrastructure, the network will be able to handle different types of message traffic within the Brigade area. The TASMUS architecture is TDMA which is providing increased flexibility and decreasing network management overheads. Voice and data communication can use the same time slot. Each slot is divided in two parts; one for voice, one for data. They are not effected by each other.

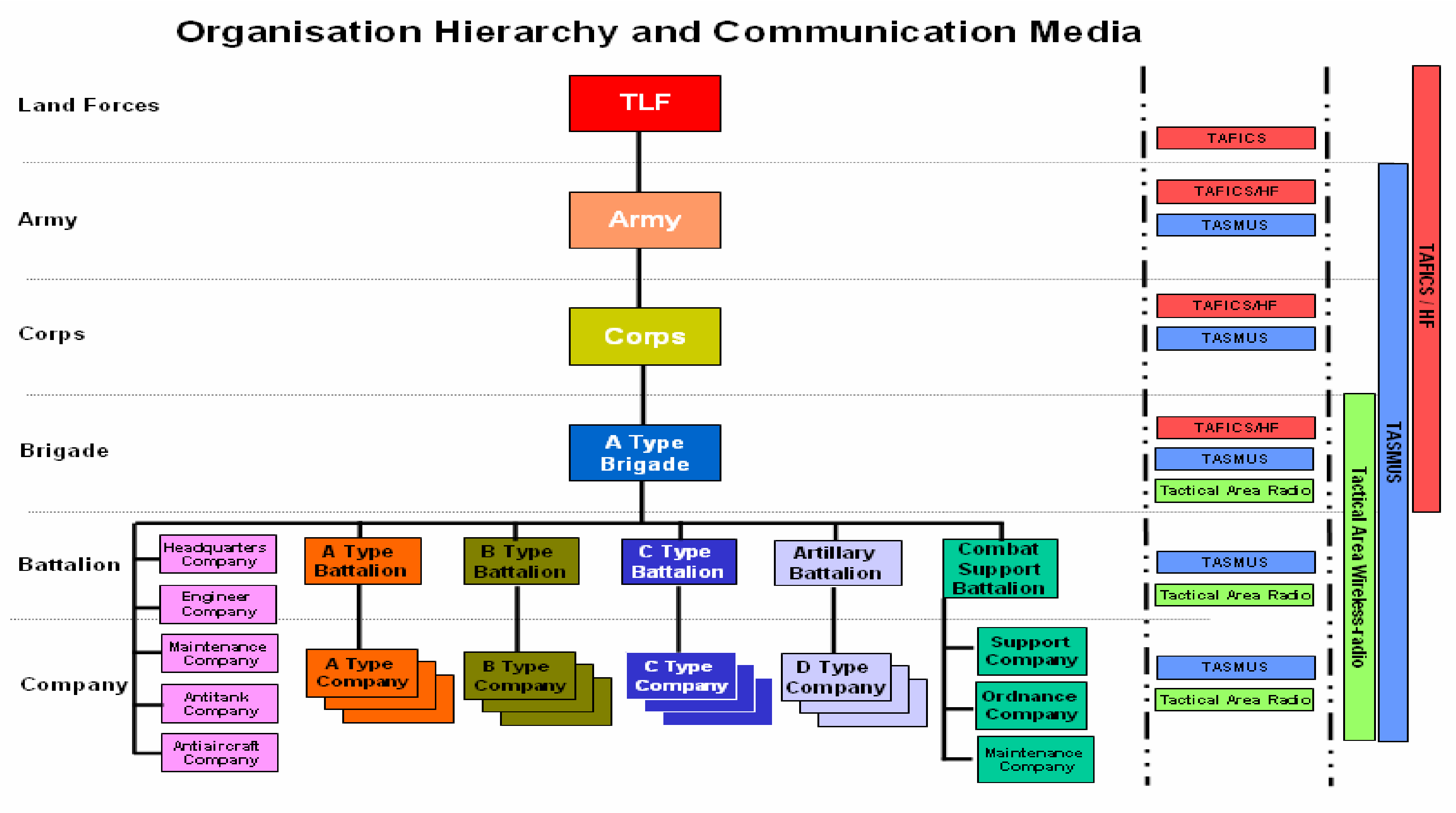


Figure 5.7: Organisational hierarchy and general communication media of the modeled brigade

5.2.6. Assumptions and Limitations

Current TLF messaging technology is based on manual message processing. The main subject of this thesis take this assumption into consideration. Defined messages which belong to a communication line are independent from other messages that belong to other communication lines. For example; A, B, and C-type messages which are transmitted from lower level units are not automatically processed and combined into one messages to send superior/subordinate units. Their format and timing is not appropriate. Message flow of lines is independent each other. Tool does not meet the need for specific message traffic flow, but can be improved.

Average size of a message is assumed as 1500 bits. Normal message size is obtained by **adding message header size** and **message body size**. While one message is transtmitted, other messsages are waited in queue. After finishing a message transmission, another (next) message can be sent.

A communication line can be used by only two units. In reality some communication lines, that are common/shared lines, can be used more than two units. The main link between a brigade and its sub-units will use TASMUS backbone for data communication. TAFICS and TASMUS will be used together for larger army units.

5.3. Simulation Tool Description

5.3.1. General Overview of The Simulation Tool

Main purpose of this tool is to verify that XML technology can be used effectively in the tactical area of military. Simulation tool is developed by using Delphi Visual Programing Language. [70] Whole data (defined units, messages, analysis data) and parameters are saved in database. Interbase/Firebird database engine[71] is used for physical database. Software diagrams are prepared with Unified Modeling Language (UML). Cabinetmaker tool is used as a compression program for calculating compression size and compression time of defined files, but compression data can be gathered externally and used for simulation.

Brigade is main maneouvring unit in TLF. A simulation approach have been used in analysing model because of the organizational outlook of the brigade.

Starting with the general outlook of the brigade, the lower level formations, battalion and company level nodes, and the connections between them are modeled. Then the interactions of the formations with one another are developed. A brief description of the structure of the Brigade is also covered below. The Brigade which has 5 battalion level units and more than 20 company level units, which is the subject of this study, is located in one of the metropolitan cities of the country. I modeled a mechanised infantry brigade, which is called 3rd Mechanised Infantry Brigade[29], with five battalions, one tank battalion (6th Tank Battalion), one mechanized infantry battalion (5th Mechanized Infantry Battalion), one border defence battalion (4th Border Defence Battalion), one artillary battalion (7th Artillary Battalion), the support and the command support units (31, 32, 33, 34 and 35th Company) and one combat support battalion, (command service support units, that is 8th Combat Support Battalion). The brigade is a sub unit of the 2nd Corps, thus it has to be performing some radio communications, messaging and reporting activities with the Corps and the Army level. A detailed structure of the brigade (the organisational hierarchy) is provided in Figure 1.1.

C2 Messaging Simulation Tool was developed to create the model and analysis of messaging among TLF units and its sub-units. It is a general purpose, dynamic, graphical simulation tool that enables the user to simulate a dynamic communication system accurately and easily. No programming is necessary for the user. Tool enables the user to define new connection lines. Each line between the units of 3rd Mechanised Infantry Brigade are specified for later use. The current means of communications is provided by 32nd Signal Company. All of these communication media are provided by different sub-units of the signals company. Also a communications infrastructure is being built that is explanined in section 5.2.5 and the result of this study can be used to analyze that infrastructure. There is a wireless backbone within the brigade for networking purposes. Some of the used devices are hand-held and some of them are mounted on vehicles. As a result of the network infrastructure, the network will be able to handle different types of message traffic within the Brigade area.

The overall network of 3rd Armored Brigade will have a wireless backbone. Assumed bandwidth of the lines between the brigade and battalion level nodes is

[29] Imaginary numbers are given to every units in the project . They are not real.

2 mbps and lines between the battalion and company level nodes have a bandwidth of 4800 bits/sec or 5600 bits/sec to go from one unit to another in the brigade. A detailed communication structure of the brigade is provided in in Figure 5.8 and in Figure 5.9.

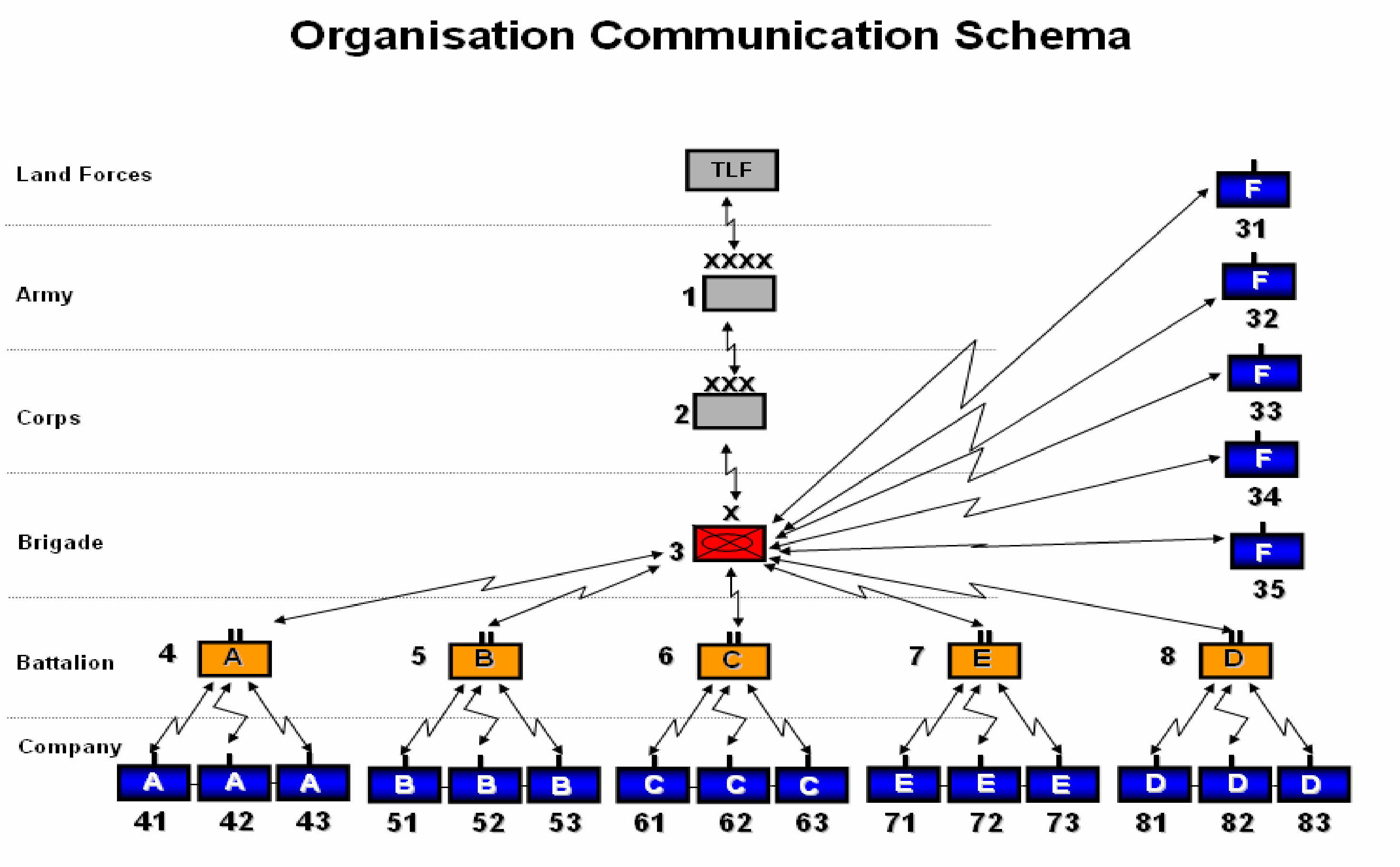


Figure 5.8 : Communication links of the modeled brigade

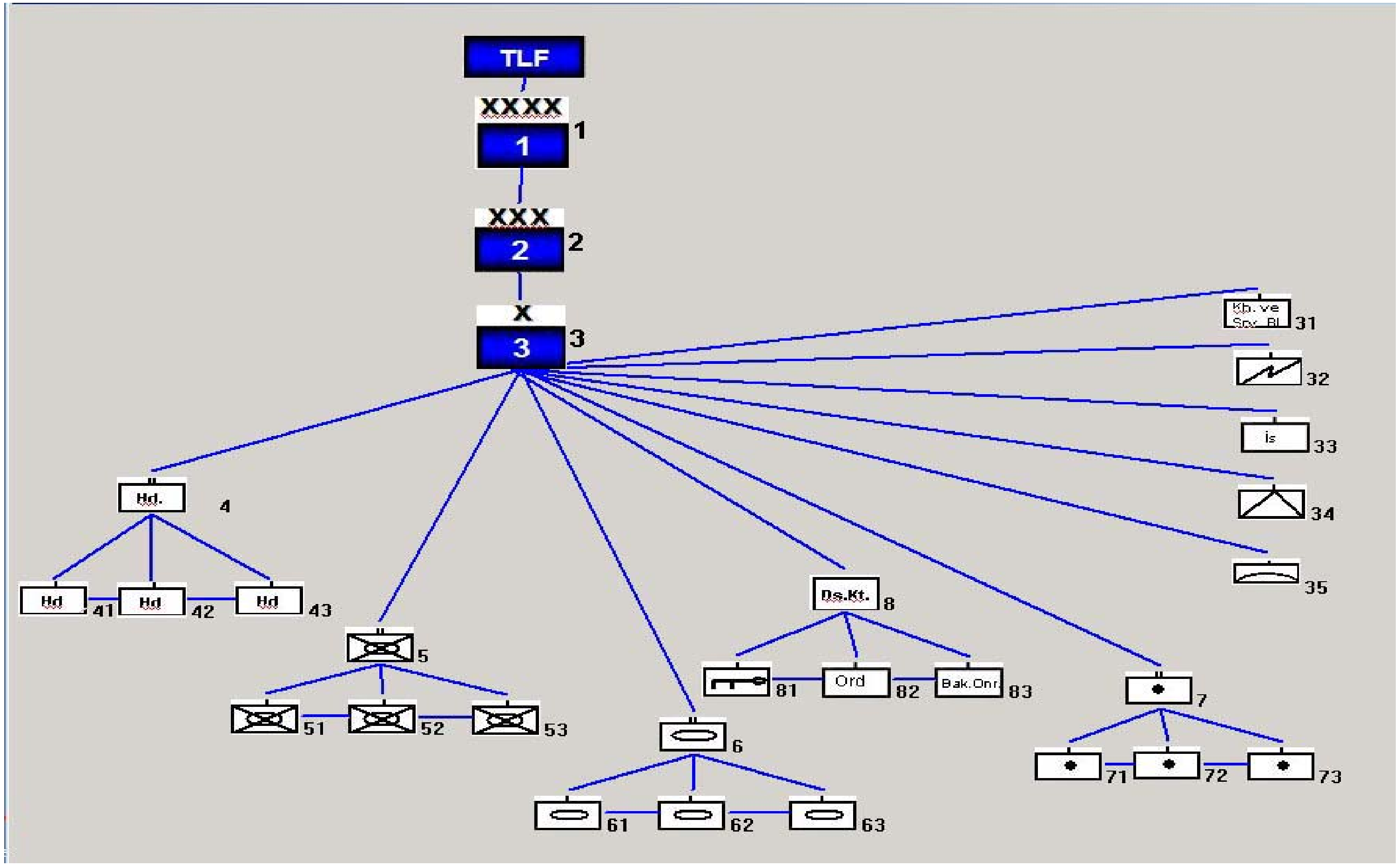


Figure 5.9 : Communication links snaphot of the modeled brigade

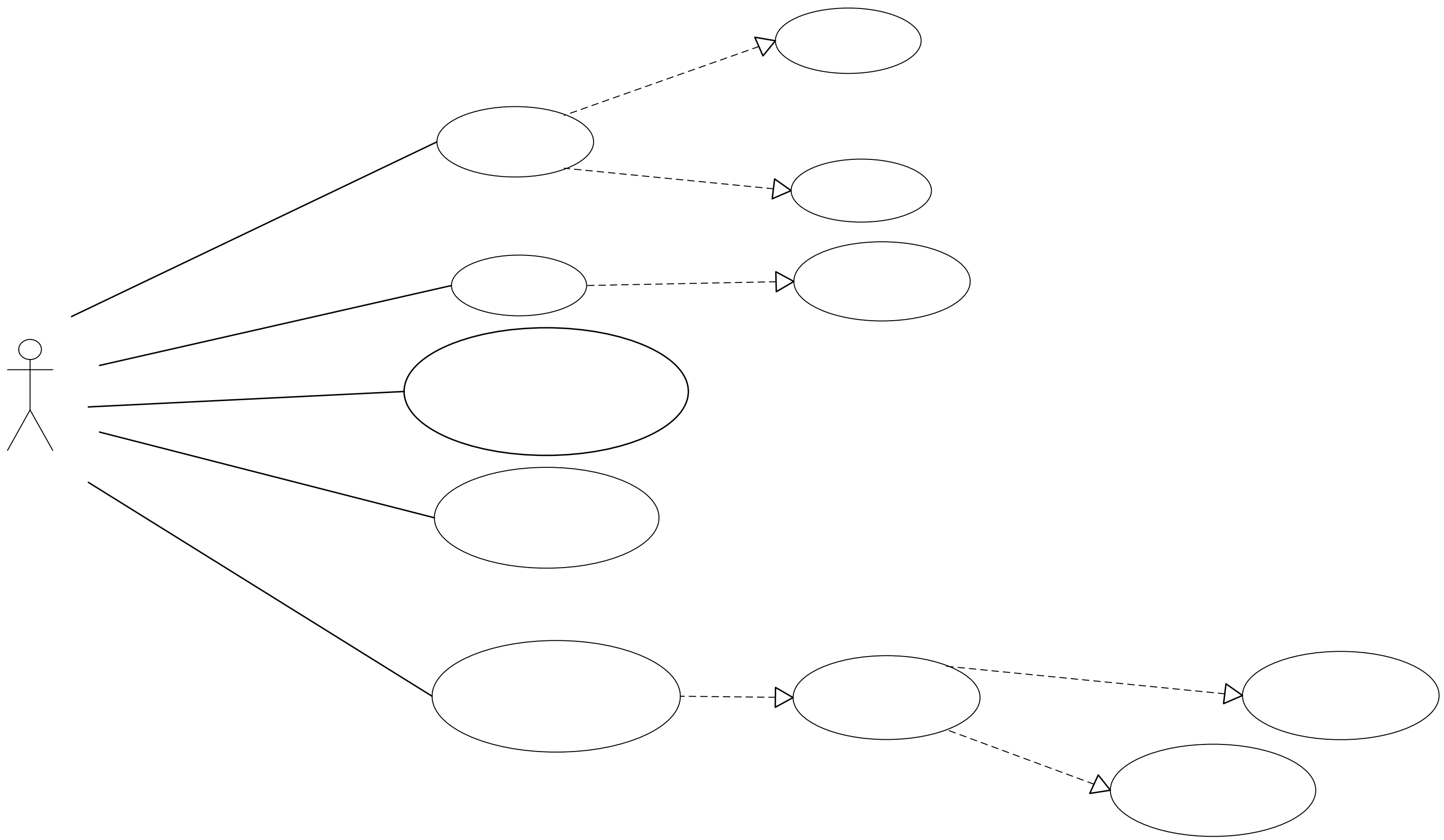


Figure 5.10 : Use case diagram of simulation

Use Case diagrams of the tool is provided in Figure 510. Tool enables the user to define new message types, that is the main subject of this study. Document Type Declaration (DTD) and Cascading Style Sheet (CSS) part or format files are assumed to be stored locally at the source and destination for all messages, therefore they are not included in transmission time calculation. Currently only four message sizes are simulated (minimum, average, maximum, a user defined “normal”, and their XML formatted counterparts). Defined messages which belong to a communication line are independent from other messages that belong to other communication lines. **MEFORS** messages have been studied**,** which is explained in Section-2.2.2, and four kinds of message books which define reports/messages currently used in the Turkish Army. These standards ***prescribe the syntax, semantics of structured alphanumerical messages*** *and message periods*, most of which were originally designed for teletypewriters or telex. They encode the agreements on information exchange for everything from logistics to intelligence reports. Current messages are not automatically processed and combined into one message to send superior/subordinate units. Their format and timing is not appropriate. MEFORS (Message Formating System) of Turkish Army meets formating needs. But **timing and combination of messages into one (or more) will be another invastigation area** for automatic message processing for TLF. Tool does not meet the need for specific message traffic flow, but can be improved to meet these type of needs.

Within the scope of this study, average size of a MEFORS message is assumed as 1500 bits, that is explained in Section 5.2.6.. Message sizes, which is used by 3rd Mechanized Infantry Brigade, is based on this assumption. Normal message size is obtained by **adding message header size** and **message body size**. The size of XML message files are computed by multiplying message header and text size with XML conversion factors. Compression time of messages can be obtained by using cab compression program, if there are real messages, during message definition. It is used in message transmission time calculation formula for compressed file.

After that, messages which are used by units are determined. The parameters of sending and receiving units, message type, priority, frequency in a day (number of messages that is transmitted in a day) are mandatory fields to simulate a message traffic of military units.

Communication media is not always free to transmit a message, that is explanined in section 5.2.6. Total busy time in a day cannot exceed the usable time in a day (this means system cannot exceed total usable time, which is in minutes, from the message lower time limit to the message upper time limits in a day). Communication line busy time is critical for wireless communication media. Communication lines can be busy (that is used by other system) at some time intervals in a day or cannot be used because of radio silence. Four parameters are defined to generate random system busy time intervals, that is minimum busy time limit, maximum busy time limit, total busy time in a day, and minimum time approach. These are explained in Section 5.2.3. System busy time generation process is explained in Section 5.2.4. System busy times have been generated for each defined communication lines.

Sending time of a message; hour, minute and second, is generated between the message lower and upper time limits for all messages and each message frequency. Eight different sizes for each message type are defined; ***normal, minimum, average, maximum, XML, minimum XML, average XML, maximum XML*** and their compressed forms. Size of the compressed files are computed to proportion the compressed average file size.

First of all, line busy time is sorted by increasing order. The messages are queued, then prioritized (they are ordered by message's hour, minute, second and priority) and transmitted. Simulation time (prior message time) denotes last transmitted message time. Simulation time and prior message hour, minute, second parameters are set to zero at the begining of simulation. First of all, line busy time and message time are read. If line is not busy, the current message time is compared with simulation time. If simulation time is less than the current message time, message is marked as "transmitted". If not (simulation time is greater than the current message time), message is pushed into waiting message queue. Current message parameters and waiting time is saved. Simulation time is set to sent message time (line occupation time, message compression time, if it is compressed file, and message overhead time is added to current message time) and next message sending time is read. This process is repeated until the simulation time is less than current message time, then all messages are transmitted from queue, waiting time is calculated and saved. If line is busy, message time is pushed into the waiting message queue, simulation time is set to ending hour, minute of line busy time, then next message time is taken. and comparison process is repeated until current message time is greater than simulation time (ending hour, minute of line busy time). After that, new line busy time is read, and waiting message queue is checked. If it is full, message transmission time and waiting time is calculated for each message, marked as transmitted and new simulation time is set to last sent message time (line occupation time, message compression time, if it is compressed file, and message overhead time is added to the last transmitted message time from queue) until queue is empty. Waiting message parameters are saved for later analysis. All process which is explained above is repeated until all of the messages which belong to that line finish. This process is repeated for all types of messages (such as normal, minimum, average, maximum, their XML counterparts and compressed forms of all) for one simulation run.

The number of messages for each line are increased (number of messages is multiplied one through eight for each simulation run) during new simulation run. At the end, each simulation run (that is to say message number multiplier) and waiting time in miliseconds, on every communication line can be compared for all file types one by one and various combinations. Class diagram of the tool is provided in Figure 5.11.

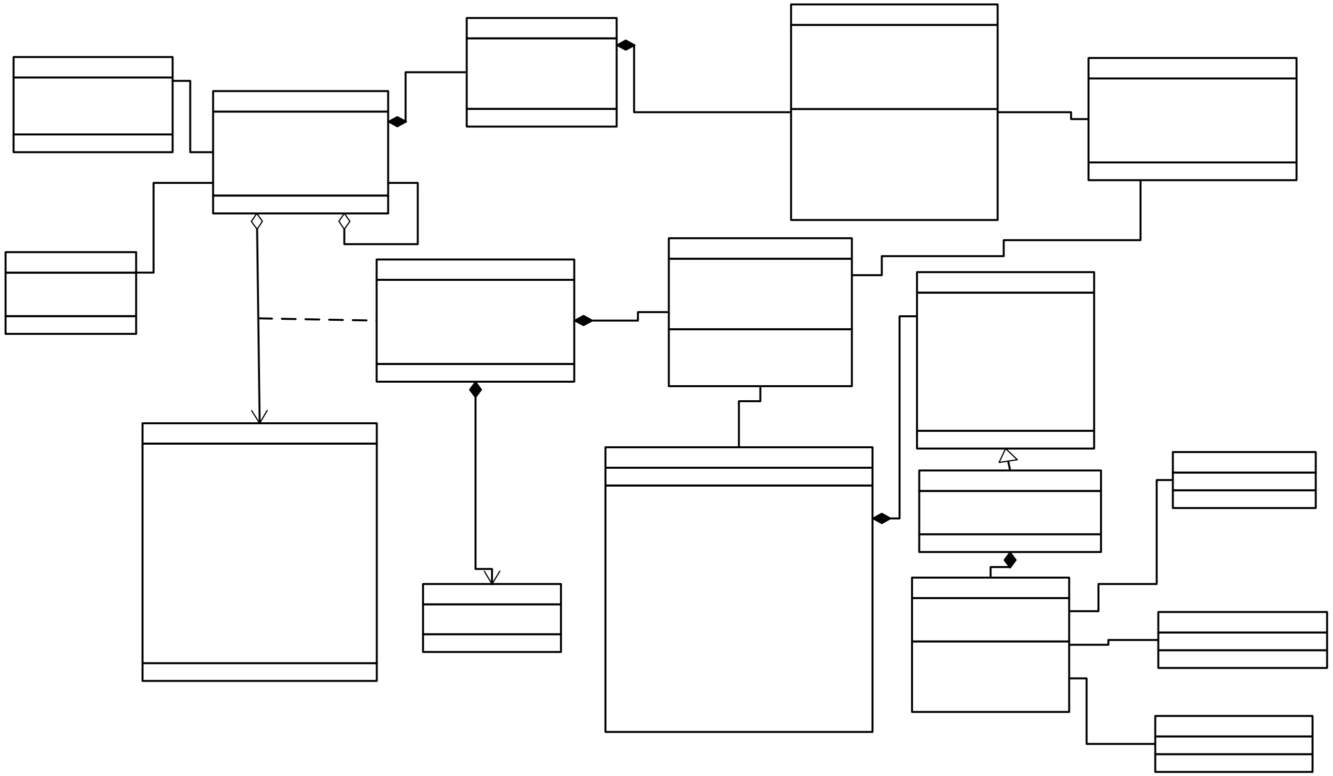


Figure 5.11 : Class diagram of simulation tool

1 .. *

UI Unit Type

5.3.2. Database Model

Entity-relationship (ER) diagram is prepared with methodology of Information Engineering (IE); which graphically depicts entities, tables, columns, attributes, relationships, and types of relationships (e.g.one-to-one). The ER diagram of the schema used by the simulation tool is shown in Figure 5.12. The logical and physical models look very different. Database sql scripts of the physical model is also given in Appendix-B.

Size of the analysis database and its tables are too big for a small database management system (e.g.Interbase/Firebird). Big database file sizes (bigger than 20 megabyte) have often been better to back-up and restored for high performance on account of (Interbase/Firebird) database system’s small scalibility. Even when it is reduced to a functional subset of the data, the amount of processed data is quite large. To overcome slowing effect of large amount of data/records on the simulation tool and database, the analysis data is gathered into one table, instead of gathering more than one table.

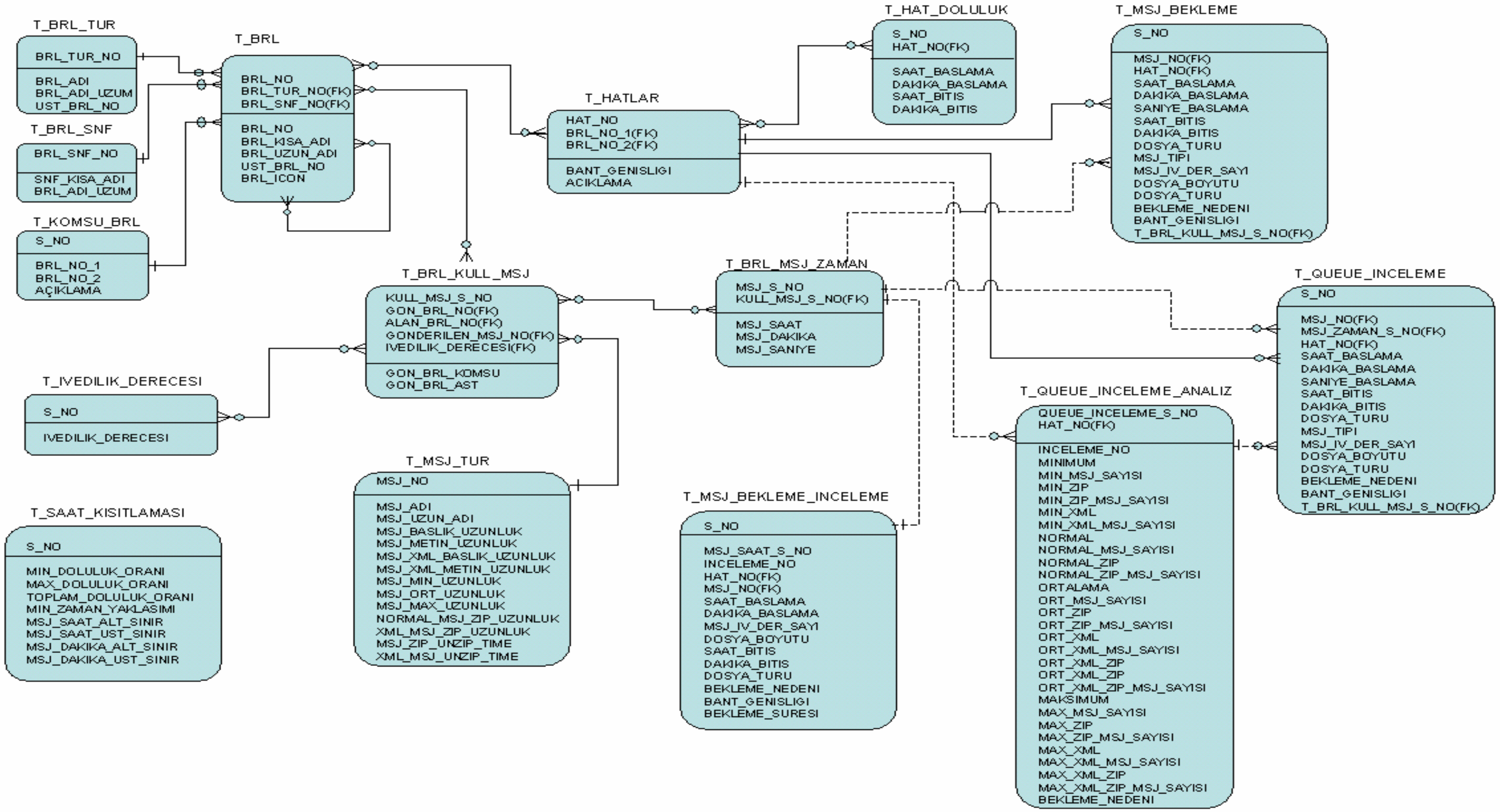


Figure 5.12 : ER diagram of simulation tool’s database

5.3.3. Definition of Unit Types

First, types of units must be defined to create a new unit as illustrated in the Figure 5.13. The defined unit types are stored in table which is named as "T_BRL_TUR".

Figure 5.13 : Unit types definition interface

Sequence diagram is shown in Figure 5.14. and state diagrams are in Figure 5.15 and Figure 5.16.

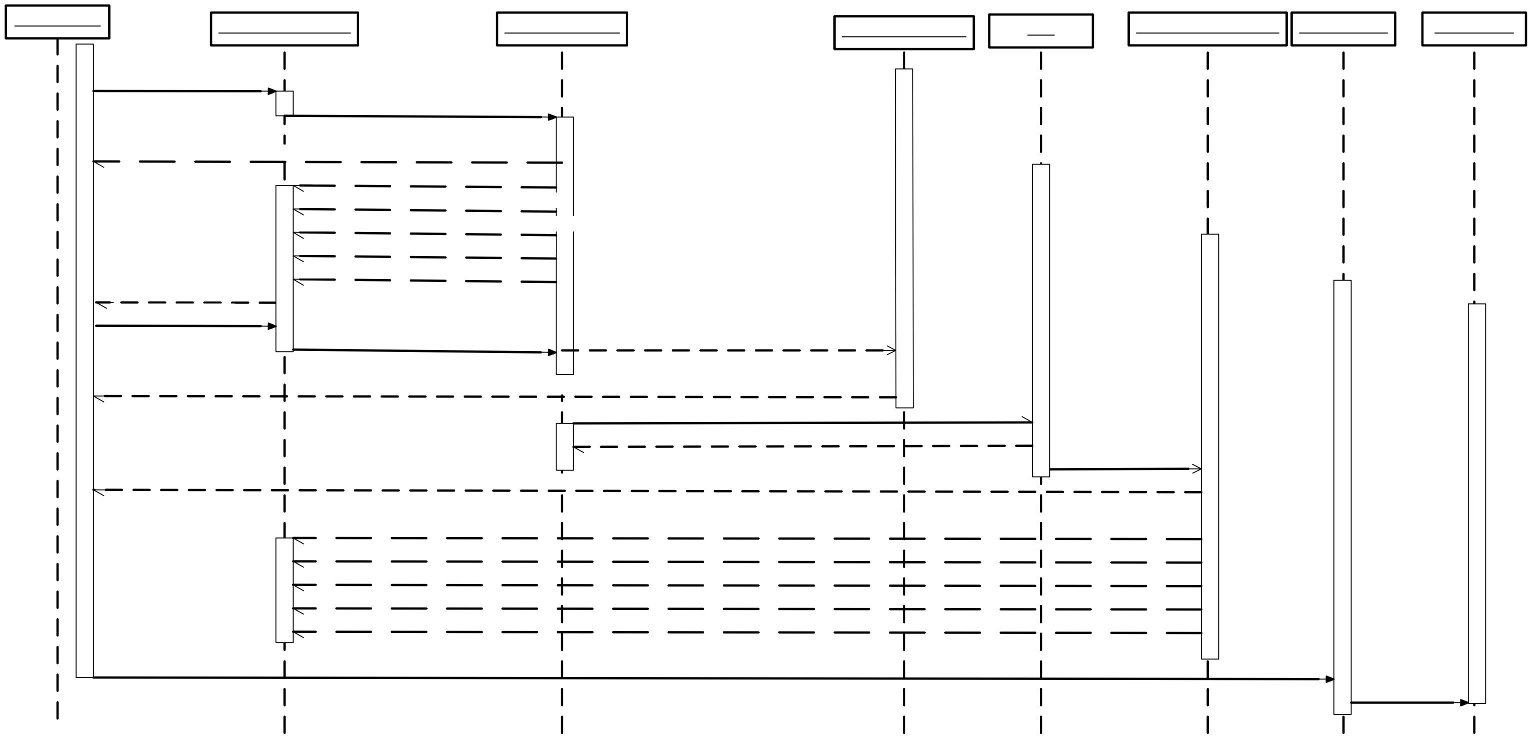


Figure 5.14 : Sequnce diagram of unit type definition

UI Unit Type

Object dbNavigato

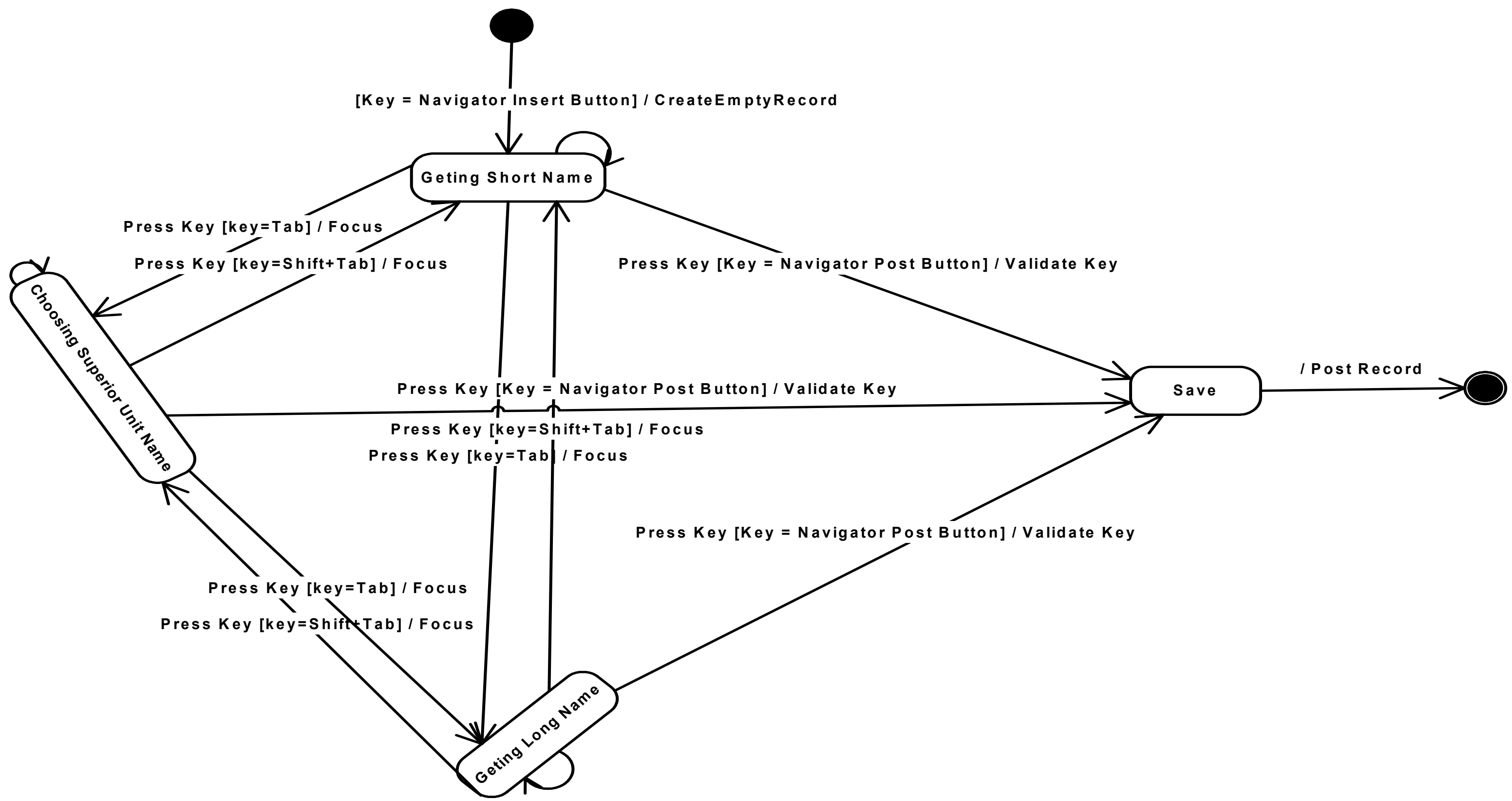


Figure 5.15 : State chart diagram of unit type insertion

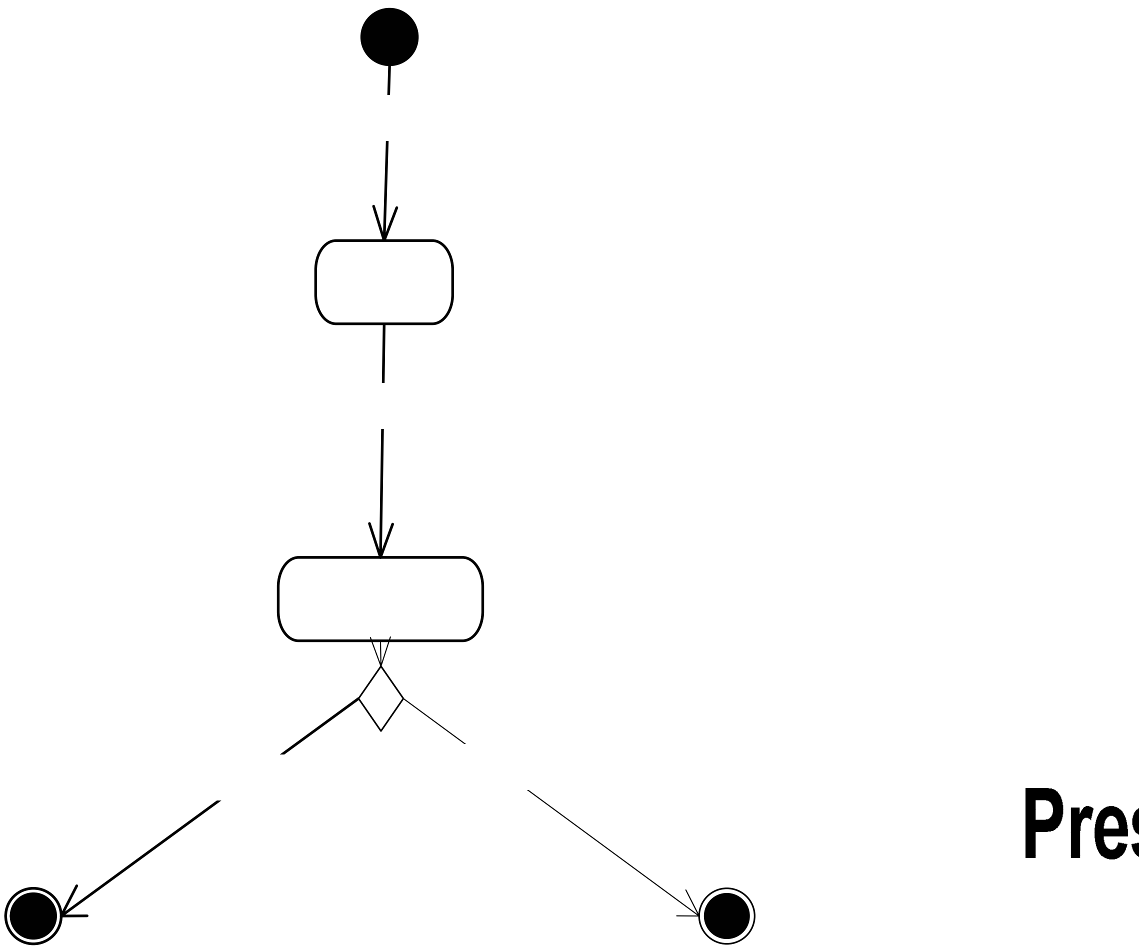


Figure 5.16 : Activity diagram of unit type’s record deletion

5.3.4. Definition of Unit

It enables detailed dynamic unit definition by using the “Definition of New Unit Interface” that is shown in Figure 5.17 Unit types which are defined as shown above are used to choose new unit’s unit type. Parameters that are necessary for a unit definition is explained above section 5.2.3. Defined parameters are stored in a table named “T_BRL”. I defined a brigade (3rd Mechanised Infantry Brigade) and I gave a brief description of its structure above and detailed structure (the organisational hierarchy), that is provided in Figure 1.1 for later analysis. 4th Border Defense Battalion and 41st Border Defense Company‘s information has been processed for given example which is given in Table 5.1.

Table 5.1 : Choosen unit's definition data

Unit Atributes	1st Unit	2nd Unit
Unit Number	4	41
Unit Short Name	4_Hd_Tb	41_Hd_Bl
Unit Long Name	4 ncü Hudut Taburu	41 nci Hudut Bölüğü
Unit Type	Battalion	Company
Unit class	Hudut	Hudut
Superior Unit Number	3	4
Unit Icon's Left Coordinates	200	180
Unit Icon's Top Coordinates	150	200

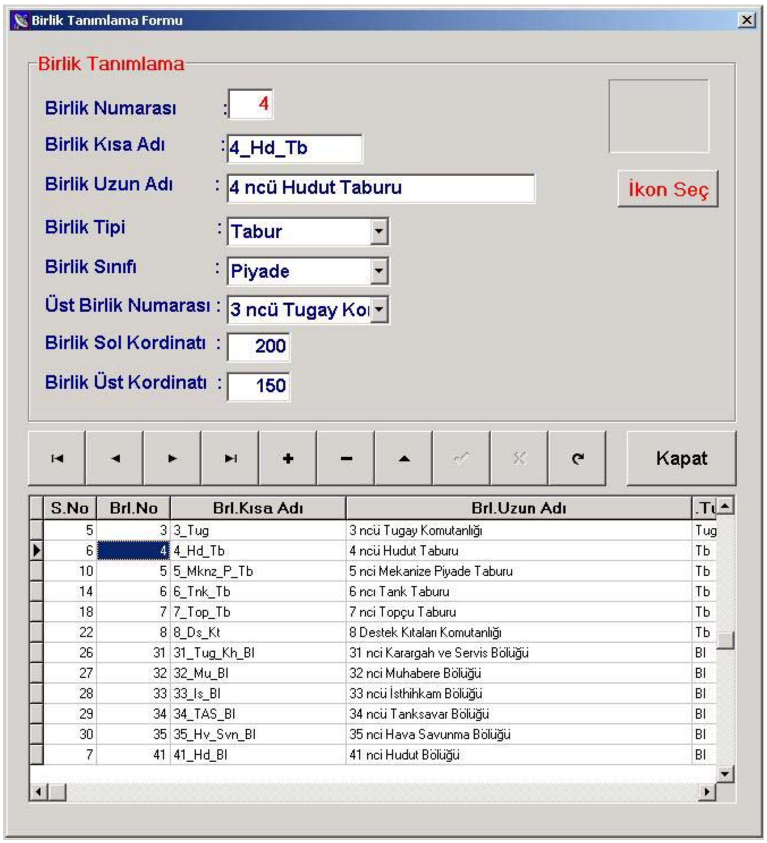
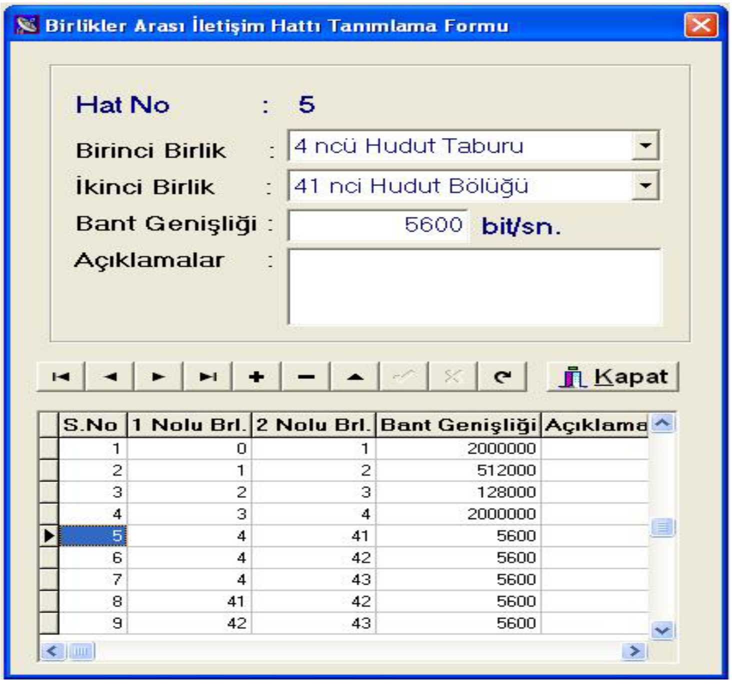


Figure 5.17 : Unit definition interface

5.3.5. Definition of Communication Lines

"Communication Lines Definition Interface", which is shown in Figure-5.18, is one of the two basic definition interfaces with "New Message Types Definition Interface" for simulation. It enables the system to define new connection lines. Each line between the units of 3rd Mechanised Infantry Brigade are specified for later use. They are stored in table which is named as "T_HATLAR". 5th communication line's information is shown below figure.



S.No	1 Nolu Brl.	2 Nolu Brl.	Bant Genişliği	Açıklama
1	0	1	2000000	
2	1	2	512000	
3	2	3	128000	
4	3	4	2000000	
5	4	41	5600	
6	4	42	5600	
7	4	43	5600	
8	41	42	5600	
9	42	43	5600	

Figure 5.18 : Communication line definition interface

5.2.6. Definition of Message Type

This is the most important part of the system. It enables the system to define new message types. DTD and CSS format files are assumed to be stored locally at the source and destination for all messages. Therefore they are not included in transmission time calculation. However, currently only four message sizes are simulated, (minimum, average, maximum, a user defined "normal", and their XML formatted counterparts). Message types' parameters are stored in the table, named as "T_MSJ_TUR". Messages are identified by message type numbers, but users can also define and use "Original Name of the Message" ("Mesaj Uzun Adı") enabling the user to define more meaningful message names. For example, “Type-12/Tip 12 “ message’s long name is defined as “Birlik Durum Bildirme Mesajı(Test)”, which is shown in Figure 5.19. Compression size and time can be calculated authomatically by compression program which name is Cabinetmaker. Compression data can be gathered externally and entered by manually.

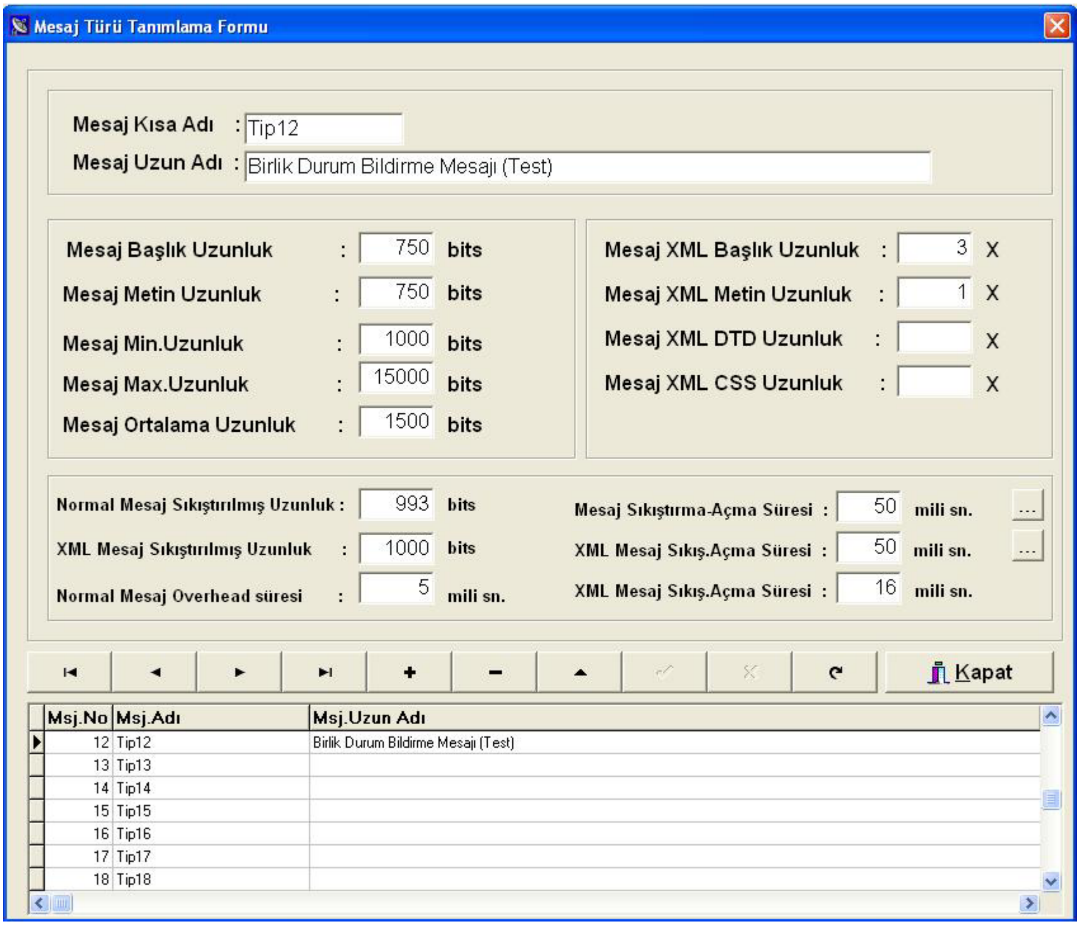


Figure 5.19 : Message types definition interface

MEFORS messages**,** which are explained in Section-2.2.2, are studied for analysis of Turkish Army and TLF messaging system. MEFORS messages are not appropriate for analysis and simulation because they are all classified as “SECRET”. Therefore, imaginary message types and their usage patterns by the artificial army units are produce for comparison of the MEFORS messages that are formatted by AdatP-3 rules, and XML formatted messages. Normal message size is obtained by adding message header size and message text size, which are calculated with the following formula:

Size of a Message = Size of Message Header + Size of Message Text

For example, size calculation of Type-12 messsage is presented below.

Type-12 Message Heading Size : 750 bit

Type-12 Message TextSize : 750 bit

Size of Type-12 Messsage = Size of Type-12 Message Header + Size of Type-12 Message Text

Size of Type-12 Messsage = 750 + 750

Size of Type-12 Messsage = 1500 bit

Type-1 (Tip-1) message has minimum message size, that is 600 bits; type-18 (Tip-18) message size has maximum size, that is 5500 bits. Minimum, average and maximum parameters show total message size. The size of XML message files are computed by the formula that is shown in Figure 5.20 :

XML Message Heading Size = Normal Message File Size * XML Heading Conversion Factor
XML Message Text Size = Normal Message File Size * XML Text Conversion Factor
XML Message File Size = XML Message Heading Size + XML Message Text Size

Figure 5.20 :XML File Size Computation Formula

Type-12 Messsage XML Heading Conversion Factor = 3

Type-12 Messsage XML Heading Conversion Factor = 1

XML Message File Size = 750*3 + 750*1

XML Message File Size = 3000 bit

Normal message overhead time is obtained experimentally by writing, and reading of a AdatP-3 formated message to disk 5.000 times and average time (in milisecon) is calculated. XML message overhead time is also calculated experimentally by writing and reading of same type of XML formatted message to disk 5.000 times and average time (in milisecon) is calculated. If the message is real, if sample message file exist, compression size and time can be computed by compression module which is shown in Figure 5.21.

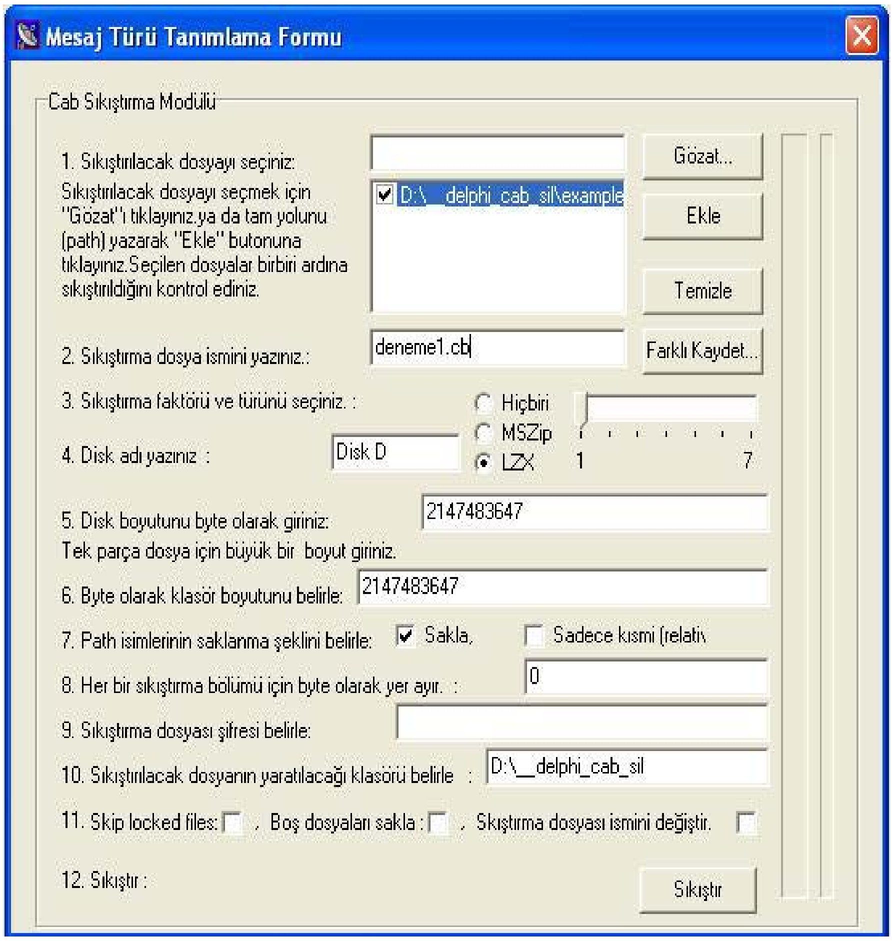


Figure 5.21 : Compression file size and time computation interface.

5.3.7. Determination of The Messages That are Used By Units

TLF C2IS projects have uncertainties because of their sub-projects' uncertainties. HERIKS project needs real-time, bitwise data and uses fixed format messages, that are explained in section 2.2.2.2. The devolopment process of all the other projects is going on. Their messaging needs are not clear yet.

The sending and receiving units, that are defined in “Unit Definition” module, message type, precedence, and frequency in a day can be defined by using the interface which is illustrated in Figure 5.22. Defined units' data are stored in the database table named as “T_BRL_KULL_MSJ”. If real message types are defined, it will represent the real system situation.

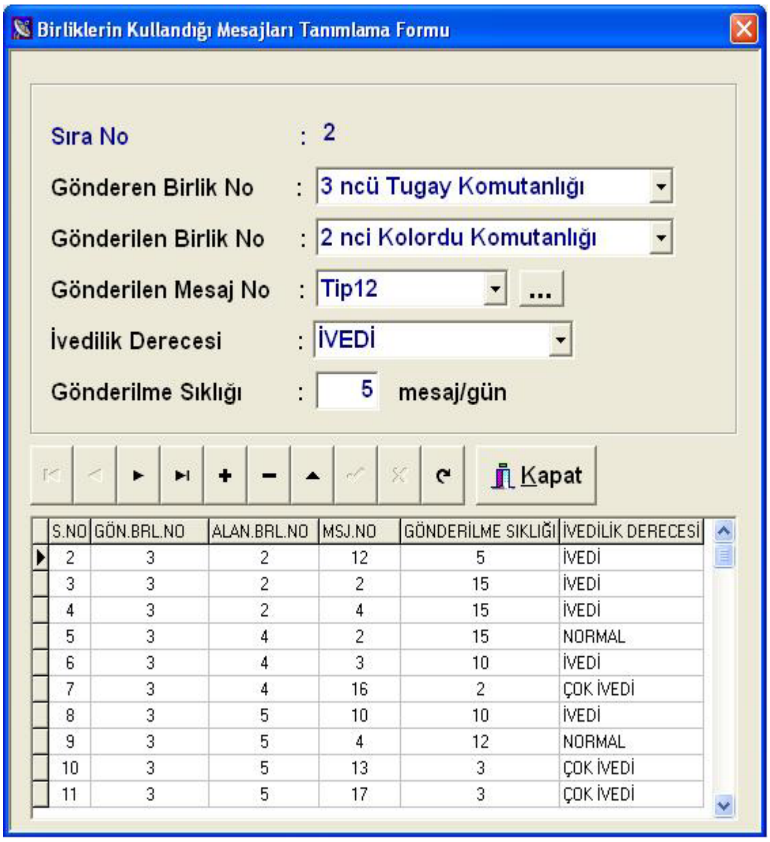


Figure 5.22: Determination interface of the messages that is used by units

5.3.8. Communication Line Busy Time and Message Transmission Time Creation

Communication line busy time is critical for wireless communication media. Four parameters, minimum busy time interval, maximum busy time interval, total busy time in a day, and minimum time between two busy time intervals (used by other communication in that line) have been defined, that is shown in “System Busy Time Parameters/Sistem Doluluk Parametreleri” part of Figure 5.23. All parameters can be set by the user, otherwise default values, that are defined before, are used. Sequence diagram of loading analysis parameters is provided in Figure 5.24. Sequence diagram of analysis is shown in Figure 5.25 and Activity diagram is in Figure 5.26 .

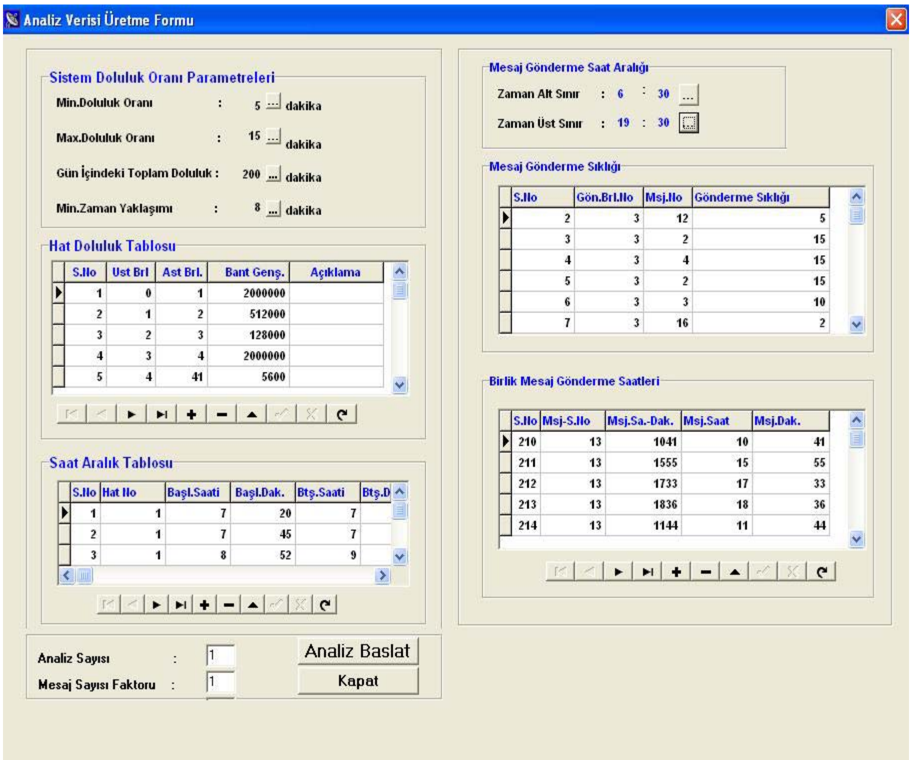


Figure 5.23 : Analysis interface

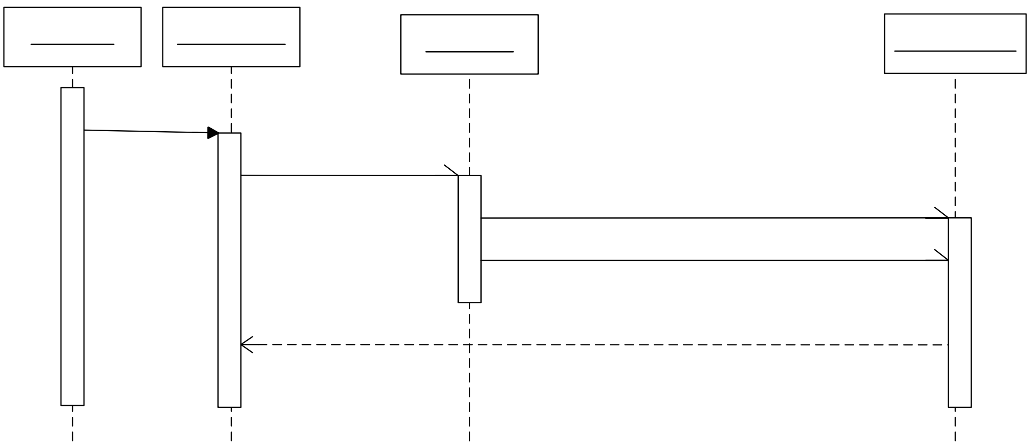


Figure 5.24 : Sequence diagram of loading analysis parameter

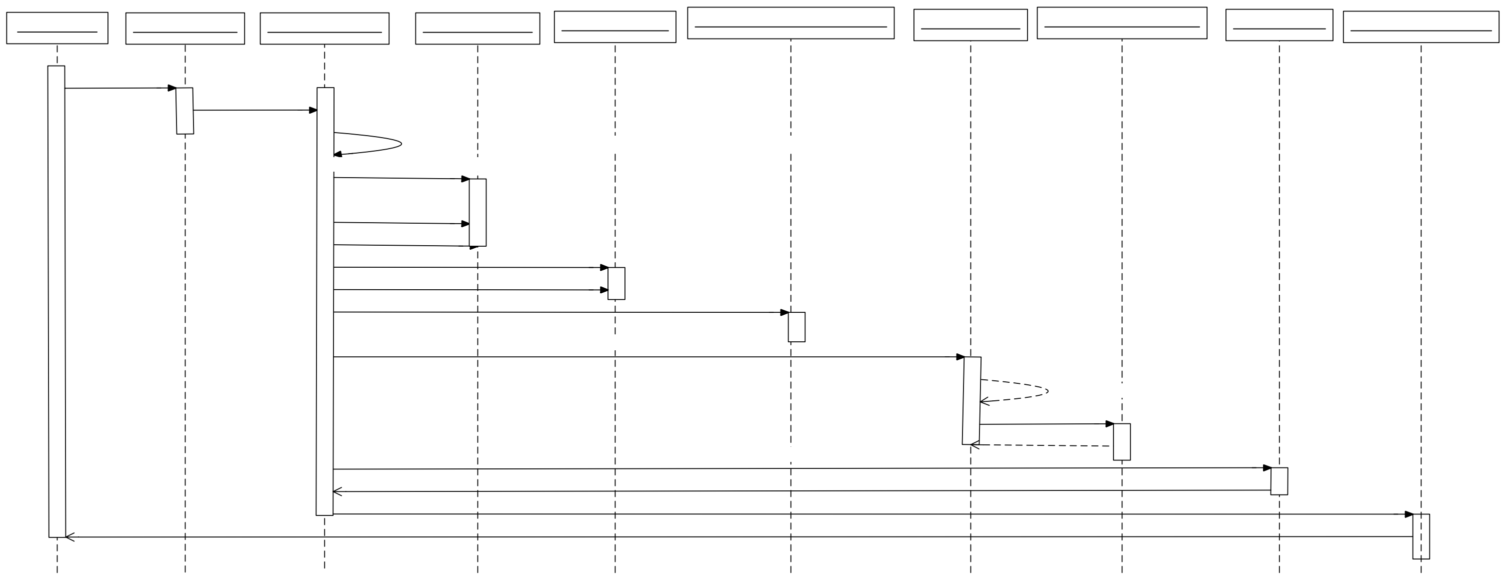


Figure 5.25 : Sequence diagram of analysis

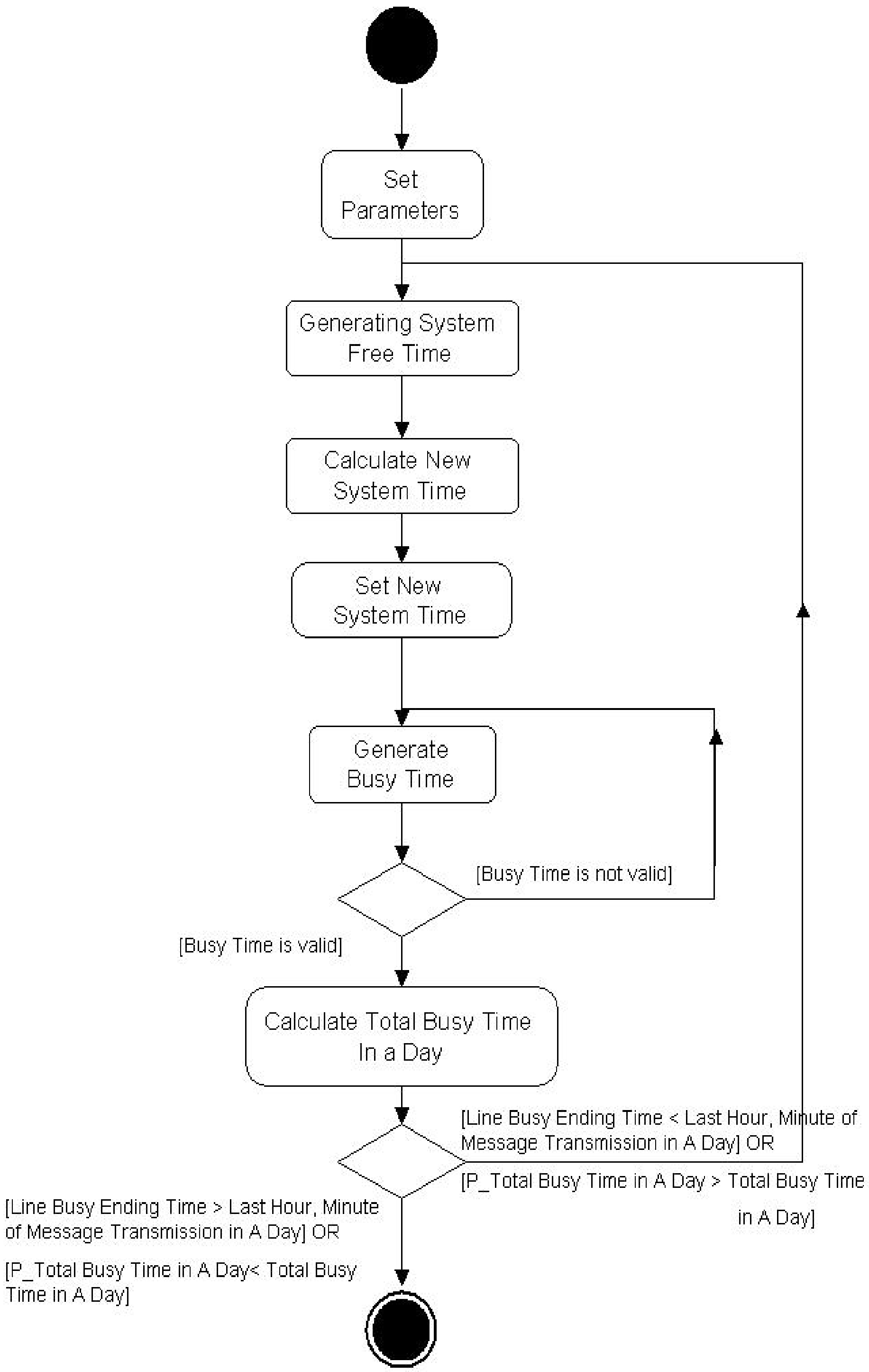


Figure 5.26 : Activity diagram of busy time generation

First of all, any old remaining records are removed from the table named "T_HAT_DOLULUK" for new data insertion. The system busy time generation starts with the lowest line number, and goes on for all communication lines which are defined using “Communication Line Definition Interface”. The system busy time generation algorithm is explained in Section 5.2.4. System busy time hour, minute generation process is explained with an example below by using parameters that is shown in “System Busy Time Parameters/Sistem Doluluk Oranı Parametreleri” part of Figure 5.23

Minimum Busy Time Interval : 5 minute

Maximum Busy Time Interval : 15 minute

Total Busy Time in a Day : 200 minute

Minimum Time Between Two Busy Time İntervals : 8 minute

Usable Time in a Day= (19 – (6 + 1)*60 + (60 – 30) + 30 = 780 minute

Usable Free Time in a Day = 780 – 200 = 580 minute

Busy Time Number = ROUND(200/ (15 + 8)) = 9

Max.Free Time Parameter = ROUND(580 / 9) = 64

Generated Sytem Busy Time = RANDOM(5;15+1) = 8 minute

// Time is generated between two numbers.

Generated Sytem Free Time = RANDOM(64) = 35 minute

System Free Hour, Minute = Time Lower Limit + Sytem Free Time

Calculated System’s Free Hour, Minute = 06:30 + 35 minute = 07:05

Calculated System’s Busy Upper Limit Hour, Minute = 07:05 + 8= 07:23

Total Busy Time = Total Busy Time + Sytem Busy Time

Total busy time is zero at the begining.

Total Busy Time = 0 + 8 = 8 minute

After System Busy Time Lower Limit, that is 07:05, and System Busy Time Upper Limit, that is 07:23, of the system busy interval is saved to database; next busy time interval is computed with similar method. After finishing system busy time generation for all lines, all existing records in “T_BRL_MSJ_ZAMAN” table are deleted for new message time data. Message time generation starts with the first message in “T_BRL_KULL_MSJ” table, and goes on all of the messages and each message frequency in which is defined, that is explained Section 5.2.7.. Sending time of a message; hour, minute and second, is generated between the message lower and upper time limits, that is shown in “Message Sending Time Interval/Mesaj Gönderme Saat Aralığı” part of Figure 5.13. It is recorded to the

“T_BRL_MSJ_ZAMAN” table. Line busy time and message time for each line is generated from the line which has lowest line number to highest line number in the application. Sequence diagram of communication line busy time generation is presented in Figure 5.27 and activity diagram of this part is shown in Figure 5.28.

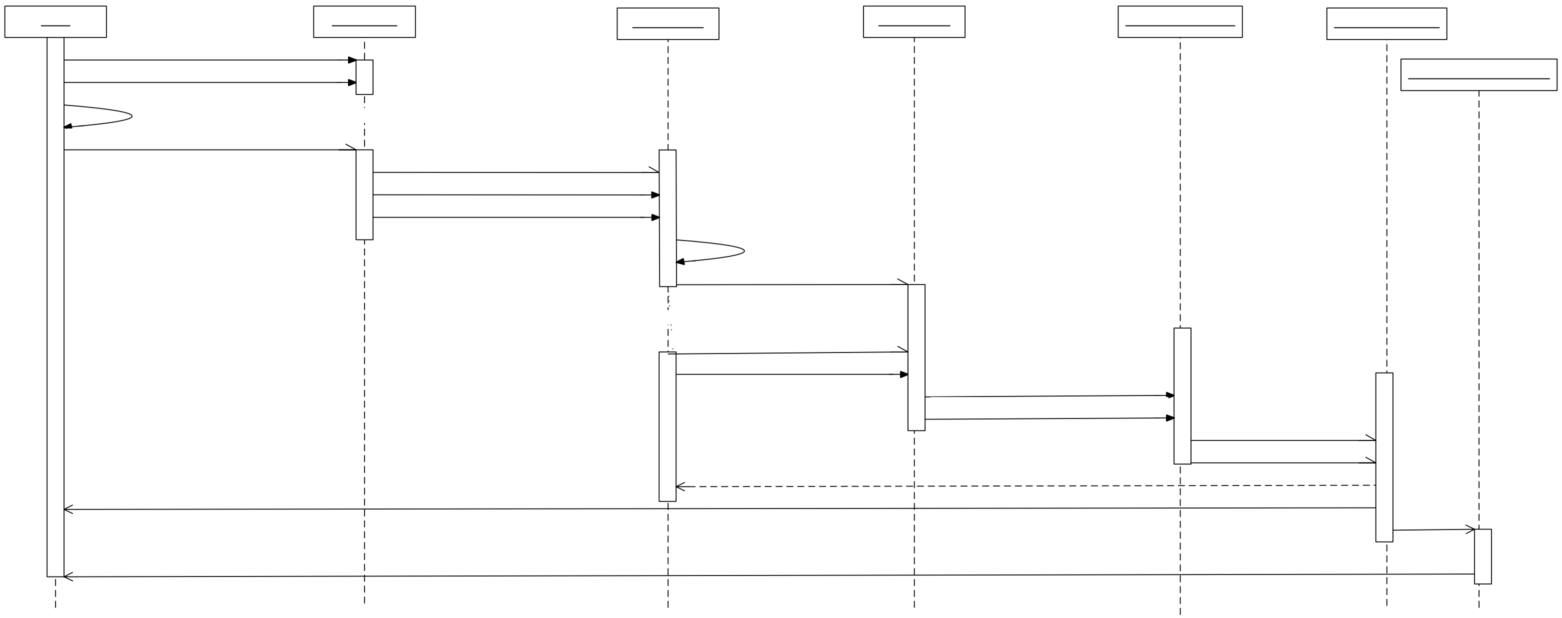


Figure 5.27 : Sequence diagram of time generation

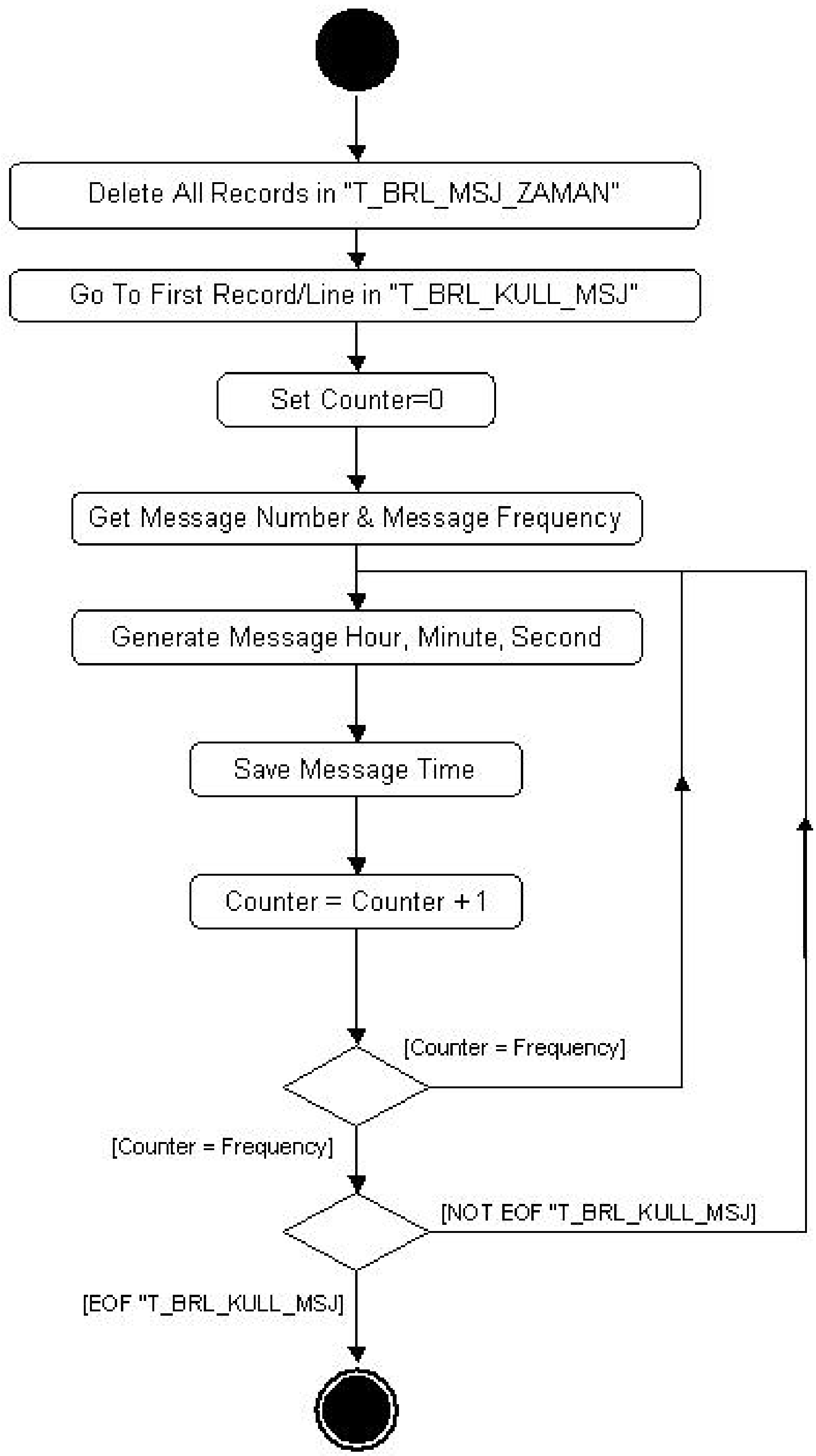


Figure 5.28 : Activity diagram of message sending time generation

5.3.9. Message Transmission Simulation

Eight different sizes for each message type; ***normal, minimum, average, maximum, XML, minimum XML, average XML,*** and ***maximum XML,*** are defined. Message sizes are computed by using the formula which is given earlier in section 5.3.6. Size of the compressed files are obtained experimentally by calculating the proportion of an averaged message file to its compressed form. This figure was obtained in a experimental analysis of XML formatted messages as discussed in section 4.2. In this simulation only Type-A and Type-B messages have been used. Message transmission algorithm is explained in Section 5.2.4. Sequence diagram of message transmission process is shown in Figure 5.29 and Figure 5.30., Activity diagram is provided in Figure 5.31. and process is explained detailed below.

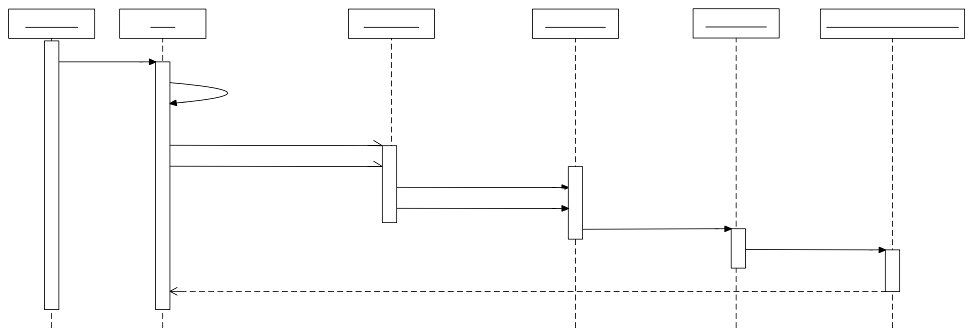


Figure 5.29 : Sequence diagram of starting analysis-1

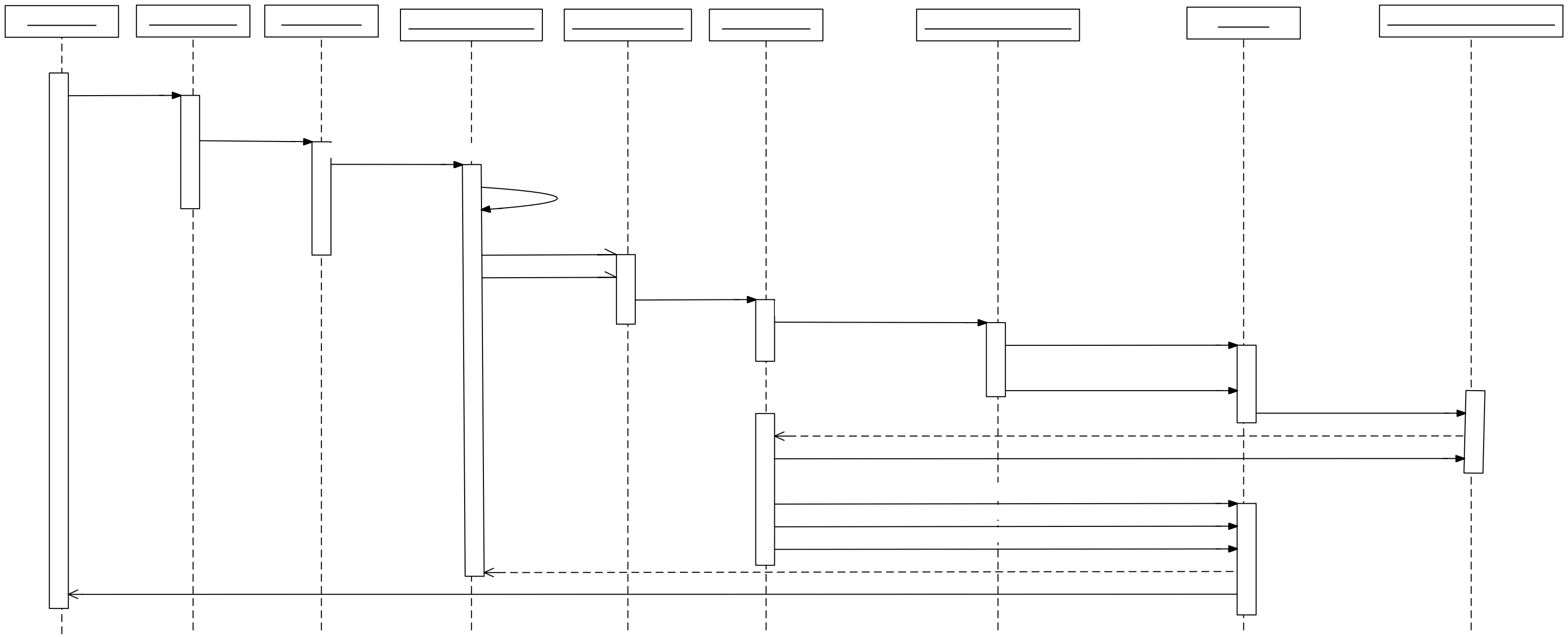


Figure 5.30 : Sequence diagram of starting analysis -2

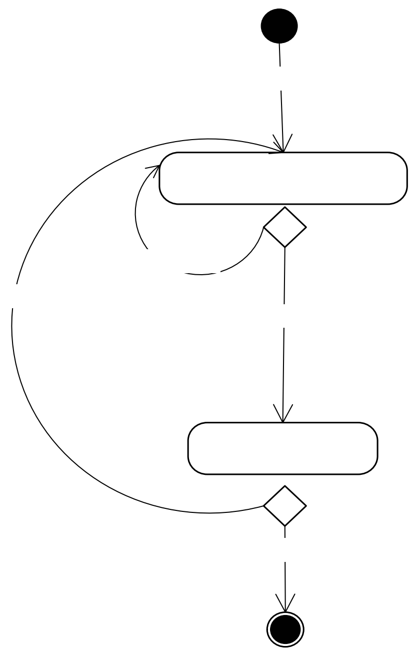


Figure 5.31 : Activity diagram of analysis

First, according to the choosen message, and on which link it will be transmitted, all the link parameters are set. Simulation time (denotes system time) and sent time of the predecessor message (last sent message time) is set as 0:0:0 (hour:minute:milisecond). “Second” part of time parameters is represented as milisecond for sensitive time calculation. Simulation starts with the line which has the lowest line number. The messages are then queued and prioritized. Pseudo code of this part is shown in Figure 5.18. First of all, line busy time and message time are read from “T_HAT_DOLULUK” and “T_BRL_MSJ_ZAMAN” tables respectively. If the message sending time is lower than the line busy starting time, simulation time is checked. If it is less than the current message time, message is marked as “transmitted”. If not, simulation run number, line number, message number, real (original) message sending time, message type, message priority, message file size, bandwidth, waiting time, and the waiting reason parameter (which is set as “**ÖNCEKİ MESAJI BEKLEDİ**”) is recorded to waiting message queue, that is the table, which name is “T_MSJ_BEKLEME”. Queue flag is set as “True”, that shows there is/are message(s) in queue. Line occupation time of the message is computed by the formula which is shown below.

Line Occupation Time = ROUND (Message File Size/ Line’s Bandwidth)

Line Occupation time (in milisecond), and message overhead time (in milisecond) (writing and reading to/from disk) is added to message sending time (when the message was prepared and a send request was made). If it is a compressed file, compression time is also added to the message overhead time for sent time of the last message. Simulation time is set (advanced) to calculated time (sent time of the last message - sum of message transmission hour, minute, second, line occupation time, compression time and message overhead time). Then next message is read. If message sending time is less than simulation time, it is pushed into queue, and the message transmission process, that is explained above, is repeated until the new message time is greater than simulation time. Then, all message(s) in queue are sent, and waiting time is calculated for each.

If the message sending time, that includes calculated new sending time (when it has been waited other messages like above paragraph), is between the lower and upper limits of the line busy time (that is to say, line is busy), current message time is added to waiting message queue, that the “T_MSJ_BEKLEME” table is used for queue, and waiting message flag, that is set when the line is busy, is set as true, that shows there is/are message(s) in queue. All parameters of the message is also recorded in “T_MSJ_BEKLEME_INCELEME” table, the waiting reason parameter is set as “**HAT DOLULUK**”. Simulation time is advanced to line busy ending time, that pseudo code is shown in Figure 5.35. All messages that have sending time which is lower than simulation time (line busy ending time) is pushed into the waiting message queue. Same process is repeated. If the message sending time is greater than simulation time, waiting messages, which are recorded in waiting message queue, are transmitted. Transmission time is calculated for each sent message, and simulation time is advanced last message sent time. Pseudo code of this part is shown in Figure 5.32 through 5.35.

Then next message is read. If the message sending time is greater than line busy end time, it is, new line busy time is read, and all process which is explained above is repeated until all of the messages which belong to that line finish. After that, next line parameters set, and above process begins again until all of the recorded lines are examined.

```
PROCEDURE SEND ALL MESSAGES
    Go To "T_HATLAR"  First Record;
    WHILE NOT  "T_HATLAR"  END of FILE DO
    BEGIN
         READ Line Number and SET v_hat_no;
         IF Ther are messages on Communication Line( v_hat_no) THEN
             REMOVE All Messages in  "T_MSJ_BEKLEME" Table;
             Query Line Busy Time( v_hat_no);
             FIND LINE'S BANDWIDTH(v_hat_no);
             SEND MESSAGES;
                 BEGIN
                     READ First Line Busy Time;
                     LINE'S MESSAGES First;
                     WHILE NOT QUERY Line's Messages END of FILE DO
                           READ Message Time; (PROCEDURE)
                           IF Line is Busy THEN
                              IF There is/are message(s) in waiting message Queue
                                 PUSH Message in Queue (Sending Time< Line Busy
                                                                  Time Starting Hour)
                              SET waiting reason ="HAT_DOLULUK"
                              SAVE Message to "T_MSJ_BEKLEME" and
                                                    "T_MSJ_BEKLEME_INCELEME"
                           ELSE
                                IF There is waiting messages THEN
                                    TRANSMIT Waiting Messages
                                ELSE IF Message Time > Busy Time THEN
                                          SET Current Message as "Transmitted"
                                          SET New Simulation Time
                           END-ELSE-IF;
                           GO TO Next Message;
                     END-WHILE
                 END;
         END-IF
         GO TO Next Commmunication Line;
    END-WHILE
```

Figure 5.32 : Detailed pseudo code of message transmission

```
PROCEDURE  TRANSMIT WAITING MESSAGES
    SET flag := True;
    SET Prior Message Parameter to line busy time ending hour, minute;
    WHILE NOT flag DO
    BEGIN
         READ First Record From Waiting Message Queue table
         IF Line is Busy THEN
            Save Record in “T_MSJ_BEKLEME”;
            flag := False;
         ELSE
             SET Simulation Time/Prior Waiting Message Parameters
    END-WHILE
```

Figure 5.33 : Pseudo code of transmission of waiting messages

```
PROCEDURE   SET Prior Waiting Message Parameters
    SAVE Original Message Time;
    Calculate Prior Message Line Occupation Time;
    Calculate Waiting Time; //Time between original message time and
                                  simulastion time/transmitted prior message
    IF Waiting Time > 0 THEN
        SET “Waiting Reson” Field as “HAT DOLULUK VE ÖNCEKİ
                                            MESAJI BEKLEDİ”;
        SAVE RECORD to “ T_QUEUE_INCELEME” Table;
    END-IF;
    SET   New Simulation Time
```

Figure 5.34 : Pseudo code of waiting message sending

```
PROCEDURE   SET Simulation Time
    SAVE Original Message Time;
    Find Previous Message Time;
    SET Parameters;
    SET new message := False;
    Calculate Waiting Time;
    IF Waiting Time > 0 THEN
        SET "Bekleme Nedeni" Field as "ÖNCEKİ MESAJI BEKLEDİ";
        SAVE RECORD to " T_QUEUE_INCELEME" Table;
    END-IF;
   SET   New Prior Message Parameter
```

Figure 5.35 : pseudo code of setting simulation time

This process, which is explained above paragraphs, is repeated for all types of messages for one simulation run. It can be done by using interface which is shown in Figure 5.36.

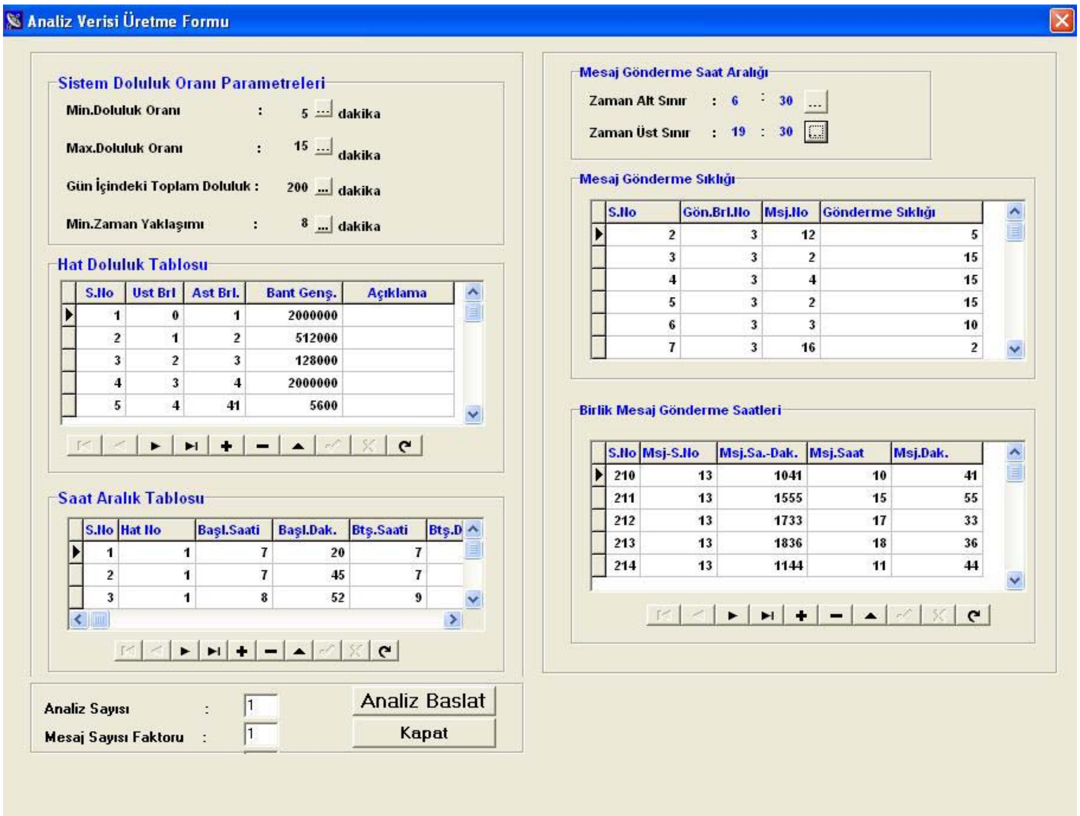


Figure 5.36 : Main analysis ınterface

To obtain an analysis data of a simulation run, below steps are done in order. When “Analiz Başlat” button is pressed, in which is shown in Figure 5.23, it is repeated as many times as determined in analysis number/“Analiz Sayısı”:

1. Determination of transmission, compression and overhead time,
2. Generating line busy time,
3. Generating message transmission time,
4. Transmiting “normal” size messages,
5. Transmiting “normal compressed” size messages,
6. Transmiting “XML” size messages,
7. Transmiting “XML compressed” size messages,
8. Transmiting “minimum” size messages,
9. Transmiting “minimum compressed” size messages,
10. Transmiting “minimum XML” size messages,
11. Transmiting “minimum XML compressed” size messages,
12. Transmiting “average” size messages,
13. Transmiting “average compressed” size messages,
14. Transmiting “average XML” size messages,
15. Transmiting “average XML compressed” size messages,
16. Transmiting “maximum” size messages,
17. Transmiting “maximum compressed” size messages,
18. Transmiting “maximum XML” size messages,
19. Transmiting “maximum XML compressed” size message,
20. Preparation of data for analysis,
21. Preparation of comparison graphics.

There are three parameters usedf for measuring communication performance. Simulation tool is run as “Simulation Run/Analiz Sayısı” times. 8 simulation runs with increasing numbers of messages (8 times mor efor the 8th run) data is used for graphical analysis, but this analysis module has been run more than 500 times[30].

Parameters of message number is used to increase the number of transmitted message. Each "Message Sending Time Interval/Mesaj Gönderme Sıklığı" is multiplied by parameter. Message number parameter is set as "1" to "8". This shows us how message number effects system performance and waiting time.

5.4. Analysis and Comparison Graphics

In this part all analysis and graphics are based on the message traffics on the communication line numbered 5. 5th line's bandwidth is 5600 bits per second. (Bandwidths of the lines that are between companies and battalions are between 4800 and 5600 bits per second.) Each graphic is drawn by simulation program gathering data from the database. 116 messages are sent on the 5th line. On the average, 65 messages are sent on each line. Total number of the messages which are sent on all the lines that belong to the simulated brigade is 1712. A custom comparison program has been developed by TLF and it has been used for calculating the comparison times and ratios of formatted military messages. The suffix of compressed files is cab. Compressed files are demostrated in graphics with their name and the "cab" suffix. (eg: minimum.cab for minimum size compressed files)

This part discusses some of the decisions that were made during the implementation stage. To test the system to its limits (lines' bandwidth), a stress test has been performed by multiplying the number of messages number that are sent on the communication lines. All file types are compared one by one and various combinations. Vertical axis shows waiting times (as miliseconds) for each type of message. Horizantal axis represents message number multiplier, that is the number of generated messages for each line is multiplied by a message number multiplier. Each simulation run and waiting time (in miliseconds) are analyzed for all communication lines. Waiting times are displayed for each simulation run for all four file types as given in Figure 5.37. Comparison graphics

[30] Runs with more messages are not reported due to the database engine performance degradation and readability of graphics.

of their XML counterparts are illustrated in Figure 5.38. When the XML counterparts of the file which has maximum size is added, there is a big impact/change in the graphics. Line traffics' comparison graphs for an average AdatP-3 formattd file against an average XML formatted file, which is shown in Figure 5.39, how the waiting times of individual messages are affected as the number of messages increases. From Figure 5.40 and Figure 5.43, it is seen that a maximum size message will experience much longer waiting times when it is formatted using XML, and it grows exponentially with increasing number of messages. This shows us that employing compression to reduce XML formatted messages. It has been verified in Figure 5.42 by using compressed XML formatted messages, the waiting times have been reduced from exponential to a linear growth as the number of messages increases. Waiting times comparison for each type of message types are given in Figure 5.37 through Figure 5.43.

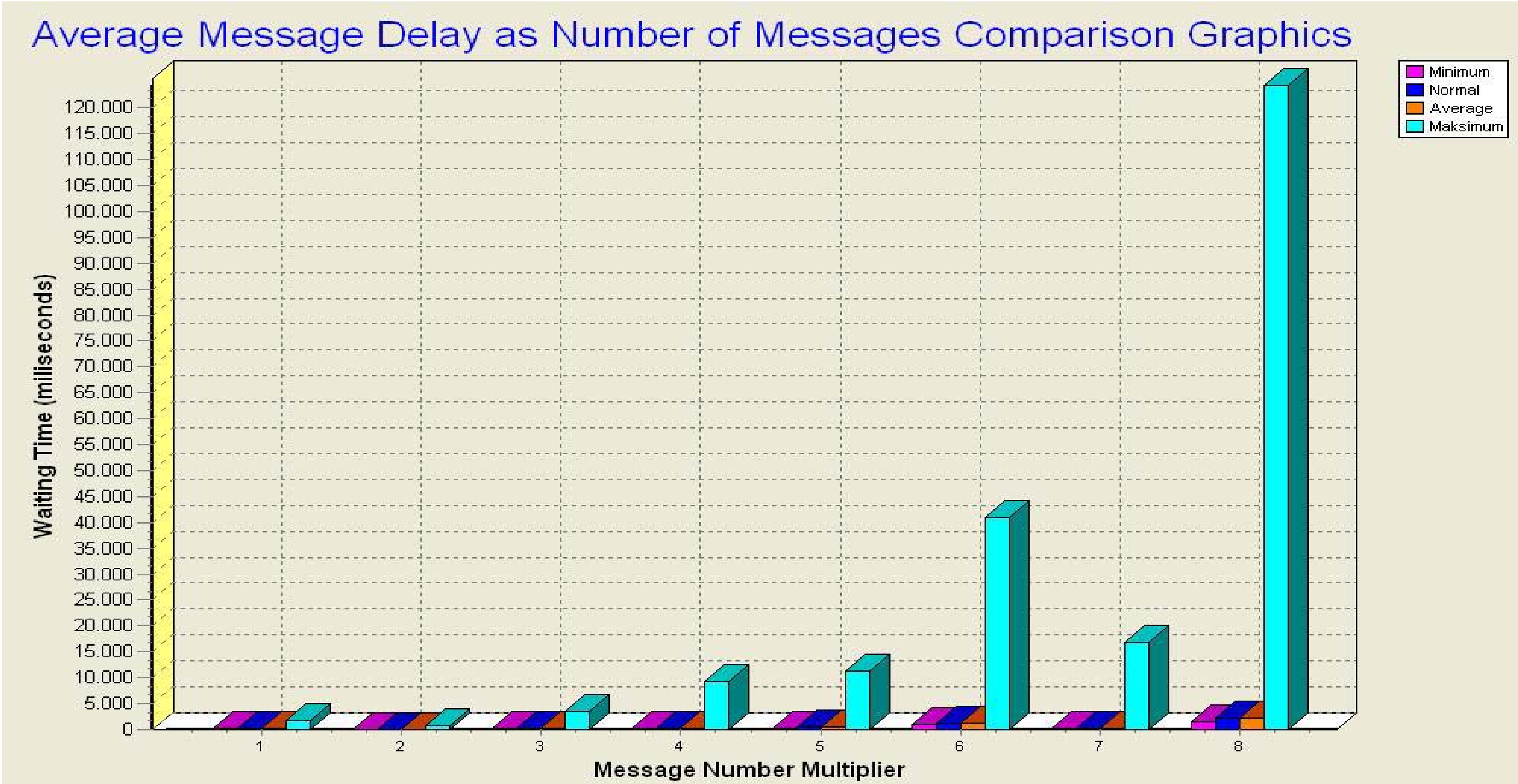


Figure 5.37 : Average message delay as number of messages increases (Minimum, normal, average and maximum file types)

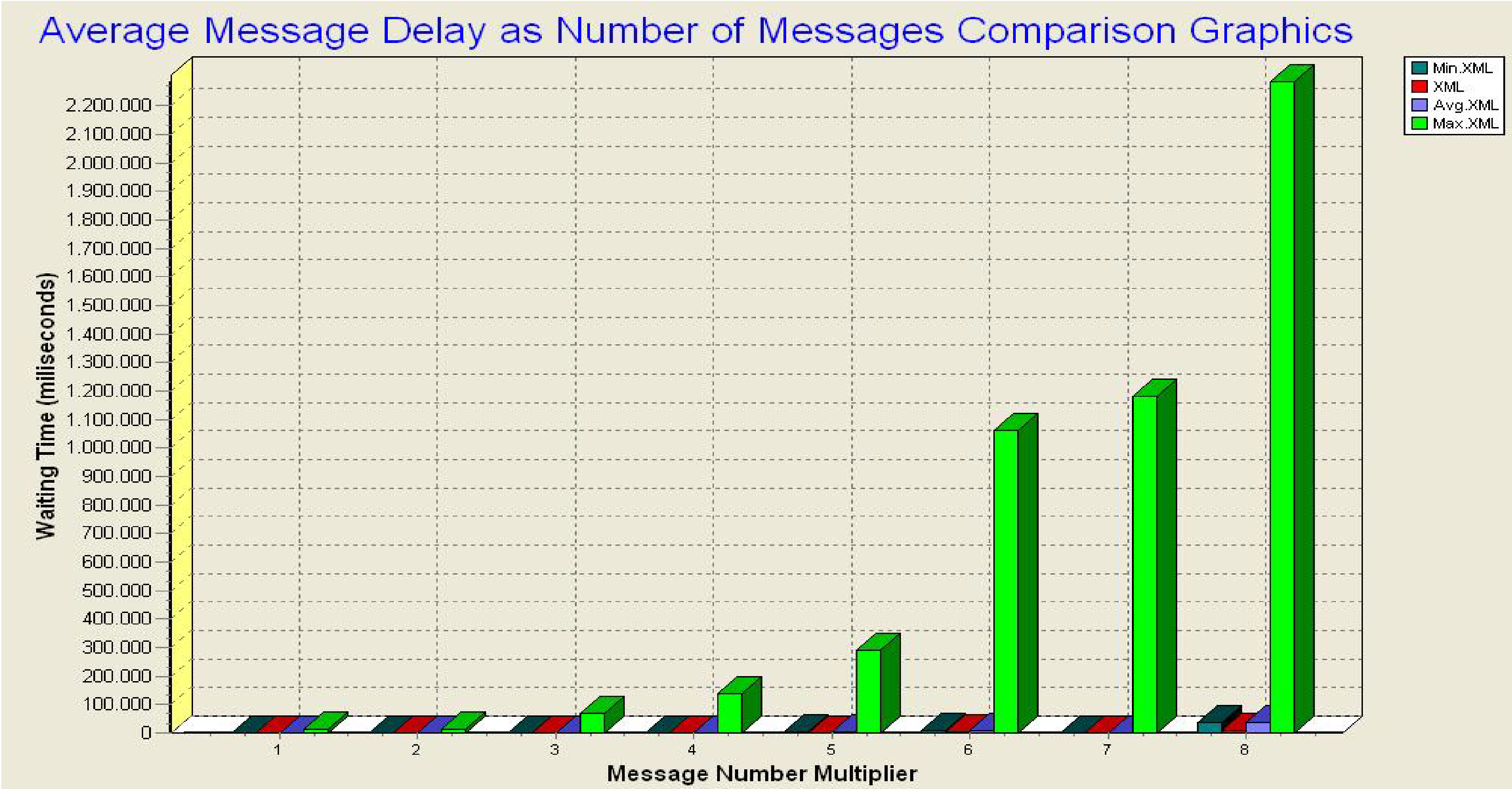


Figure 5.38: Average message delay as number of messages increases (Minimum XML, XML, average XML and maximum XML file types)

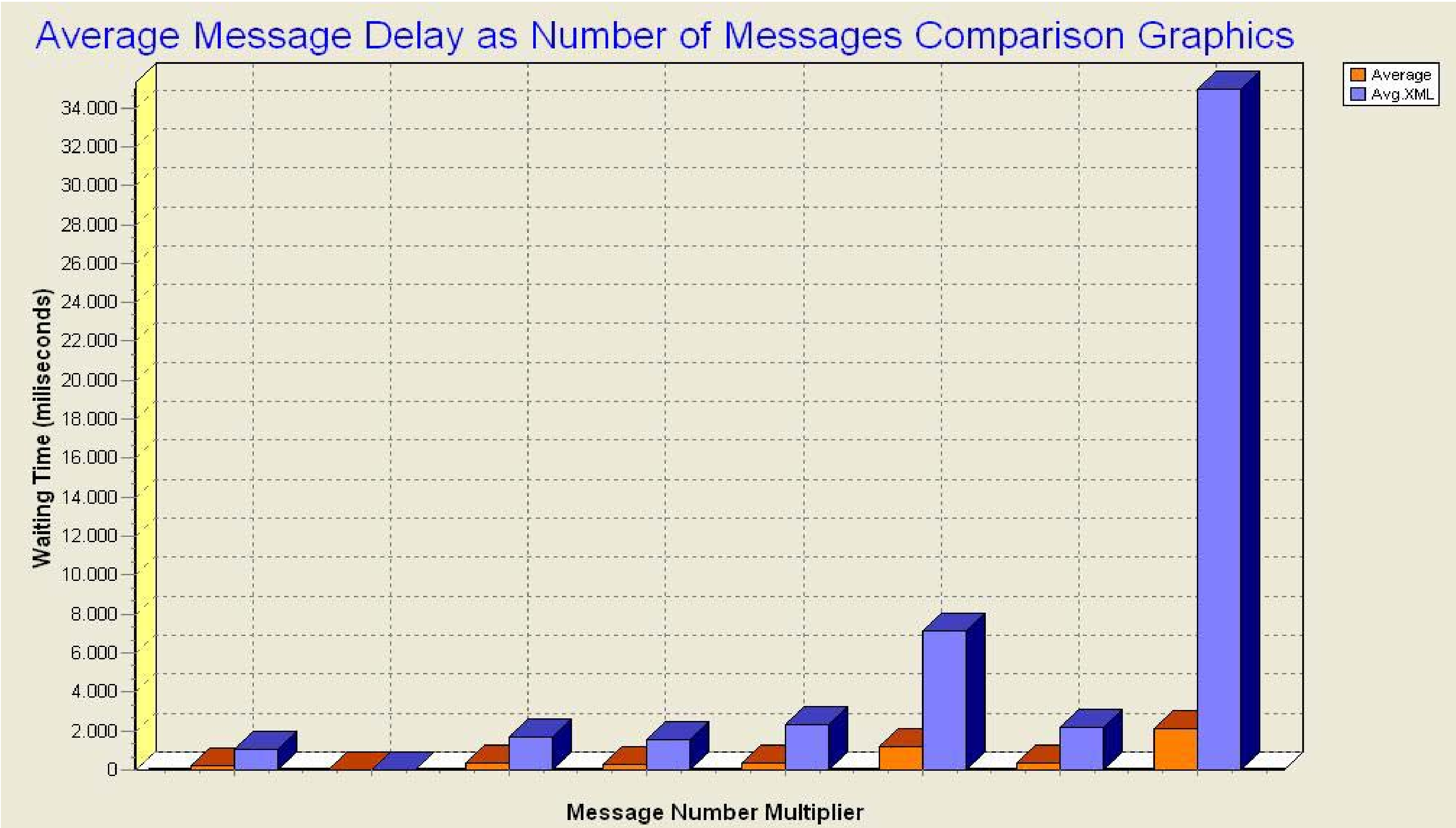


Figure 5.39 : Average message delay as number of messages increases (Average vs. average XML file types)

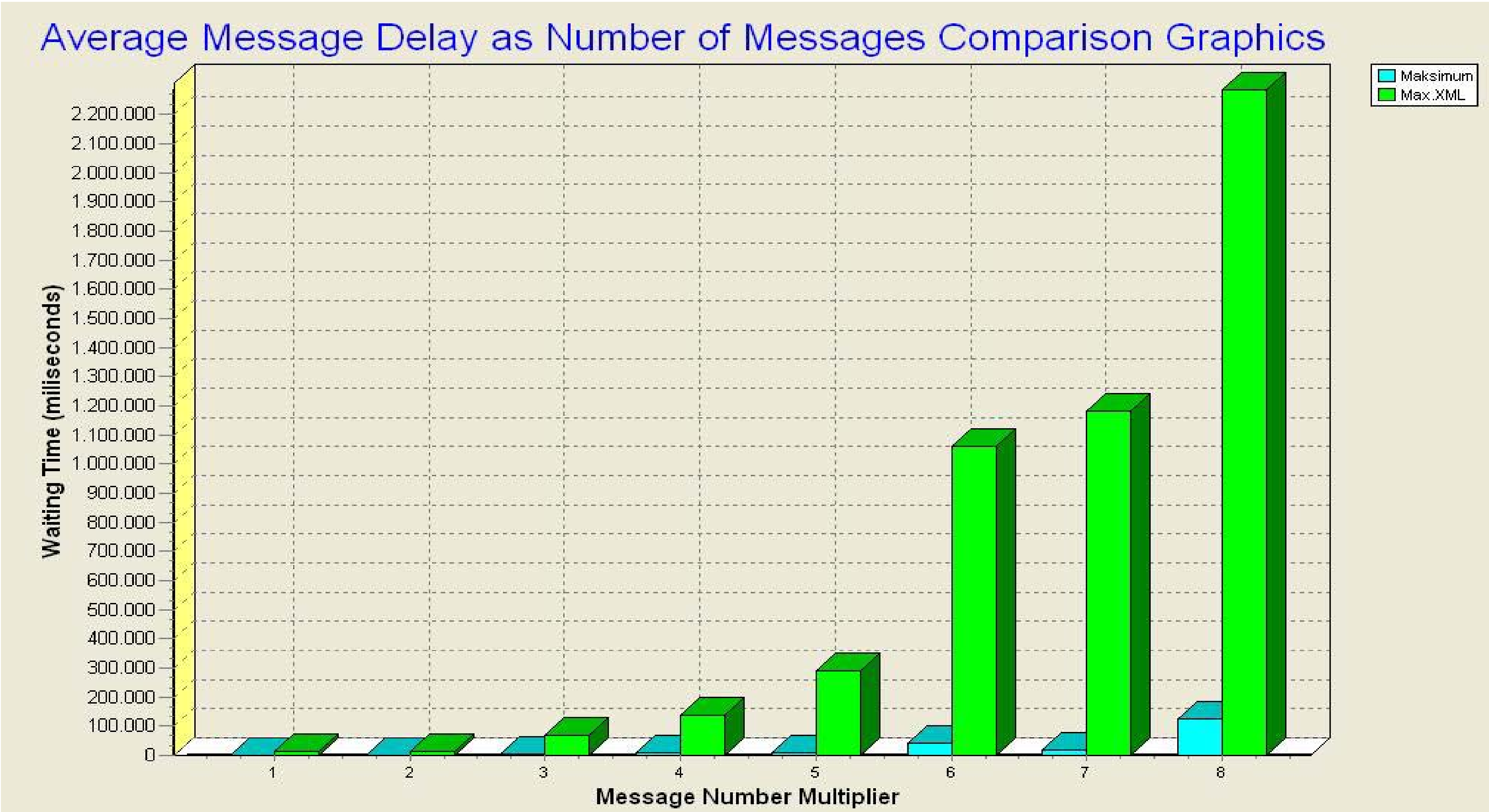


Figure 5.40 : Average message delay as number of messages increases (Maximum vs. maximum XML file types)

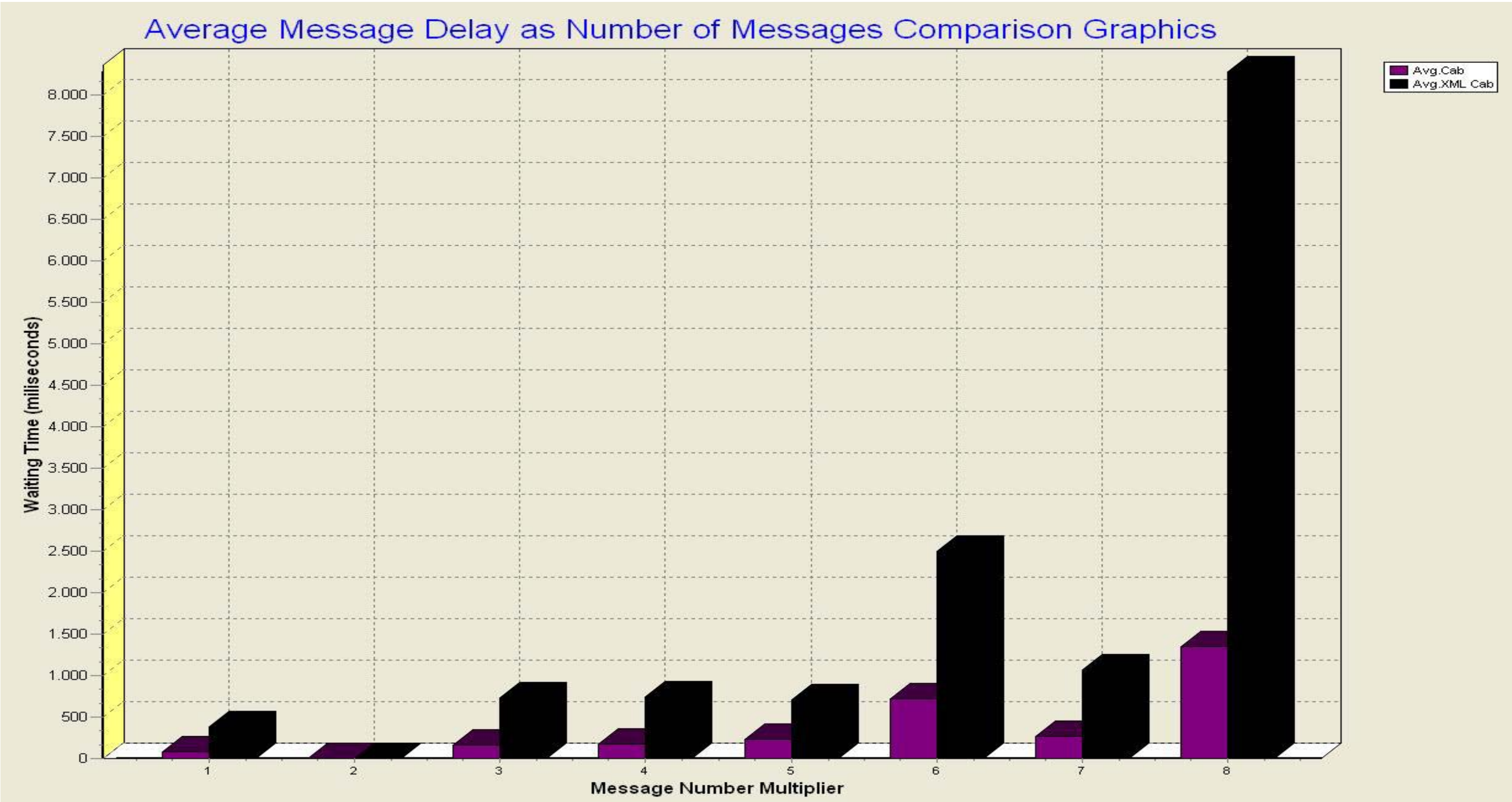


Figure 5.41 : Average message delay as number of messages increases (Compressed average vs. compressed average XML file types)

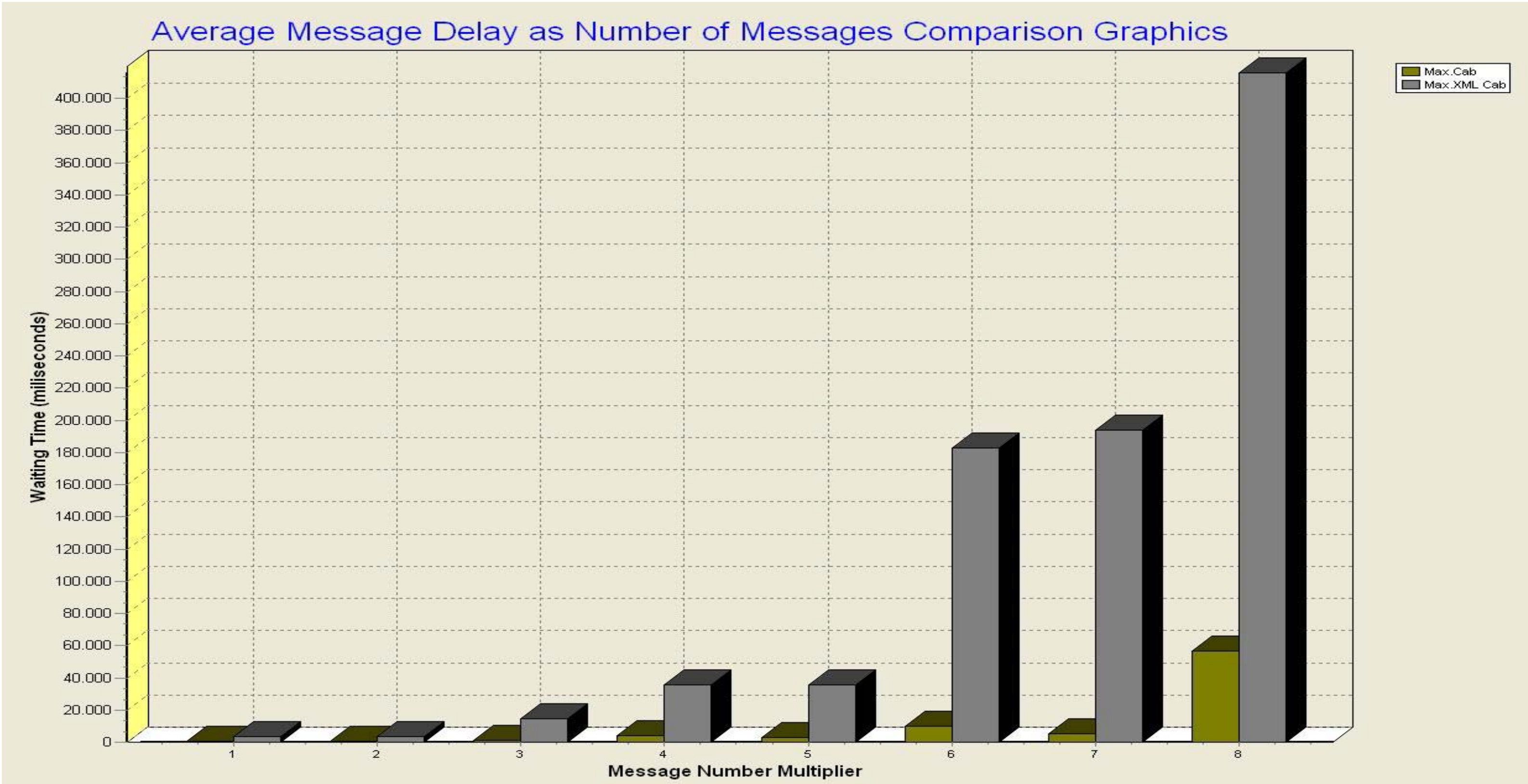


Figure 5.42 : Average message delay as number of messages increases (Compressed maximum vs. compressed). maximum XML file types)

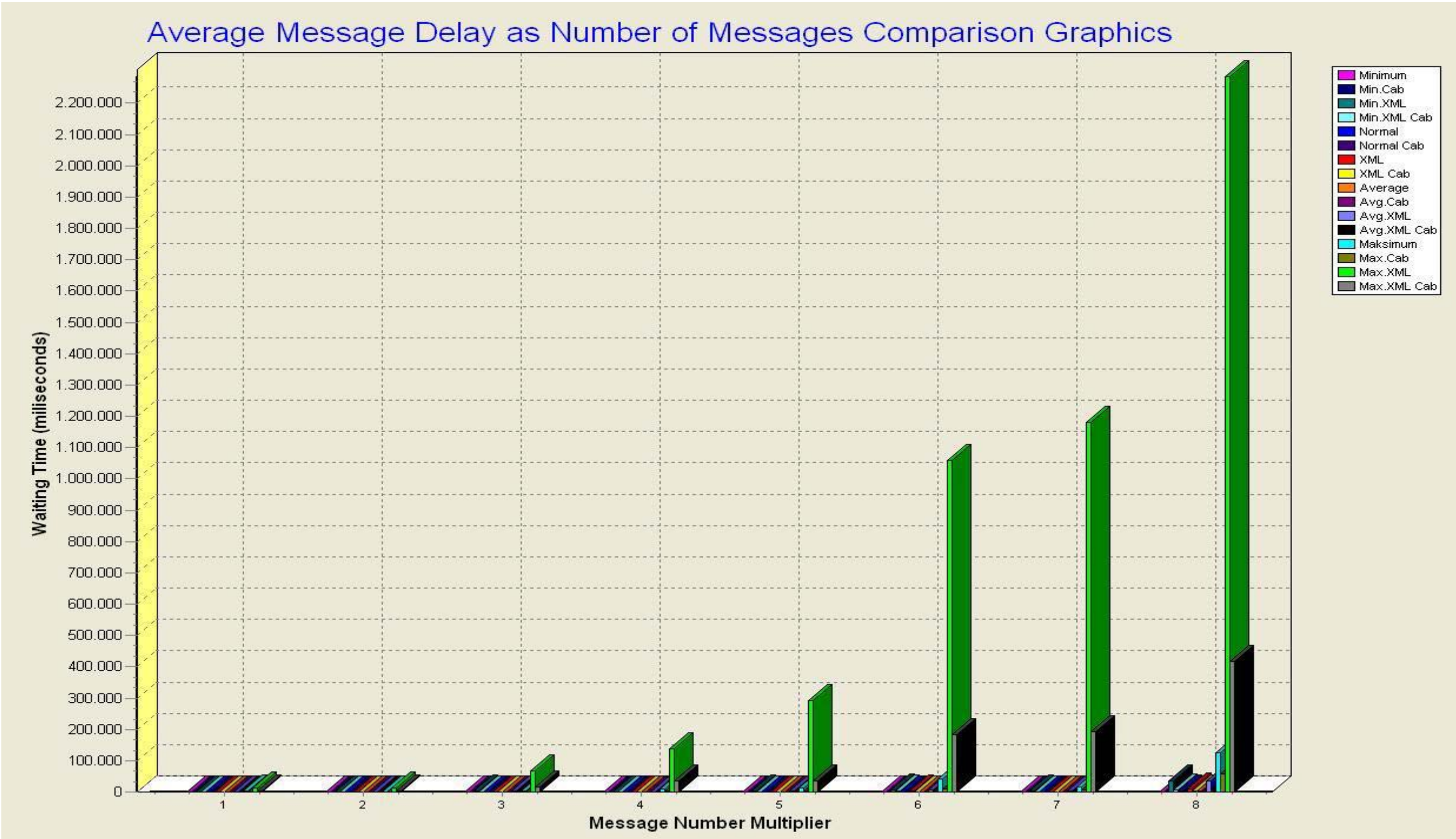


Figure 5.43 : Average message delay as number of messages increases (All file types)

Average waiting message numbers for each type of files are compared in Figure 5.44 through 5.46. There are not big differences among the waiting message numbers of all file types, maximum size and their counterparts excluded. Comparison graphs of lines' occupation ratios of all file types are illustrated in Figure 5.47 through Figure 5.52.

XML substantially increases the size of these files compared to the same data is represented in its AdatP-3 format. There is a lot of replicated text in an XML document, as illusturated in Figure 4.2. XML documents are therefore a prime candidate for compression. Compression increases the ***entropy*** of the compressed text. The benefits of compression rate and compression time are minimising storage space requirements, reduction in transmission bandwidth requirements and main memory required for processing the message. There are disadvantages caused by compression. Because the data is compacted, corruption of one byte could cause the entire message to be lost. Figure 5.44 through Figure 5.53 show that there are no big advantages gained from compression of minimum, normal and average size file. Compression generates some advantage for maximum and maximum size XML formatted messages. Line utilizations ratios that are presented in Table 5.2 are very small. They are actually less than 1 %. This means XML verbosity is negligible and XML is appropriate for MTF for current means of communication.

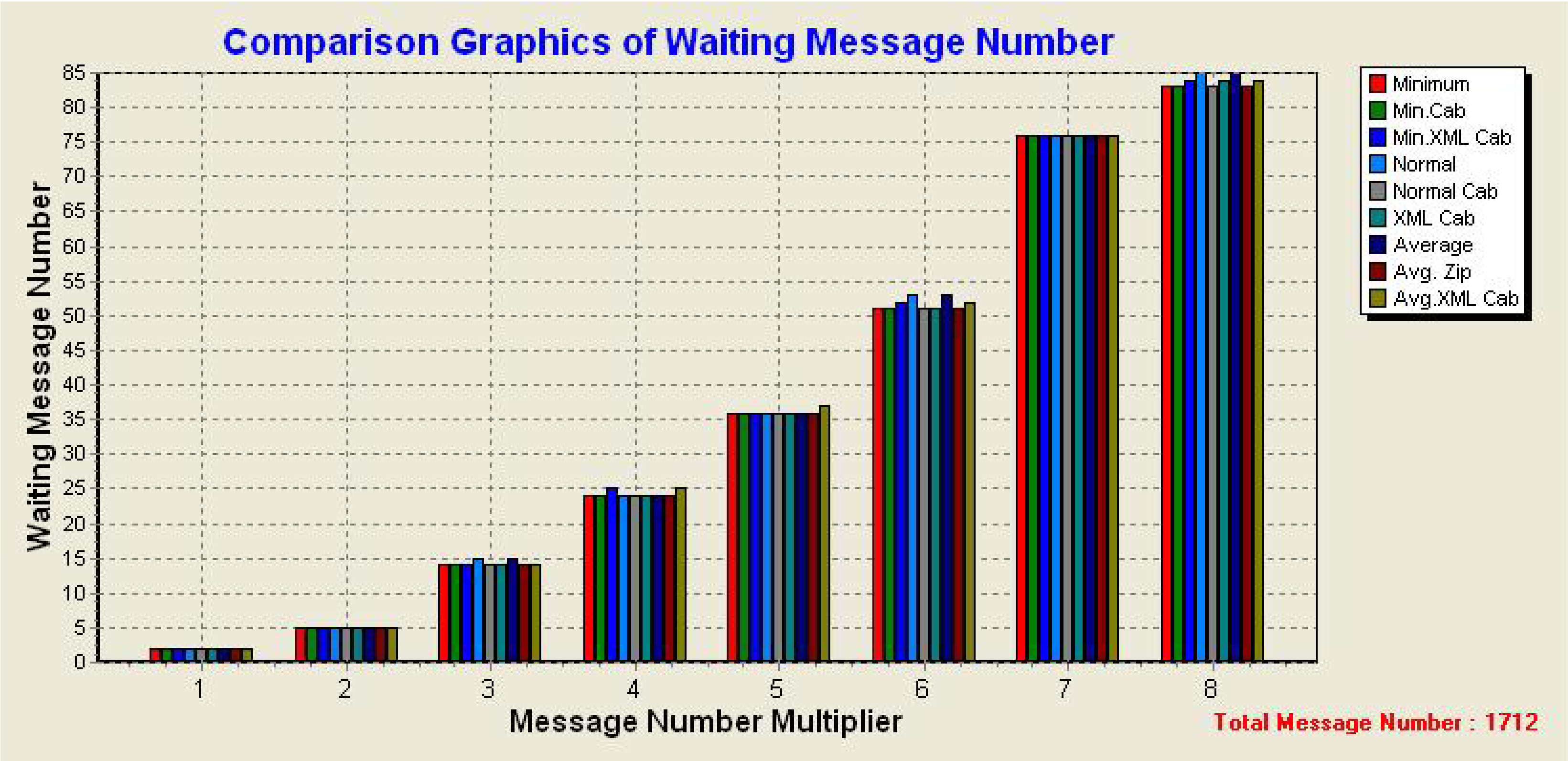


Figure 5.44 : Waiting message number comparison graphics (All file types, maximum and counterparts excluded)

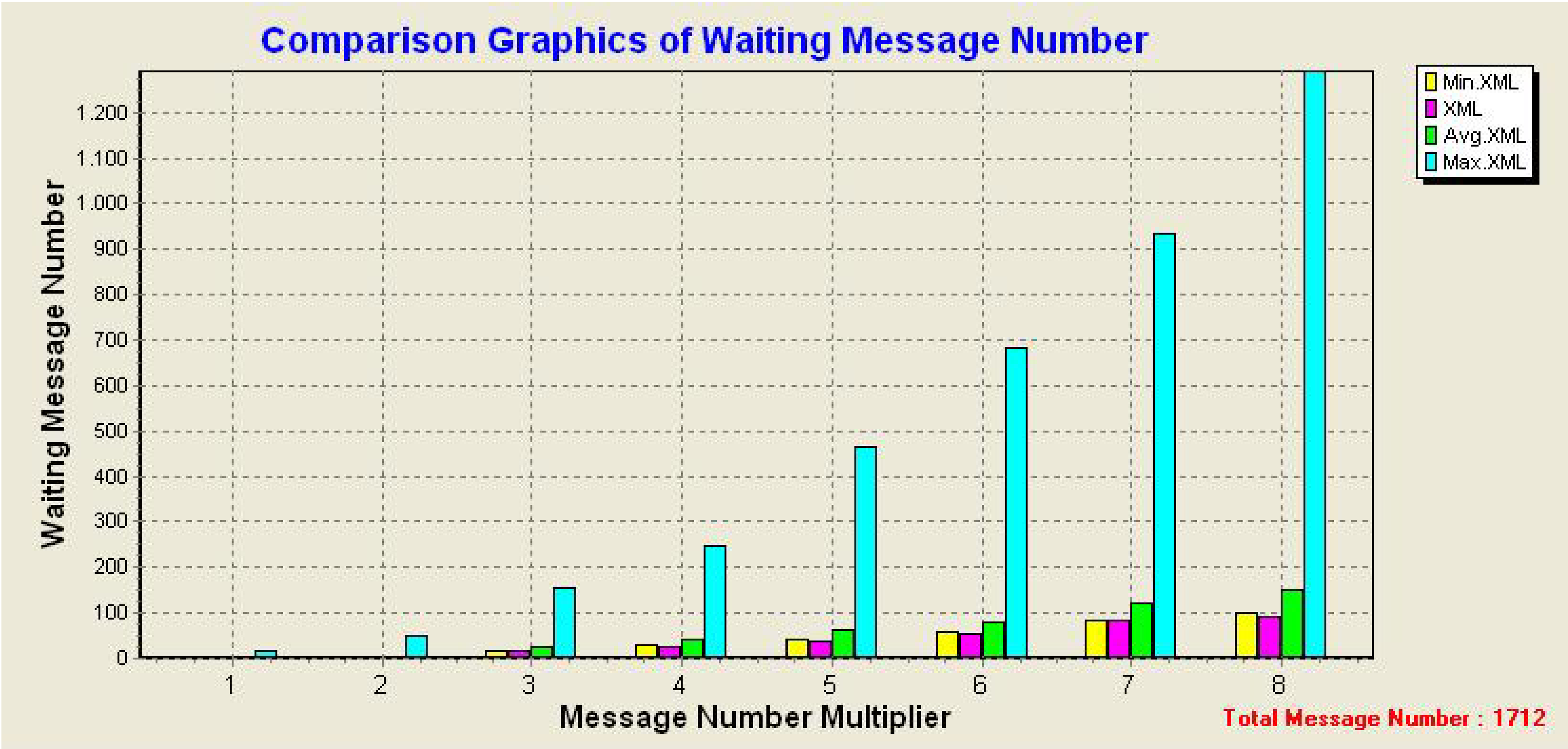


Figure 5.45 : Waiting message number comparison graphics (Minimum XML, XML, average XML and maximum XML file types)

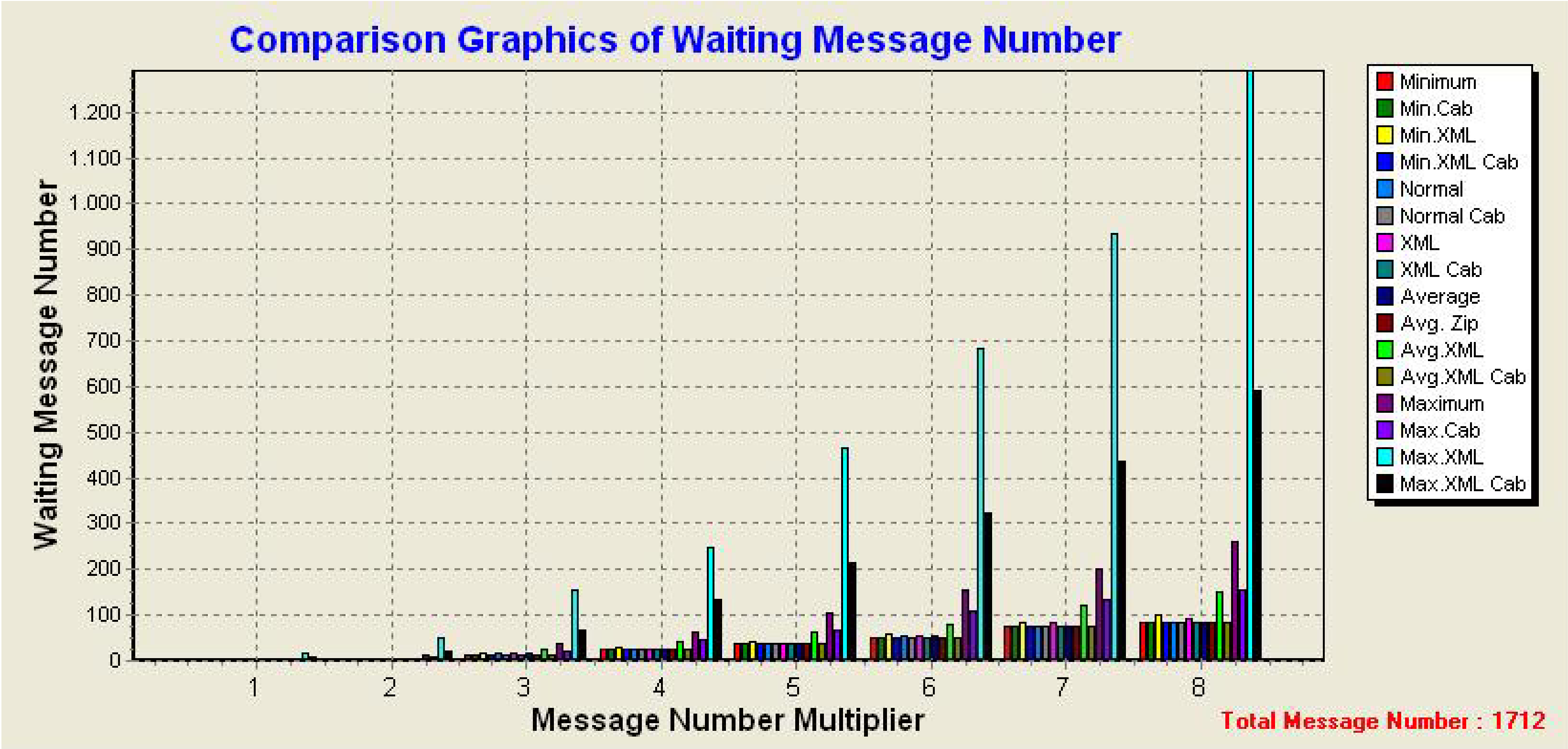


Figure 5.46 : Waiting message number comparison graphics (All file types)

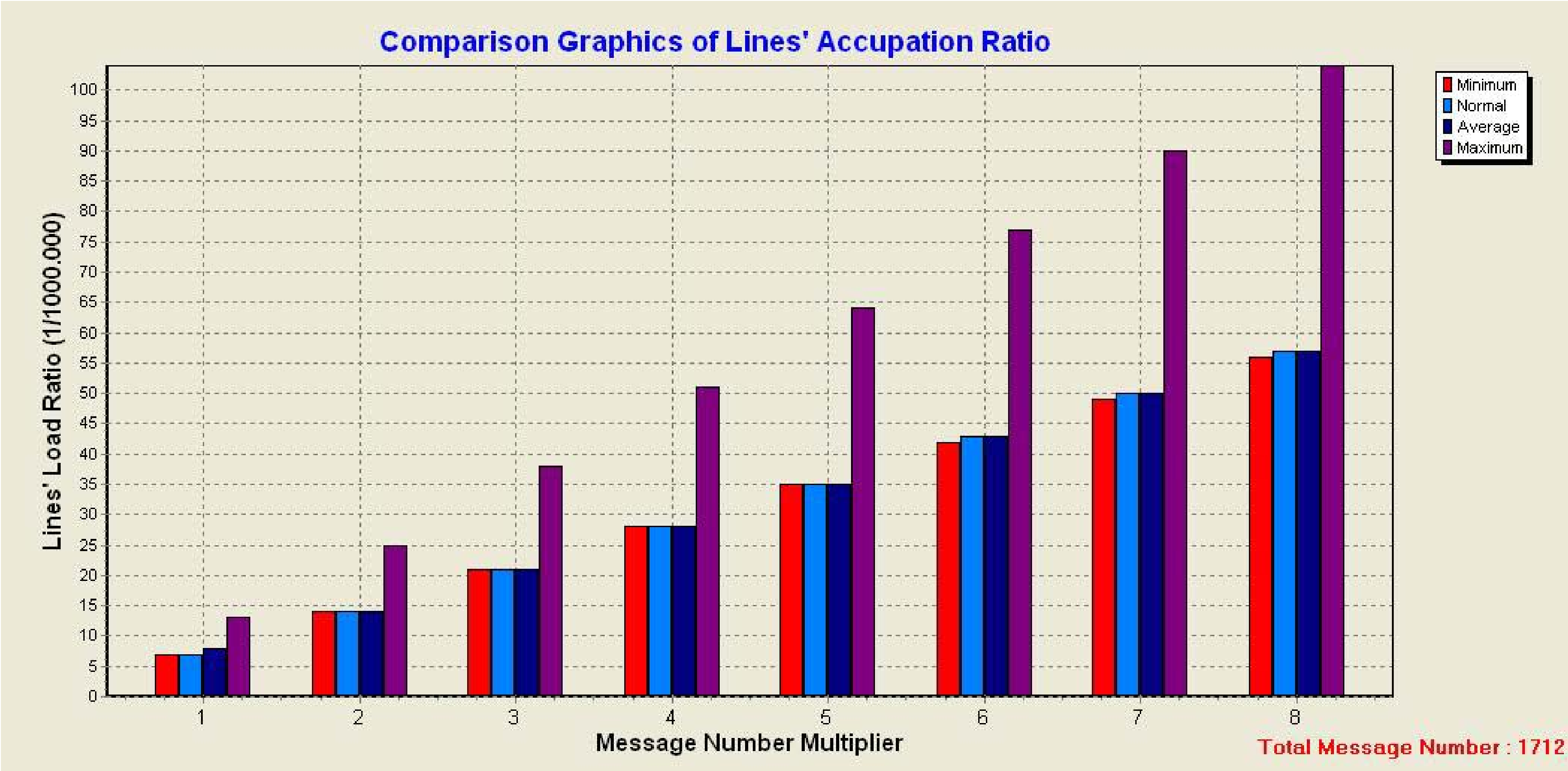


Figure 5.47 : Lines' occupation ratios (Minimum, normal, average, maximum file types)

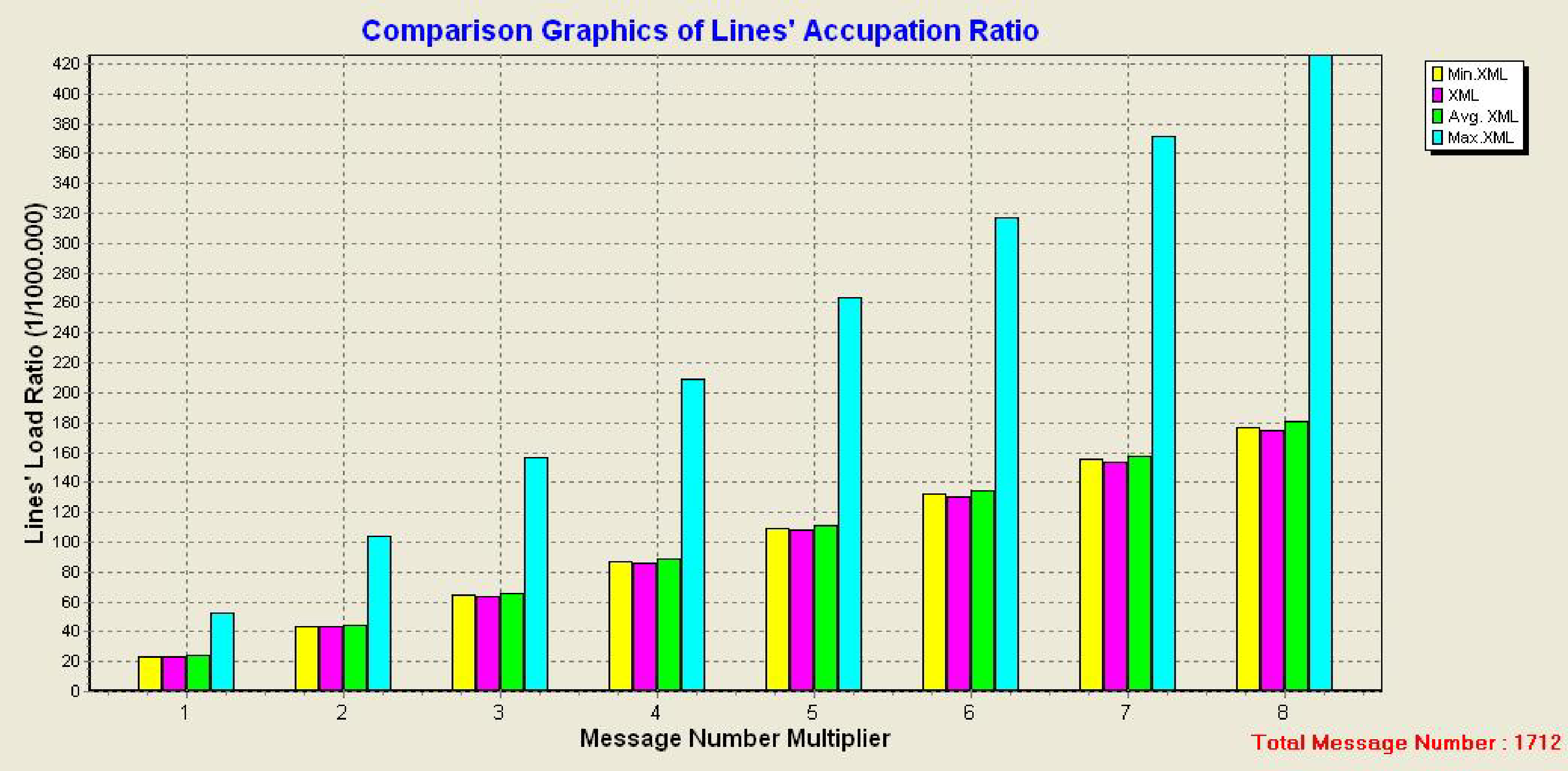


Figure 5.48 : Lines' occupation ratios (Minimum XML, XML, average XML and maximum XML file types)

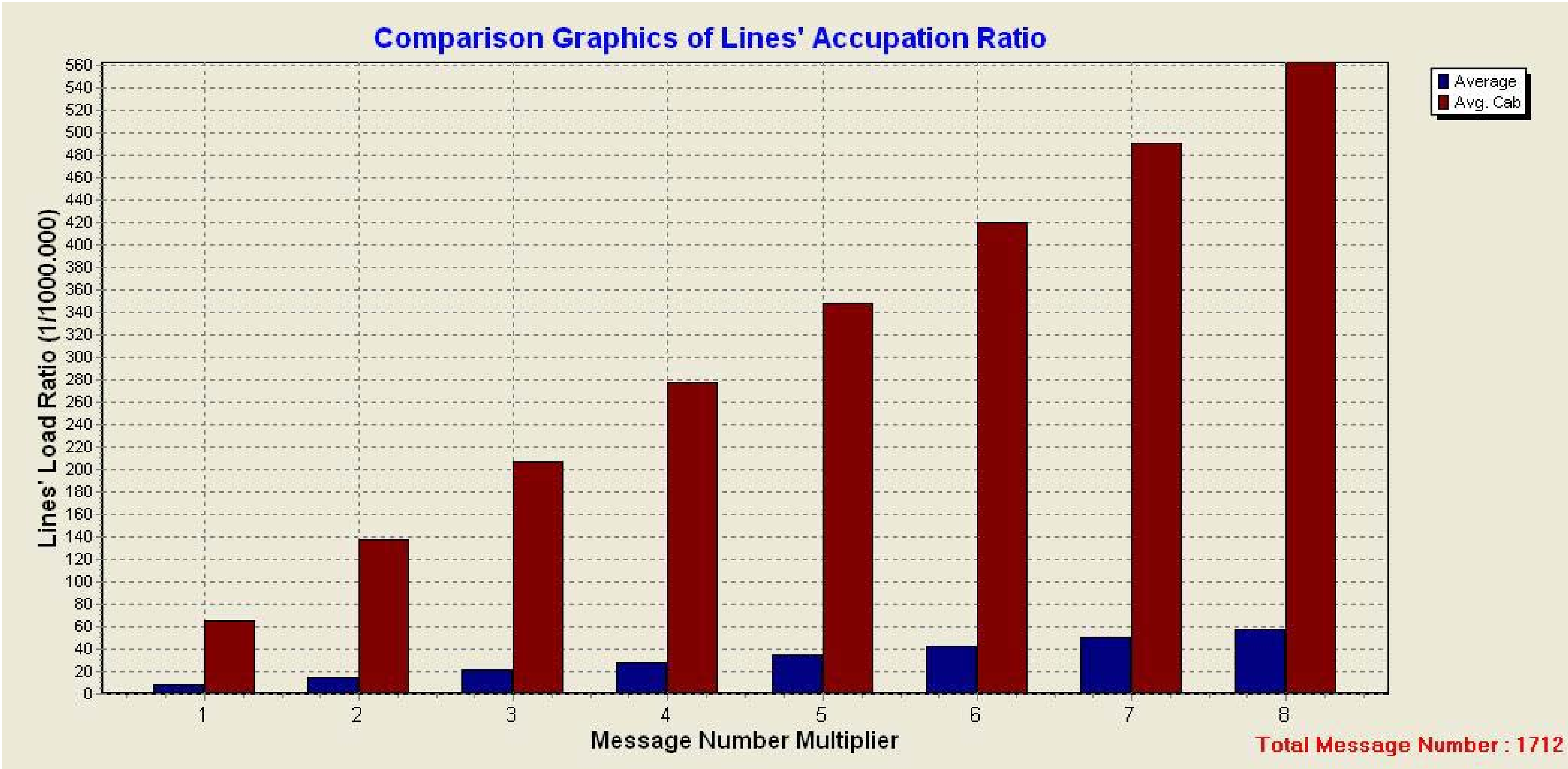


Figure 5.49 : Lines' occupation ratios (Average vs. compressed average file types)

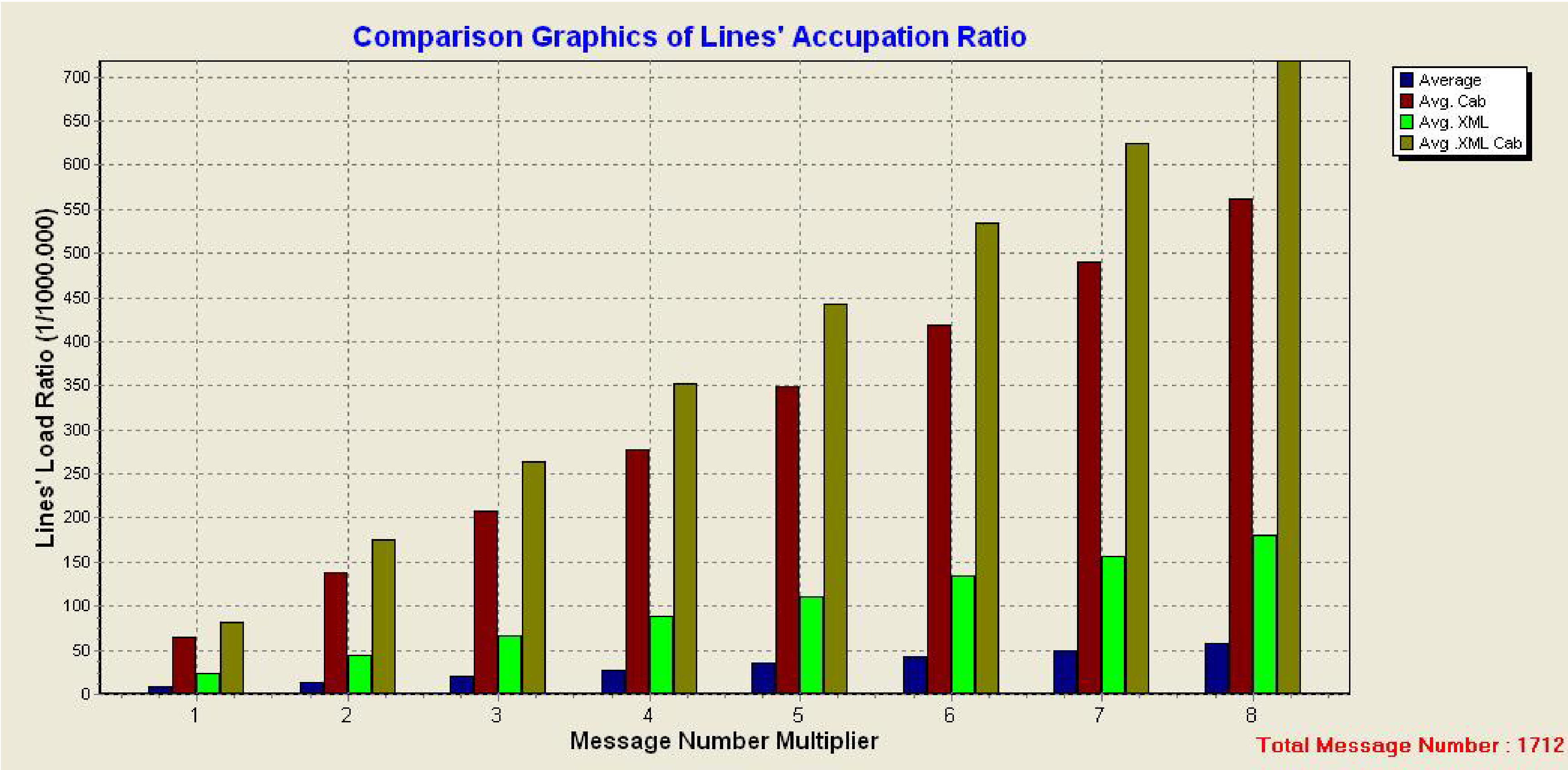


Figure 5.50 : Lines' occupation ratios (Average, average XML and their compressed counterparts)

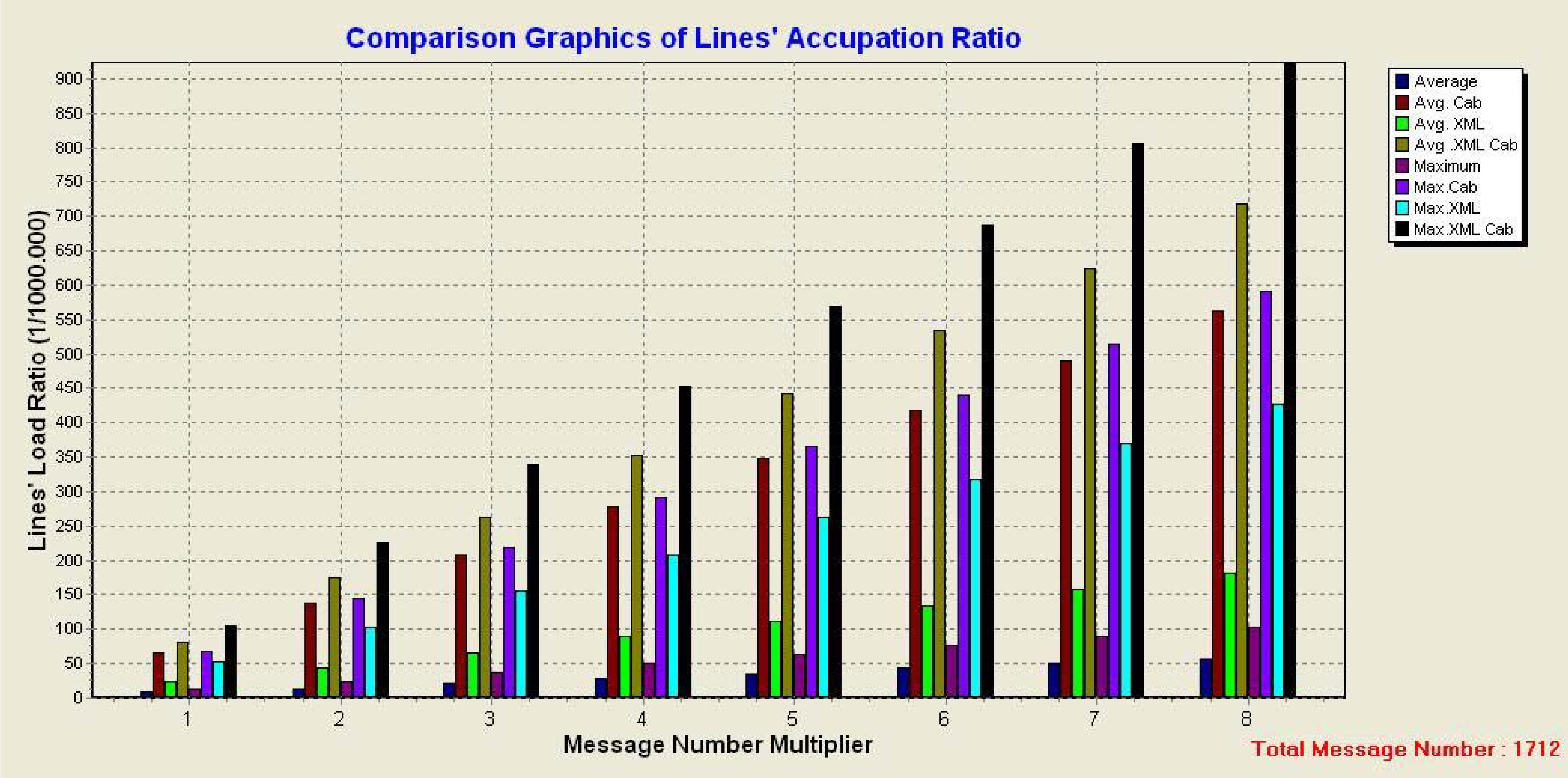


Figure 5.51 : Lines' occupation ratios (Average, maximum, and their XML counterparts; and compressed counterparts of all)

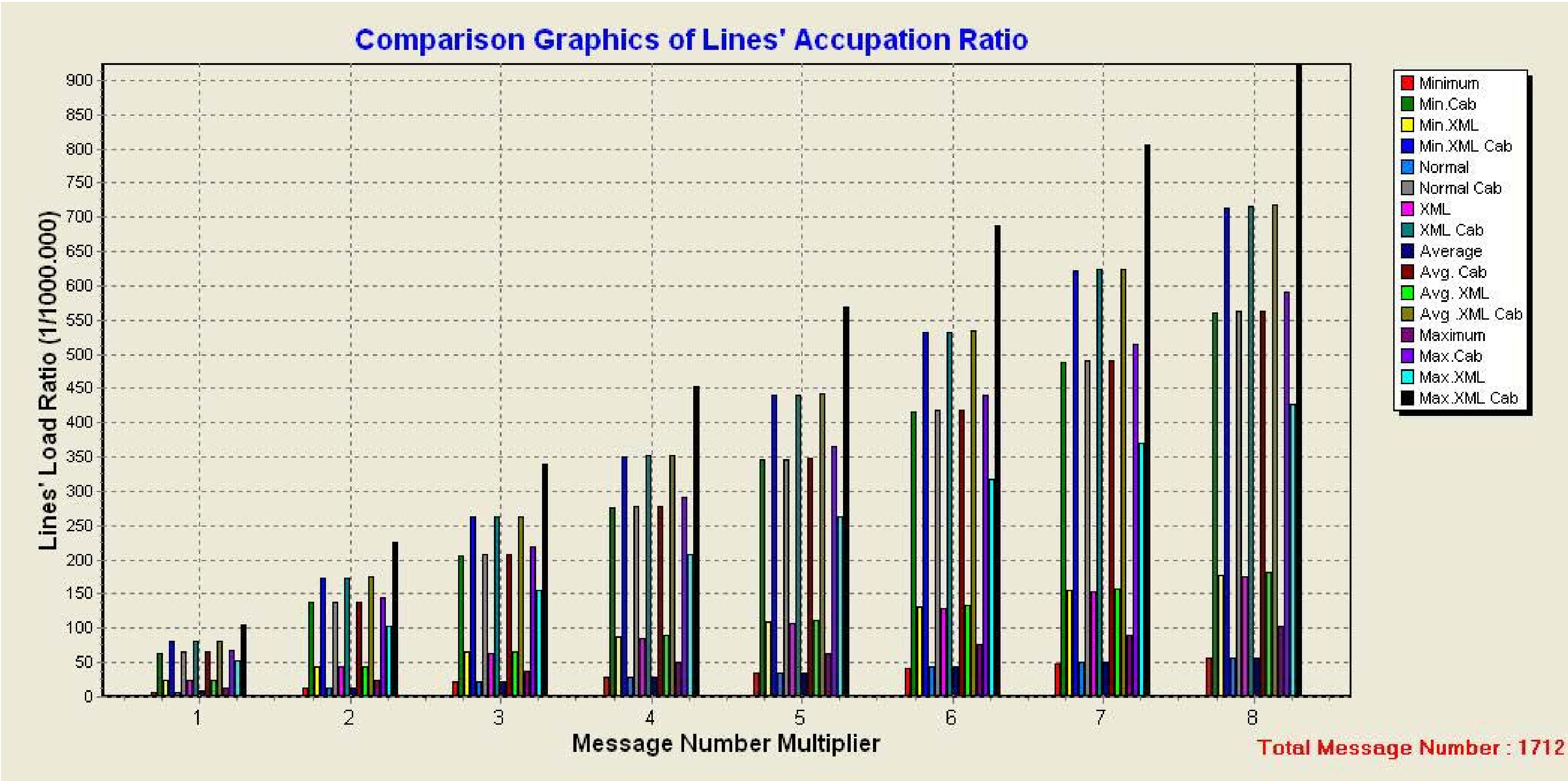


Figure 5.52 : Lines' occupation ratios (All file types)

Table 5.2 : Lines' Occupation Ratios (1/1,000,000)

Message Number Multiplie	Minimum		Min.XML		Average		Ort.XML		Normal		XML		Maximum		Max.XML	
	Min.	Min.Cab	Min.XML	Min.XML Cab	Avg.	Avg. Cab	Avg.XML	Avg.XML Cab	Normal	Normal Cab	XML	XML Cab	Max.	Max.Cab	Max.XML	Max.Xml Cab
1	7	64	23	82	8	65	24	82	7	65	23	82	13	68	52	106
2	14	137	43	174	14	137	44	175	14	137	43	174	25	144	104	225
3	21	206	65	262	21	207	66	263	21	207	64	263	38	218	156	339
4	28	276	87	351	28	277	89	353	28	277	86	352	51	291	209	454
5	35	346	109	441	35	348	111	443	35	347	108	441	64	365	263	570
6	42	417	132	531	43	419	134	534	43	418	130	532	77	440	317	687
7	49	488	155	622	50	490	157	625	50	490	153	623	90	515	371	805
8	56	560	177	714	57	562	181	718	57	562	175	715	104	591	426	924

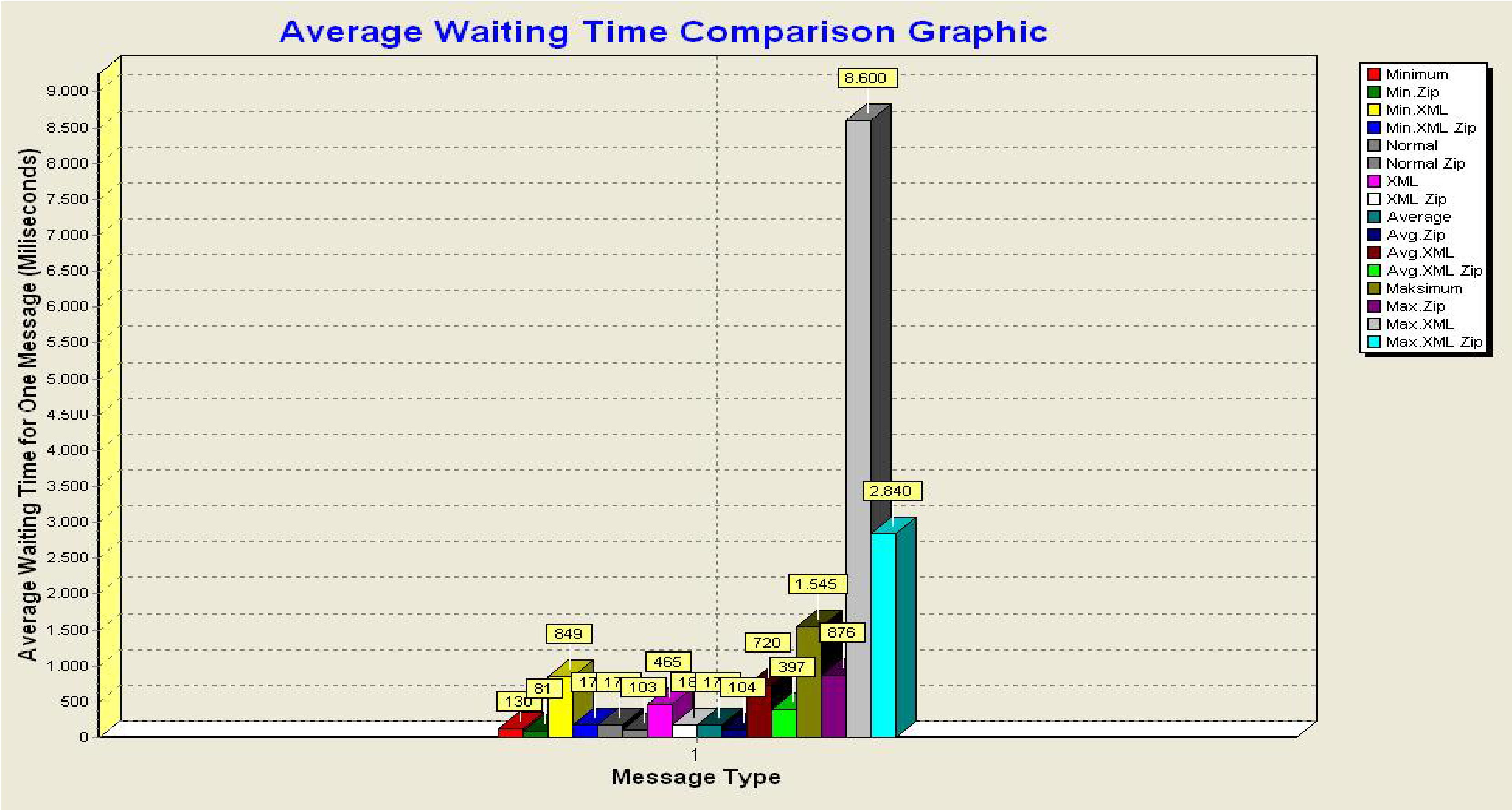


Figure 5.53: Comparison graphics of average waiting time for one message

CHAPTER 6

CONCLUSION

Collecting, transfering and use of timely battlefield information is called as "digitization". It will enhance "situational awareness" of decision making commanders and improve their performance. "Situational awareness" is greatly provided by digitization. It will provide a clear picture and decrease decision-making time and errors as the flow of information can be optimized. Collecting, processing and decision making on information to achieve superiority in all the military operations is mandatory to enable, enhance, and protect our force's ability in continuous military operations within the military information environment. Accessibility of information that the commanders need, awareness of that information, and ultimately the timely delivery of that information can be achieved by information management. Effective information management is especially important for decision makers who have to respond to dynamic situations.

XML has an amazing rate of momentum since September of 1996 when the development of XML specification began. XML as a format has a lot of nice properties. It is a perfectly general way of representing arbitrary data structures. XML allows developers to easily describe and deliver rich, structured data from any application in a standard, consistent way. But XML is verbose. There is a lot of replicated data in an XML document. It substantially increases the size of files when the same data is represented in its raw format. Current army wireless technology has limited transmission bandwidth. Large XML files consume substantially more bandwidth. The structure of messages, the optimal network structure with consideration of organization's hierarchical structure needs to be considered and then planned. XML documents are a prime candidate for compression because of verbosity. This study highlights work being done about exposition of requirements for data compression and sufficiency for bandwidth capacity of current Turkish Land Forces (TLF) communication media for XML-MTF in tactical theatre.

Average message size is 1500 bits per second. Maximum size of average type message size is 5500 bits per second. Minimum size of maximum type message size is 5000 bits per second, and maximum size of maximum type message size is 20,000 bits per second. Average XML conversion factor used in this study is 4(3 for message header + 1 for message body). Possible message types and their attributes that effect the occupation of a data line/link bandwidth have been defined. How much bandwidth of a link is used by ADatP-3 formatted version of a message and XML representation of the same file, and their compressed form has been investigated. Message times and waiting times have been generated randomly for each line., but waiting times of the maximum, maximum XML size files and their compressed counterparts increase exponentially and considerably higher than other file types. Lines' occupation ratios of all file types, that is presented in Table 5.2, are less than %1. This shows that the capacities of currently used communication links are enough to support XML-MTF. Almost always, the size of XML message which is used by modelled brigade (necessary parameters are given above) really makes no difference.

Data compression is widely used in a variety of programming contexts. Classical trade-off between **time** and **space** exists. The benefits of compression rate and compression time are minimising storage space, transmission bandwidth, and main memory requirements for the message. Relative importance of CPU usage, memory usage, channel demands and storage requirement is important to decide using compression or not. ***Much of the time, the size of XML really makes no difference since hard disks are cheap, and the transmission time might be only a small part of the total time in the process.*** But in some applications, bandwidth and storage space can be limited. In some cases it may be faster to read and transmit a compressed message because its size is smaller (even when the compression time is also included) , and the total time taken by compressing and transmitting a message can be less than spring and tranmitting a non-compressed XML formatted message. But at other times, bandwidth and storage space can be limited so a more time consuming but more powerful compression method can be used.

XML is human readable (more or less). **XML's future will depend on efficiency.** Its flexibility will enable integration with existing technologies as a data exchange format that can be manipulated by internal processes or through external display. The simplicity, flexibility generality, versatility, exchangeability and low development cost of XML will make it attractive in today’s environment. XML, will be an enhanced alternative to current methods of information exchange.[46] It is an important family of new technologies which deserve the attention of TLF, and even Turkish Army, but it should be applied in a controlled manner. Compression serves significant advantage for maximum and maximum XML size files. Additional steps should be taken in the short term to ensure that the potential of long-term benefits offered by the introduction of XML are fully exploited.

Future Work :

Current TLF messaging technology is based on manual message processing. Messages sent on a communication line are independent from other messages sent on other communication lines. They are not automatically processed and combined into one message (as it is done in real life) to send superior or subordinate units. Message flow on lines is independent from usually each other. TLF C2IS projects are new and evolving. Because of uncertainties about fully automated messaging among TLF Units, imaginary parameters, but near to actual, are used in this study. The tool does not model any actual message traffic flow for TLF. To use results of this study in any real life decision making, significant modelling and extension work will be needed along with complete validation and verification.

XML is not necessarily the best solution for all information exchange applications, but it is an alternative that must be considered. In order to be able to exploit the capabilities of XML, a detailed analysis of the to be exchanged and its purpose is a precondition. With this basis, concepts can be developed which are exactly aimed at the specific requirements. The success of XML depends on how professionally these tasks will be solved in the near future. ***XML will have a place in the future of the web.*** The decisive factor is that XML can be tailored completely to the needs of users, the information they want to exploit and finally the application that will use it. At the same time, the enormous possibilities are a great danger to the success of XML. ***XML is not a solution but rather a tool to develop solutions.*** It is likely that XML will continue to have strong support over the next decade in terms of tools, standards, and users. An XML enabled C4I system would be much more likely easier to integrate with XML-enabled Commercial-Off-The-Shelf (COTS) products, than a C4I system using a proprietary format. Indeed, C4I systems could probably be constructed by XML-enabled COTS products alone. To facilitate interoperability between C4I systems in different organisations (especially between military organisations and non-governmental organisation), XML based messages could be used in the future to exchange information between the systems. [66]

REFERENCES

[1] Cherinka, R.D.,& Miller, R.W., " Using XML-based Web Services to Enhance Information Interoperability in the Department of Defense", XML Conference & Exposition 2001, Florida, December 2001

[2] Alberts, D.S., Garstka, J.J., Hayes, R.E., and Signori, D.A., "Understanding Information Age Warfare", Command and Control Research Program (CCRP), Washington, pp. 15-30, 2001

[3] Holzner, S., "Inside XML", New Riders Publication, pp.1-34, 2001

[4] Clausewitz, C.V. and , Graham, J.J., "On War", N. Trübner, London, 1873

[5] Peach, W., "Message Text Formats - A Solution To The Problem of Interoperability", Journal of Battlefield Technology, Argo Press, Volume 2, Number 1, March 2000,

[6] "Tactical Information Links (TADIL)", Federation of American Scientists Intelligence Resource Program page, URL:http://www.fas.org/irp/program/disseminated/tadil.htm, (Last visited July 2003)

[7] "Digital Data Links-Link 1 (Point-to-Point)/ NATO Standardization Agreement STANAG 5501", NATO Publication, Edition 3, July 1997, (NATO Unclassified)

[8] "Standard Operating Prodedures For Link-1/ NATO AdatP-31", NATO Publication, November 1998, (NATO Unclassified)

[9] "Tactical Data Links-Link 1", Sytasys Systems, Technology & Management Consultants web page, URL:http://www.stasys.co.uk/tdl/tdl8.htm, (Last visited 6 March 2003)

[10] “Tactical Data Link For the Control of Aircraft-Link 4/TADIL-C/ NATO Standardization Agreement STANAG 5504”, NATO Publication, Edition 2, February 1996, (NATO Classified)

[11] “Standard Operating Prodedures For Link-1/ NATO AdatP-4”, NATO Publication, January 1995, (NATO Classified)

[12] “Tactical Data Links-Link 4/TADIL-C”, Sytasys Systems, Technology & Management Consultants web page, URL:http://www.stasys.co.uk/tdl/tdl10.htm (Last visited 3 July 2003)

[13] “Tactical Data Exchange Link11/11B /NATO Standardization Agreement STANAG 5511”, NATO Publication, Edition 2, March 1997, (NATO Unclassified)

[14] “Standard Operating Prodedures For NATO Link-11/11B /NATO AdatP-11”, NATO Publication, April 2001 (NATO Unclassified)

[15] “Tactical Data Links”, ViaSat web page URL:http://www.viasat.com/products/datalinks/index.htm, (Last visited 3 July 2003)

[16] “Tactical Data Links-Link 16”, Sytasys Systems, Technology & Management Consultants web page, URL:http://www.stasys.co.uk/tdl/tdl14.htm (Last visited July 2003)

[17] “Tactical Data Links-Link 16”, Sytasys Systems, Technology & Management Consultants web page, URL:http://www.stasys.co.uk/tdl/tdl4.htm, (Last visited July 2003)

[18] “Tactical Data Exchange - LINK 16/NATO Standardization Agreement STANAG 5516”, NATO Publication, Edition 1, March 1990 (NATO Unclassified)

[19] “Standard Operating Prodedures For NATO Link-16/NATO AdatP-16”, NATO Publication, April 2001, (NATO Unclassified)

[20] “Tactical Data Links-Link 22”, Sytasys Systems, Technology & Management Consultants web page, URL:http://www.stasys.co.uk/tdl/tdl6.htm (Last visited July 2003)

[21] “NATO Improved Link Eleven (NILE)-Link-22/NATO Standardization Agreement STANAG 5522”, NATO Publication, Draft , September 1995

[22] NATO AdatP-22, “Standard Operating Prodedures For NATO Link-22”, NATO Publication, Draft , September 1995 (NATO Unclassified)

[23] Jarrell, R., “Digitization of the Battlefield Using Variable Message Format (VMF)”, Defense Standardization Program Journal, Volume1, Number 2, January/February 2001

[24] "Message Text Formatting Systems, Allied Data Publication 3/ NATO AdatP-3", NATO Publication, Baseline 10, 2001, () (NATO Unclassified)

[25] "NATO Message Text Formatting Systems (FORMETS)/NATO Standardisation Agreement STANAG 5500", NATO Publication, Edition 4, July 1996, (NATO Unclassified)

[26] "United States Message Text Format (USMTF)/Military Standard 6040/MIL STD 6040", US Defence Information Systems Agency, 1999

[27] Chairman of The Joint Chiefs of Staff Manual, "Tactical Command and Control Planning Guidance for Joint Operations, Joint Interface Operational Procedures (JIOP) for Message Text Formats (CJCSM 6120.05)", US Defence Information Systems Agency, 1999

[28] "C4I Technical Architecture", US Defense Information Systems Agency, Version 3.1, Section 4, March 1995

[29] American, British, Canadian, and Australian Armies (ABCA), "Interoperability Technical Architecture", Version 1.1, Section 3, April 1997

[30] "TLF Information Exchange Data Model (TLF IEDM/KKBDVM-01)", Temel Sürüm 1.0, December 2002

[31] TLF Command Control and Information Systems Projects Replication Mechanism, TLF Information Exchange Model (TLF IEM/KKBDM-01), Temel Sürüm 1.0, December 2002

[32] W3C Recommendation, "HTML 4.0 Specification", December 1997

[33] O'Reilly & Associates, Inc, "Rich Text Format (RTF) Version 1.5 Specification", Microsoft Press, Redmond, April 1997

[34] Pardi, W.J., "XML in Action. Web Technology"; Microsoft Press, pp. 20-30, 1999

[35] "Rich Text Format (RTF) Version 1.5 Specification", Microsoft Press, Redmond, Introduction Chapter, April 1997

[36] "Rich Text Format (RTF) Version 1.5 Specification", Microsoft Press, Redmond, RTF Syntax Chapter, April 1997

[37] Maclachlan, M., "XML Will Make Life Simpler", Techweb News, pp. 22-29, March 1999

[38] “Extensible Markup Languages-HTML“,Konsynski, B., Muraski, C., Oaten, A., Ott B., HTML web page, URL:http://www.emory.edu/BUSINESS/et/P98/xml/html.htm, (Last visited July 2003)

[39] Pardi, W.J., “XML in Action. Web Technology”; Microsoft Press, pp. 22, 1999

[40] “Introduction to XML For The Beginner: Understanding Markup Languages”, Devhood web page, URL:http://www.devhood.com/tutorials/tutorial_details.aspx?tutorial_id=597, (Last visited July 2003)

[41] Goldfarb, C.F., “Document Composition Facility Generalized Markup Language: Concepts and Design Guide, Form No. SH20-9188-1”, IBM Corporation, White Plains, 1984

[42] "What is XML?- So XML Is Just Like SGML?”, O’reilly xml.com web page, URL:http://www.xml.com/pub/a/98/10/guide1.html#AEN58, (Last visited July 2003)

[43] “ISO 8879: Information processing-Text and Office Systems-Standard Generalized Markup Language (SGML)”, ISO publication, Geneva, 1986,

[44] “Do you Need XML? A Checklist…”, URL:http://www.arbortext.com/Think_Tank/XML_Resources/XMLfor_Managers/xml_for_managers.html, Abortex Think Tank web page, (Last visited July 2003)

[45] “Extensible Markup Language (XML) 1.0”, W3C Recommendations, February 1998,

[46] Norman, P., “A Study of eXtensible Markup Language (XML)”, Decerno AB, Näsby Park, Sweden, pp.14-23, February 1999

[47] Akyokuş, S., “XML ve XML Uygulamaları”, Turkey Information System Summit, İSTANBUL, pp. 6-7, September 2002

[48] Hunter, D., “Beginning XML (Programmer to Programmer)”, Wrox Press Ltd, London, pp.7-28, November, 2001

[49] Chang, D. & Harkey, D., “Client/Server Data Access with Java and XML”, Wiley Press, Second Edition, pp. 458-459, 1998

[50] “Document Type Declaration (DTD)”, W3C Specification, June 1998

[51] “XML Schema Requirements”, W3C Recommendations, February 1999

[52] “XML Namespaces Specification”, W3C Recommendations, January 1999

[53] "Document Object Model (DOM) Specification", W3C Recommendations, Version 1.0, October 1998

[54] "Cascading Style Sheets Level 1 specification (CSS1)", W3C Recommendations, December 1996.

[55] "Cascading Style Sheets Level 2 (CSS2) specification", W3C Recommendations, May 1998.

[56] "ISO/IEC 10179: Information processing-Text and office systems-Document Style Semantics and Specification Language (DSSSL)", ISO Publication, Geneva, 1996,

[57] "Extensible Stylesheet Language Specification (XSL)", W3C Recommendations, Version 1.0, August 1997

[58] "XSL Transformation Language Specification (XLL-T)", W3C Recommendations, Version 1.0, November 1999

[59] "ISO/IEC JTC1/SC18/WG8 N1920 Hypermedia/Time-based Structuring Language (HyTime)", ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) Publication, Second Edition, August 1997

[60] "eXtensible Linking Language Specification (XLL)", W3C Recommendations, Version 1.0, March 1998

[61] "XML Linking Language Specification (XLink)", W3C Recommendations, Version 1.0, June 2001

[62] "XML Pointer Language", W3C Working Draft, 16 August 2002

[63] Gabrich K.A. & Weiss D.B., "J2EE and XML Development", Manning Publication Co., pp. 16, 2002

[64] Neumüller, M., "Compression of XML Data", MSc Thesis in InformationTechnology Systems, University of Strathclyde, September 2001

[65] Blair, W., D., "Who Knows, Wins?' Knowledge Management And Emergıng Army Tactıcal Command And Control Concepts", Journal of Battlefield Technology, Argo Press, Volume 5, Number 2, pp. 56, July 2002

[66] Macleod, I., "Use of Cots Technology In C2 Information Systems: Balancing The Benefits And Risks", Journal of Battlefield Technology, Argo Press, Volume 4, Number 1, March 1998

[67] Hansen, T., J., "XML for Military Applications", Systematic Software Engineering A/S, Danish, pp. 6, June 1999

[68] Widdowson , J.M., “Digitisation of the Battlespace (Land) and The Principles of War”, Journal of The Battlefield Technology, Argo Press, Volume 4, Number 1, March 2000

[69] Elmasri, R., Navathe, S., B., “Fundamentals of Database Systems”, Buschlen/Mowart Fine Arts Ltd., Canada, Third Edition, pp. 41-110, pp. 295-320, 2000

[70] Borland Delphi Enterprise Version 7.0, Borland Software Corporation, 2002

[71] “Firebird™ Relational Database“, Firebird Version 1, March 2002

APPENDIX-A TLF C2IS Sample Project's Reports and Their Periods

NO	REPORT NAME	SECRECY (TS/S/R/U)	PRECEDENCE	CATEGORY	REPORTS PERIOD	PERIOD	WEEKLY	15 DAILY	MONTHLY	$ MONTHLY	4 MONTHLY	YEARLY	OTHER	SENDING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	REGIMENT	BATTALION	COMPANY	PLATOON	RECEIVING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	BATTALION/ INDEPENDENT COMPANY
1	A Type Report	S	P	1	B/G/S								X			X	X	X	X	X				X		X	X	X		
2	B Type Report	S	P	1	B/G/S	X							X			X	X	X						X		X	X	X	X	
3	C Type Report	S	N	3	B/G/S								X			X	X	X						X		X	X	X	X	
4	D Type Report	S	P	1	B/G/S								X			X	X	X	X					X		X	X	X		
5	E Type Report	S	P	3	B/G/S								X			X	X	X	X					X		X	X	X		
6	F Type Report	S	P	2	B/G/S								X			X	X	X	X					X		X	X	X		
7	G Type Report	S	P	1	B/G/S						X					X	X	X	X	Independent	Independent			X		X	X	X	X	
8	H Type Report	S	P	1	B/G/S								X			X	X	X	X	X				X		X	X	X	X	

APPENDIX-A (Continued)

NO	REPORT NAME	SECRECY (TS/S/R/U)	PRECEDENCE	CATEGORY	REPORTS PERIOD	PERIOD	WEEKLY	15 DAILY	MONTHLY	6 MONTHLY	4 MONTHLY	YEARLY	OTHER	SENDING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	REGIMENT	BATTALION	COMPANY	PLATOON	RECEIVING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	BATTALION/ INDEPENDENT COMPANY
9	I Type Report	S	P	1	B/G/S						X					X	X	X	X	X				X		X	X	X	X	
10	J Type Report	S	P	3	B/G/S								X			X	X	X	X	X	X	X	X	X		X	X	X	X	X
11	K Type Report	S	P	3	B/G/S								X			X	X	X						X		X	X			
12	L Type Report	S	P	3	B/G/S								X			X	X	X						X		X	X			
13	M Type Report	S	P	4	B/G/S								X			X	X	X						X		X	X			
14	N Type Report	S	P	1	B/G/S	X										X	X	X						X		X	X	X		
15	O Type Report	S	P	2	B/G/S	X									X									X		X	X	X		

APPENDIX-A (Continued)

NO	REPORT NAME	SECRECY (TS/S/R/U)	PRECEDENCE	CATEGORY	REPORTS PERIOD	PERIOD	WEEKLY	15 DAILY	MONTHLY	3 MONTHLY	4 MONTHLY	YEARLY	OTHER	SENDING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	REGIMENT	BATTALION	COMPANY	PLATOON	RECEIVING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	BATTALION/ INDEPENDENT COMPANY
16	P Type Report	S	P	2	B/G/S				X						X											X				
17	Q Type Report	S	P	1	B/G/S								X			X	X	X	X	X	X			X		X	X	X	X	
18	R Type Report	S	P	1	B/G/S		X						X		X											X	X	X		
19	S Type Report	S	HP/P	1	B/G/S	X									X											X	X	X		
20	T Type Report	S	F	1	B/G/S								X		X											X	X	X		
21	U Type Report	TS/S	HP/P	2	S											X	X		X					X		X	X	X		
22	V Type Report	S	HP	1	G/S								X		AIRFORCE											X	X	X		
23	W Type Report	S	HP/P	1	G/S								X		AIRFORCE											X	X	X		

APPENDIX-A (Continued)

NO	REPORT NAME	SECRECY (TS/S/R/U)	PRECEDENCE	CATEGORY	REPORTS PERIOD	PERIOD	WEEKLY	15 DAILY	MONTHLY	6 MONTHLY	4 MONTHLY	YEARLY	OTHER	SENDING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	REGIMENT	BATTALION	COMPANY	PLATOON	RECEIVING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	BATTALION/ INDEPENDENT COMPANY
24	X Type Report	S	P	1	B/G/S								X									X	X	X		X	X	X	X	X
25	Y Type Report	U	F	1	B/G/S								X						X			X	X	X		X	X	X	X	X
26	Z Type Report	S	P	1	B/G/S				X													X	X	X		X	X	X	X	X
27	AA Type Report	S	P	1	B/G/S								X						X			X		X		X	X	X	X	X
28	AB Type Report	S	P	1	B/G/S								X									X								X
29	AC Type Report	S	P	1	B/G/S								X									X								X
30	AD Type Report	S	P		B/G/S								X									X								X
31	AE Type Report	S	P	1	B/G/S								X								X	X	X						X	X
32	AF Type Report	S	P	1	B								X									X	X	X		X	X	X	X	X
34	AG Type Report	U	HP		B/G/S								X									X	X	X		X	X	X	X	X
35	AH Type Report	S	P	1	B/G/S								X								X	X	X						X	X
36	AI Type Report	S	P	1	B/G/S								X								X	X	X						X	X
37	J Type Report	S	P	1	S								X								X	X	X						X	X
38	AK Type Report	S	P	1	G/S								X								X	X	X						X	X

APPENDIX-A (Continued)

NO	REPORT NAME	SECRECY (TS/S/R/U)	PRECEDENCE	CATEGORY	REPORTS PERIOD	PERIOD	WEEKLY	15 DAILY	MONTHLY	3 MONTHLY	4 MONTHLY	YEARLY	OTHER	SENDING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	REGIMENT	BATTALION	COMPANY	PLATOON	RECEIVING ORGANISATION	GENERAL STAFF	LAND FORCES HEADQUARTER	ARMY	CORPS	BRIGADE	BATTALION/ INDEPENDENT COMPANY
39	AL Type Report	U/S	ZZ	1	G/S								X						X		X	X	X	X		X	X	X	X	X
40	AM Type Report	S	I	1	G/S								X								X	X	X						X	X
41	AN Type Report	U/S	Ç/I	1	B/G/S	X										X	X									X	X	X	X	X
42	AO Type Report	S	Ç/I	1	B/G/S								X			X	X	X	X	X	X				AIRFORCE	X	X	X	X	
43	AP Type Report	S	Ç/I	1	B/G/S								X			X	X	X	X	X	X					X	X	X	X	İHA Brt.
44	AQ Type Report	S	Ç/I	1	B/G/S								X			X	X	X	X	X	X				X, AIRFORCE	X	X	X	X	X

U : Unclassified
R : Restricted
S : Secret
TS : Top Secret

APENDIX-B PHYSICAL DATABASE SCHEMA SCRIPTS

```
CREATE TABLE T_BRL_SNF (
        SNF_NO                    INTEGER        NOT NULL,
        SNF_KISA_ADI              CHAR(25) CHARACTER SET WIN1254 NOT NULL COLLATE WIN1254,
        BRL_SNF_UZUN_ADI CHAR(30) CHARACTER SET WIN1254  COLLATE WIN1254,
        CONSTRAINT T_PK_BRL_SNF_NO PRIMARY KEY (SNF_NO))


CREATE TABLE T_BRL_TUR(
        BRL_TUR_NO          INTEGER        NOT NULL,
        BRL_ADI             CHAR(30) CHARACTER SET WIN1254 NOT NULL COLLATE WIN1254,
        BRL_ADI_UZUN CHAR(30) CHARACTER SET WIN1254  COLLATE WIN1254,
        UST_BRL_NO          INTEGER,
        CONSTRAINT T__PM_BRL_TUR PRIMARY KEY (BRL_TUR_NO))

CREATE TABLE T_IVEDILIK_DERECESI(
        S_NO                      INTEGER        NOT NULL,
        IVEDILIK_DERECESI   CHAR(20) CHARACTER SET WIN1254 NOT NULL COLLATE WIN1254,
        CONSTRAINT PK_IVEDILIK_DERECESI PRIMARY KEY (S_NO))


CREATE TABLE T_HATLAR(
        S_NO                INTEGER        NOT NULL,
        BRL_NO_1       INTEGER        NOT NULL,
        BRL_NO_2       INTEGER        NOT NULL,
        BANT_GENISLIGI      INTEGER        DEFAULT 5600 NOT NULL,
        ACIKLAMA       CHAR(255) CHARACTER SET WIN1254  COLLATE WIN1254,
         CONSTRAINT T_PM_HATLAR PRIMARY KEY (S_NO))
```

```
CREATE TABLE T_BRL(
        S_NO                    INTEGER         NOT NULL,
        BRL_NO                  INTEGER         NOT NULL,
        BRL_KISA_ADI            CHAR(25) CHARACTER SET WIN1254 NOT NULL COLLATE WIN1254,
        BRL_UZUN_ADI CHAR(50) CHARACTER SET WIN1254  COLLATE WIN1254,
        BRL_TURU                CHAR(30) CHARACTER SET WIN1254 DEFAULT '1' NOT NULL COLLATE WIN1254,
        UST_BRL_NO              INTEGER         NOT NULL,
        BRL_SINIF        CHAR(30) CHARACTER SET WIN1254  COLLATE WIN1254,
        BRL_UST_KOOR            INTEGER,
        BRL_SOL_KOOR            INTEGER,
        BRL_ICON         BLOB SUB_TYPE 0 SEGMENT SIZE 80,
         CONSTRAINT T_PK_BRL PRIMARY KEY (S_NO, BRL_NO))
```

```
        CREATE TABLE T_MSJ_TUR(
                MSJ_NO                          INTEGER         NOT NULL,
                MSJ_ADI                         CHAR(    25) CHARACTER SET WIN1254 NOT NULL COLLATE WIN1254,
                MSJ_UZUN_ADI                    CHAR(   100) CHARACTER SET WIN1254  COLLATE WIN1254,
                MSJ_BASLIK_UZUNLUK              INTEGER   DEFAULT 1000 NOT NULL,
                MSJ_METIN_UZUNLUK               INTEGER           DEFAULT 2000 NOT NULL,
                MSJ_XML_BASLIK_UZUNLUK          INTEGER           DEFAULT 3000 NOT NULL,
                MSJ_XML_METIN_UZUNLUK           INTEGER           DEFAULT 3000,
                MSJ_ORT_UZUNLUK                 INTEGER           DEFAULT 2000,
                MSJ_MIN__UZUNLUK                INTEGER   DEFAULT 1000 NOT NULL,
                MSJ_MAX_UZUNLUK                 INTEGER           DEFAULT 10000,
                MSJ_ZIP_UNZIP_TIME              INTEGER   DEFAULT 500 NOT NULL,
                XML_MSJ_ZIP_UNZIP_TIME          INTEGER   DEFAULT 400 NOT NULL,
                NORMAL_MSJ_ZIP_UZUNLUK                  INTEGER           DEFAULT 400 NOT NULL,
                XML_MSJ_ZIP_UZUNLUK                     INTEGER           DEFAULT 1000 NOT NULL,
                CONSTRAINT T_PK_MSJ_TUR PRIMARY KEY (MSJ_NO))
```

```
CREATE TABLE T_BRL_MSJ_ZAMAN(
        S_NO                         INTEGER        NOT NULL,
        BRL_KULL_MSJ_S_NO            INTEGER        NOT NULL,
        MSJ_SAAT_DAKIKA              INTEGER,
        MSJ_SAAT                     SMALLINT DEFAULT 6,
        MSJ_DAKIKA                   SMALLINT DEFAULT 30,
        MSJ_SANIYE                   SMALLINT DEFAULT 1,
         CONSTRAINT PK_KULL_MSJ_SAAT PRIMARY KEY (S_NO))

CREATE TABLE T_SAAT_KISITLAMASI(
        S_NO                              INTEGER        NOT NULL,
        MIN_DOLULUK_ORANI                 INTEGER,
        MAX_DOLULUK_ORANI                 INTEGER,
        TOPLAM_DOLULUK_ORANI              INTEGER,
        MIN_ZAMAN_YAKLASIMI               INTEGER,
        MSJ_SAAT_ALT_SINIR                INTEGER,
        MSJ_SAAT_UST_SINIR                INTEGER,
        MSJ_DAKIKA_ALT_SINIR              INTEGER,
        MSJ_DAKIKA_UST_SINIR              INTEGER,
        CONSTRAINT PK_SAAT_KISITLAMASI PRIMARY KEY (S_NO))


CREATE TABLE T_BRL_KULL_MSJ (
         S_NO                        INTEGER        NOT NULL,
         GON_BRL_NO                  SMALLINT NOT NULL,
        ALAN_BRL_NO                  SMALLINT,
        GONDERILEN_MSJ_NO            SMALLINT NOT NULL,
        GONDERILME_SIKLIGI           INTEGER        DEFAULT 15,
        IVEDILIK_DERECESI            CHAR(10) CHARACTER SET WIN1254  COLLATE WIN1254,
        GON_BRL_KOMSU                CHAR(5) CHARACTER SET WIN1254  COLLATE WIN1254,
        GON_BRL_AST                  CHAR(5) CHARACTER SET WIN1254  COLLATE WIN1254,
```

```
    IVEDILIK_DERECESI_SAYI  SMALLINT,
    CONSTRAINT PK_BRL_KULL_MSJ PRIMARY KEY (S_NO))


CREATE TABLE T_HAT_DOLULUK (
    S_NO                INTEGER         NOT NULL,
    HAT_NO              INTEGER,
    SAAT_BASLAMA        SMALLINT,
    DAKIKA_BASLAMA      SMALLINT,
    SAAT_BITIS          SMALLINT,
    DAKIKA_BITIS        SMALLINT,
    YUZDE_KAPASITE_DOLULUK          SMALLINT,
    DOLULUK_ORANI       SMALLINT,
    CONSTRAINT PK_HAT_DOLULUK PRIMARY KEY (S_NO))


CREATE TABLE T_MSJ_BEKLEME (
    S_NO                INTEGER       NOT NULL,
    HAT_NO              SMALLINT,
    MSJ_NO              SMALLINT,
    SAAT_BASLAMA        SMALLINT,
    DAKIKA_BASLAMA      SMALLINT,
    SANIYE_BASLAMA      SMALLINT,
    SAAT_BITIS          SMALLINT,
    DAKIKA_BITIS        SMALLINT,
    MSJ_TIPI            SMALLINT DEFAULT 1,
    MSJ_IV_DER_SAYI     SMALLINT DEFAULT 0,
    DOSYA_BOYUTU        INTEGER         DEFAULT 1000,
    DOSYA_TURU          VARCHAR(20) CHARACTER SET WIN1254  COLLATE WIN1254,
    BEKLEME_NEDENI VARCHAR(50) CHARACTER SET WIN1254 DEFAULT 'HAT DOLULUK' COLLATE WIN1254,
```

```
	BANT_GENISLIGI			INTEGER		DEFAULT 4800,
	MSJ_ZIP_TIME			INTEGER,
	MSJ_UNZIP_TIME			INTEGER,
	CONSTRAINT PK_MSJ_BEKLEME PRIMARY KEY (S_NO))

CREATE TABLE T_MSJ_BEKLEME_INCELEME(
	S_NO			INTEGER		NOT NULL,
	MSJ_SAAT_S_NO		INTEGER		NOT NULL,
	INCELEME_NO  INTEGER		NOT NULL,
	HAT_NO			SMALLINT,
	MSJ_NO			SMALLINT,
	SAAT_BASLAMA		SMALLINT,
	DAKIKA_BASLAMA		SMALLINT,
	SANIYE_BASLAMA		SMALLINT,
	SAAT_BITIS		SMALLINT,
	DAKIKA_BITIS		SMALLINT,
	MSJ_TIPI		SMALLINT DEFAULT 1,
	MSJ_IV_DER_SAYI		SMALLINT DEFAULT 0,
	DOSYA_BOYUTU		INTEGER		DEFAULT 1000,
	DOSYA_TURU		VARCHAR(20) CHARACTER SET WIN1254  COLLATE WIN1254,
	BEKLEME_NEDENI		VARCHAR(50) CHARACTER SET WIN1254 DEFAULT 'HAT DOLULUK' COLLATE WIN1254,
	BANT_GENISLIGI		INTEGER  DEFAULT 4800,
	BEKLEME_SURESI		INTEGER,
	CONSTRAINT PK_MSJ_BEKLEME_QUEUE_INCELEME PRIMARY KEY (S_NO))
```

```
CREATE TABLE T_QUEUE_INCELEME(
	S_NO			INTEGER		NOT NULL,
	MSJ_SAAT_S_NO	INTEGER		NOT NULL,
	INCELEME_NO  INTEGER		NOT NULL,
	HAT_NO			SMALLINT,
	MSJ_NO			SMALLINT,
	SAAT_BASLAMA		SMALLINT,
	DAKIKA_BASLAMA	SMALLINT,
	SANIYE_BASLAMA	SMALLINT,
	SAAT_BITIS		SMALLINT,
	DAKIKA_BITIS		SMALLINT,
	MSJ_TIPI			SMALLINT DEFAULT 1,
	MSJ_IV_DER_SAYI	SMALLINT DEFAULT 0,
	DOSYA_BOYUTU		INTEGER		DEFAULT 1000,
	DOSYA_TURU		VARCHAR(20) CHARACTER SET WIN1254  COLLATE WIN1254,
	BEKLEME_NEDENI	VARCHAR(50) CHARACTER SET WIN1254 DEFAULT 'HAT DOLULUK' COLLATE WIN1254,
	BANT_GENISLIGI	INTEGER		DEFAULT 4800,
	BEKLEME_SURESI	INTEGER,
	CONSTRAINT PK_QUEUE_INCELEME PRIMARY KEY (S_NO))

CREATE TABLE T_QUEUE_INCELEME_ANALIZ(
	S_NO			INTEGER		NOT NULL,
	HAT_NO			SMALLINT,
	INCELEME_NO  INTEGER		NOT NULL,
	MINIMUM			INTEGER,
	MIN_ZIP			INTEGER,
	MIN_MSJ_SAYISI	INTEGER,
	MIN_XML			INTEGER,
	MIN_XML_ZIP		INTEGER,
	MIN_XML_MSJ_SAYISI	INTEGER,
```

```
NORMAL                    INTEGER,
NORMAL_ZIP                INTEGER,
NORMAL_MSJ_SAYISI         INTEGER,
XML                       INTEGER,
XML_ZIP                   INTEGER,
XML_MSJ_SAYISI            INTEGER,
ORTALAMA                  INTEGER,
ORT_ZIP                   INTEGER,
ORT_MSJ_SAYISI            INTEGER,
ORT_XML                   INTEGER,
ORT_XML_ZIP               INTEGER,
ORT_XML_MSJ_SAYISI        INTEGER,
MAKSIMUM                  INTEGER,
MAX_ZIP                   INTEGER,
MAX_MSJ_SAYISI            INTEGER,
MAX_XML              INTEGER,
MAX_XML_ZIP               INTEGER,
MAX_XML_MSJ_SAYISI        INTEGER,
BEKLEME_NEDENI                 VARCHAR(50) CHARACTER SET WIN1254  COLLATE WIN1254,
MIN_ZIP_MSJ_SAYISI        INTEGER,
NORMAL_ZIP_MSJ_SAYISI INTEGER,
XML_ZIP_MSJ_SAYISI             INTEGER,
ORT_ZIP_MSJ_SAYISI             INTEGER,
ORT_XML_ZIP_MSJ_SAYISI         INTEGER,
MAX_ZIP_MSJ_SAYISI             INTEGER,
MAX_XML_ZIP_MSJ_SAYISI         INTEGER,
MIN_XML_ZIP_MSJ_SAYISI         INTEGER,
CONSTRAINT PK_QUEUE_INCELEME_ANALIZ PRIMARY KEY (S_NO))
```

```
CREATE VIEW V_BMZ(
          G_BRLNO, A_BRLNO,
          GON_MSJNO, IV_DER,
          IV_DER_SAYI, MSAAT,
          MDAKIKA, MSANIYE,
          MSJ_SNO, ZAMAN_SNO
         ) AS

         select m.gon_brl_no, m.alan_brl_no, m.gonderilen_msj_no,
                m.ivedilik_derecesi, m.ivedilik_derecesi_sayi,z.msj_saat, z.msj_dakika, z.msj_saniye,
                m.s_no as msj_s_no, z.s_no as z_s_no from t_brl_kull_msj m, t_brl_msj_zaman z
         where (z.brl_kull_msj_s_no=m.s_no)


CREATE VIEW V_BHMZ (
         HAT_NO, GON_BRLNO,
         ALAN_BRLNO, MSJ_TIP_NO, IV_DER,
         SAAT, DAKIKA,
         SANIYE, IV_DER_SAYI,
         MSJ_SNO, ZAMAN_SNO,
         BANT_GENISLIGI) AS

         select h.s_no,
                z.g_brlno,  z.a_brlno,        z.gon_msjno, z.iv_der,
                z.msaat, z.mdakika, z.msaniye,        z.iv_der_sayi, z.msj_sno,
                z.zaman_sno, h.bant_genisligi from v_bmz z, t_hatlar h
         where ((h.brl_no_1=z.g_brlno) AND (h.brl_no_2=z.a_brlno)) or
                ((h.brl_no_2=z.g_brlno) AND (h.brl_no_1=z.a_brlno))
```

```
CREATE VIEW V_ANALIZ_QUEUE(
        INCELEME_NO, HAT_NO, GON_BRLNO,
        ALAN_BRLNO, MSJ_TIP_NO, IV_DER,
        IV_DER_SAYI, MSJ_SNO, ZAMAN_SNO,
        DOSYA_TURU, DOSYA_BOYUTU, BEKLEME_NEDENI,
        BEKLEME_SURESI) AS

       select i.inceleme_no,
             v.hat_no,  v.gon_brlno,
             v.alan_brlno, v.msj_tip_no,
             v.iv_der, v.iv_der_sayi,
             v.msj_sno, v.zaman_sno,
             i.dosya_turu, i.dosya_boyutu,
             i.bekleme_nedeni, i.bekleme_suresi
       from v_bhmz v, t_queue_inceleme i
       where (i.msj_saat_s_no=v.zaman_sno)
       group by
             i.inceleme_no, v.hat_no, i.dosya_turu,
             i.bekleme_nedeni, i.bekleme_suresi,
             v.gon_brlno, v.alan_brlno,
             v.msj_tip_no, v.iv_der,
             v.iv_der_sayi, v.msj_sno,
             v.zaman_sno, i.dosya_boyutu
```

Dosya adı: XML_1224682.doc
Dizin: F:\hsayin\huseyin_sayin_tez_enformatik_1\huseyin_sayin_tez_enformatik
Şablon: C:\Documents and Settings\67633023\Application Data\Microsoft\Templates\Normal.dot
Başlık: CHAPTER 1-2-3-4-5-6
Konu:
Yazan: Yzb.Hüseyin SAYIN
Anahtar Sözcük:
Açıklamalar:
Oluşturma Tarihi: 17.11.2003 11:19
Düzeltme Sayısı: 14
Son Kayıt: 22.11.2003 1:42
Son Kaydeden: Hüseyin SAYIN
Düzenleme Süresi: 103 Dakika
Son Yazdırma Tarihi: 22.11.2003 1:42
En Son Tüm Yazdırmada
Sayfa Sayısı: 162
Sözcük Sayısı: 26.383(yaklaşık)
Karakter Sayısı: 150.387(yaklaşık)