Laura Liitsalo

Anne Salomaa

THE USE OF METADATA IN NETWORK MANAGEMENT SYSTEM AS A SOCIAL ACTION

Master's Thesis in Information Systems Science
2.2.2000

University of Jyväskylä

Department of Computer Science and Information Systems

Jyväskylä

ABSTRACT

Liitsalo, Laura Elisa
Salomaa, Anne Maria
The Use of Metadata in Network Management System as a Social Action / Laura Liitsalo and Anne Salomaa.
Jyväskylä: University of Jyväskylä, 2000.
131 p.
Master's Thesis

The purpose of our thesis is to study the possibility of using different metadata solutions in third generation mobile systems' network management and especially in performance management functional area.

The frameworks used in our thesis are Habermas' critical social theory and Lyytinen's Ph.D. thesis' general framework for information systems development. Lyytinen's framework is based on Habermas' theory. The differences between different metadata applications are studied by evaluating two solutions: Extensible Markup Language's and Resource Description Framework's applicability to presenting and defining metadata related to performance management.

We show that the change process in Universal Mobile Telecommunications System is derived from the organization context. We also show that both the amount of the interest groups and the services are increased. That poses a need for social activity both in the network management systems development and in the system itself. This is why critical social theory can be used in the examination of Universal Mobile Telecommunications System. We show that the use of metadata in performance management is most of all communicative action. We find that communicative action type dominated also in different types of metadata. Performance management has been traditionally seen as a normative and technical system. We suggest that with an appropriate metadata solution performance management can become more communicative by nature. We also show that the fundamental problem in language context, the problem of meaning, can be partly fulfilled with a suitable schema, for example Extensible Markup Language and Resource Description Framework schemas.

Keywords: critical social theory, Universal Mobile Telecommunications System (UMTS), network management, performance management, metadata, schema, Extensible Markup Language (XML), Resource Description Framework (RDF)

TIIVISTELMÄ

Liitsalo, Laura Elisa
Salomaa, Anne Maria
Metatiedon käyttö verkonhallintajärjestelmässä sosiaalisena toimintana / Laura Liitsalo ja Anne Salomaa.
Jyväskylä: Jyväskylän yliopisto, 2000.
131 s.
Pro gradu -tutkielma

Tämän tutkielman tarkoituksena on selvittää eri metatietoratkaisujen eroja kolmannen sukupolven matkapuhelinverkkojen verkonhallinnassa sekä erityisesti suorituskyvyn hallinnan toiminnallisella osa-alueella.

Tutkielman viitekehyksenä käytetään Habermasin kriittistä sosiaalista teoriaa sekä Lyytisen väitöskirjan yleistä tietojärjestelmän kehittämisen viitekehystä, joka pohjautuu Habermasin teoriaan. Eri metatietosovelluksien välisiä eroja tutkitaan arvioimalla kahden ratkaisun, extensible markup language ja resource description framework soveltuvuutta suorituskykymittauksiin liittyvän metatiedon kuvaamiseen ja suorituskykymittaustiedon käsittelyyn.

Me osoitamme, että muutosprosessi kolmannen sukupolven verkonhallintajärjestelmissä on organisaatiolähtöinen. Osoitamme myös, että sekä sidosryhmien että palvelujen määrä on kasvanut. Tämä aiheuttaa tarpeen sosiaaliselle toiminnalle sekä verkonhallintajärjestelmän kehittämisessä että järjestelmässä itsessään. Näistä syistä kolmannen sukupolven matkapuhelin verkkoja voidaan tutkia kriittisen sosiaalisen teorian avulla. Osoitamme metatiedon käytön suorituskyvyn hallinnassa olevan pääasiassa kommunikatiivista toimintaa. Me havaitsemme, että kommunikatiivinen toiminta hallitsee myös eri metatietotyypeissä. Perinteisesti suorituskyvyn hallinta on nähty normatiivisena ja teknisenä järjestelmänä. Me ehdotamme, että tarkoituksenmukaisen metatietoratkaisun avulla suorituskyvyn hallinnasta voi tulla luonteeltaan kommunikatiivisempi. Me osoitamme myös, että kielikontekstin perimmäinen ongelma, merkityksen ongelma, voidaan osaksi ratkaista sopivaa skeemaa käyttämällä. Extensible markup language ja resource description framework skeemat voivat olla tällaisia sopivia skeemoja.

Avainsanat: kriittinen sosiaalinen teoria, universal mobile telecommunications system (UMTS), verkonhallinta, suorituskyvyn hallinta, metatieto, extensible markup language (XML), resource description framework (RDF)

TABLE OF CONTENTS

1 INTRODUCTION

The use of meta languages does not have a long history, even though the need for meta languages has existed a long time. As Pearson and Slamecka have said "A computer's universal ability to manipulate signs is a basis of uses of computers in information systems" (Lyytinen 1986, 80). According to Lyytinen (ibid.), this distinguishes computers from other technology. Thus, computers have always been able to use and manipulate signs but they have not been able to "understand" the meanings and contents behind the signs. This is why metadata is needed in several areas of computing. With new meta languages, like Extensible Markup Language (XML), it will be possible for example to interpret the contents of a web page or any other structured document. Some research has already been made in the Internet applications' area but very little on the use of new meta languages in other possible metadata usage areas. In our research, the use of metadata is studied in the field of telecommunications and to be more specific in the field of telecommunications network management. In the telecommunications network management's functional area, we concentrate on performance management.

The rapid development of new technologies and the need for appropriate information in the field of telecommunications puts a lot of pressure to the efficient information processing and presenting. New types of network elements challenge the network management system, which has to be able to process the information with an ever-growing speed and capability. The network management system has to also be able to accurately describe the network and the information in it. Other significant change in the third generation networks compared to second generation networks is the increased amount of services. The network and the network management system has many interest groups, including operators, end-users and service providers, who have to agree on the concept of service, what it is and what it includes. This agreement clearly requires social interaction between different interest groups. We see that social issues are emphasized in the context of third generation networks compared to the context of second generation networks. Thus, the network management system has to be able to take social issues into consideration in its functional areas so that it serves all the interest groups and their needs.

To these issues, we pose as a solution metadata and metadata applications. We claim that with an appropriate metadata solution, it is possible to model and represent the performance management data in a suitable way. Metadata solutions are also able to serve better all the interest groups and their various needs. This is because we believe that metadata solutions offer better visibility to the system and also offer more accurate and specific information that improves the quality of the service. In our thesis, we introduce XML and Resource Description Framework (RDF) schemas as two advantageous metadata solutions.

1.1 Research Objectives and Methods

The objectives of this research are *defining* a telecommunication network and network management, *pointing out* why network management is needed, *showing* how network management system can be analyzed in Kalle Lyytinen's framework and Jürgen Habermas' critical social theory and *showing* why metadata is needed in network management. By a telecommunication network, we mean a network that consists of both analogue and digital telecommunications and associated support equipment (ITU-T 1996, 2). Network management is a group of activities needed in identifying conditions that may affect network performance and service to the customer, and the application of network controls to minimize their impact (ITU-T 1998, 3). The most important objectives are *finding out* the differences and similarities in data presenting between XML and RDF schemas and the possibilities of using both of them in the field of network management and specifically performance management. Schema can be understood as the unit on which the internal model of world is built on (Arbib, Conklin & Hill 1987, 7).

The framework used in this research is Kalle Lyytinen's Ph.D. Thesis' framework. Framework is based on Jürgen Habermas' critical social theory (CST). Critical social theory is a conceptual framework that belongs to the class of scholarly works known as social action theory (Lyytinen 1986, 134). Our thesis emphasizes the technical and practical cognitive interests of Habermas' three cognitive interests. ITU-Ts (Telecommunication standardization sector of International Telecommunication Union)

Telecommunications Management Network (TMN) concept is used as a frame for network management. A telecommunication network is also defined in the TMN context.

The main research problem in our thesis is the following:

> How does Habermas' critical social theory and Kalle Lyytinen's Information Systems Development framework support the selection between two different kinds of schemas in UMTS network management?

We can divide the main problem into four sub-problems:

> What kinds of issues have had an impact on the increased need of social activity in third generation networks and network management systems?
>
> How does network management system and performance management allow the activity of different social action types?
>
> How do different types of metadata and social action types relate?
>
> How does XML and RDF schemas differ from each other in presenting data in the network management and performance management context?

In our research, we use Kalle Lyytinen's framework and Habermas' cognitive interests and social action types to outline the research area. First chapters in our thesis are mostly based on a literature review. The research itself is evaluative study based on social action theoretic analysis. The comparison between XML and RDF schemas in metadata presenting is done in the context of telecommunications network management and in the functional area of performance management. In the comparison, we use Habermas' four social action types and different types of metadata based on Gilliland-Swetland's (1998) division.

1.2 Thesis Outline

In chapter two we present Jürgen Habermas' critical social theory and Kalle Lyytinen's Ph.D. Thesis' framework, which is based on Habermas' theory.

ITUs TMN concept and its relationship to a telecommunications network are described in chapter three. Also the TMN relationship to Open Systems Interconnection (OSI) model is discussed. We also present Universal Mobile Telecommunications System (UMTS), which is known as third generation mobile system. The most important part of chapter three is the description of performance management and the use of metadata in UMTS performance management. In UMTS, services and therefore service management are emphasized. UMTS is the first telecommunications system that enables the use of metadata and the use of XML or RDF schemas.

Metadata, the usage areas of metadata and meta languages are discussed in chapter four. Definition of metadata will be a basis for examples of a meta language and a metadata application. We introduce XML as an example of a meta language and RDF as an example of metadata application.

Chapter five consists of the evaluation of presented concepts and the analysis of the research problems stated above.

2 CRITICAL SOCIAL THEORY AND INFORMATION SYSTEMS DEVELOPMENT

This chapter is structured as the table 2.1 shows. The gray background indicates that it is selected as an approach or is otherwise essential to this study. The numbers refer to the sections in this chapter. Lyytinen's general framework of information systems development is not shown in table 2.1 but it is presented in section 2.4.

At first, we will describe the history of Jürgen Habermas' critical social theory, which forms the basis for Lyytinen's framework. After that, we will introduce the five social action types Habermas identifies. Next, we will introduce the three cognitive interests. Two of them, namely technical and practical cognitive interests have special significance to our thesis and to the comparison of XML and RDF (schemas). Technical and practical cognitive interests will thus be described in more detail than the emancipatory cognitive interest. The emancipatory cognitive interest is only shortly introduced. After introducing the three cognitive interests, we will define information systems development methodology (ISDM) and describe the paradigm, normative and resource components in teleological and cultural information systems development methodologies. We will also describe the three object system contexts in each social action type. Again, the discursive action type, which belongs to the emancipatory cognitive interest, is left out of the introduction. The three contexts are in an important role also in Lyytinen's framework, which will be described last in this chapter.

Table 2.1: The structure of chapter two

Habermas' Social Action Types 2.1.1	**Habermas' Cognitive Interests 2.1.2**	**ISDM Components**	**Contexts 2.3**
Purposive-rational	Technical 2.2	Paradigm component 2.2.2 Normative component 2.2.3 Resource component 2.2.4	Technology Language Organization
Normatively regulated	Practical 2.2		
Communicative			
Discursive	Emancipatory		

2.1 Habermas' Critical Social Theory

In the following sections we shortly describe the history of Habermas' critical social theory, introduce the typology of social action and theory of cognitive interests. We also describe the three cognitive interests.

German social philosopher Jürgen Habermas' (1929-) critical social theory was developed in the post war period. It belongs to the class of scholarly works known as *social action theory*. To quote Habermas' definition, social action theory is a "conceptual framework that clarifies conditions, means, contents, constraints, and objectives of all socially organized human behavior" (Lyytinen 1986, 134). Social action theory provides two answers: an answer to the question of "how human behavior and the proliferation of ideas are affected by social influences" and an answer to the question of "what are the purposes and contents of scientific inquiry and how is it related to socially organized human behavior" (Lyytinen 1986, 134-135).

A group of scholars associated with the institute of social research at the University of Frankfurt originated a school of thought, which was named the critical social theory. Its primary objective is the improvement of the human condition. It was intended to be a radically different approach from traditional social theories that would take into account the human construction of social forms of life and the possibility of their recreation. The researcher's attitude towards his world and work typifies the fundamental difference between traditional social theory and critical social theory. Traditional social theory is based on the analysis and understanding of the existing conditions. Critical social theory is concerned with finding alternatives to existing social states that more adequately deal with human desires. (Ngwenyama 1991, 268-269)

There are five fundamental assumptions on which the critical social theory programme for social research and practice is grounded. These are: people have the power to change their world, knowledge of the social world is socially constructed, reason and critique are inseparable, theory and practice must be interconnected, and reason as well as critique must be reflexive in practice. (ibid.)

Lyytinen has taken critical social theory into account because ISDMs could be modified to become more sensitive to the human and social issues by recognizing the relationship between technical progress and social change (Lyytinen 1986, 134). This means that ISDs should be seen as social actions and information systems as social systems. The influence of human being in the development and use of information systems cannot be ignored. The development from OSI to TMN and from TMN to UMTS is a good example of social change in information system. This is because UMTS is not just a normative model as TMN is. In UMTS, the importance of services is emphasized. It is also a good example of an ISD where social viewpoints have been taken into consideration. Network management system has to also be seen as a social system because of the various interest groups related to it. Other reason why Lyytinen has considered CST is that he believes that CST can help in solving the problems concerning the effective use of information systems (Lyytinen 1986, 6). Lyytinen divides these problems into five categories. We believe that especially problems concerning the information overload, incorrect data and the complexity of information systems can be at least partly solved with an appropriate metadata solution. These are the main reasons why we have chosen CST as a basis for our thesis.

Lyytinen's Ph.D. thesis "Information Systems Development as a Social Action: Framework and Critical Implications" is based on Habermas' interest theory and the theory of communicative action (Lyytinen 1990, 10). He shows that a majority of approaches in the ISD field shares an interest in technical control. However, information system development affects three areas of human activity. Lyytinen (1986, 78) calls these areas *object system contexts*. The three areas or object system contexts that have to be considered in ISD are *the technology context*, *the language context* and *the organization context*. These object system contexts are further discussed later in this chapter. Most information system development approaches treat all areas similarly presuming that knowledge of these domains can be acquired and evaluated in the same way (Lyytinen 1990, 10). This is, according to Lyytinen (ibid.) one reason for the high failure-rate and endemic problems in ISDs.

Several other researchers like Ngwenyama, Klein and Hirschheim have also considered CST for developing new and alternative approaches in designing and delivering information systems. The scope of these researches varies a lot as well as does the distance from the abstract reconstruction of critical theory. In most cases, these approaches are untried. (ibid.)

In the following sections, we will define the two principal aspects of critical social theory: the typology of social action and the theory of cognitive interests.

2.1.1 Typology of Social Action

According to Lyytinen (1986, 138), Habermas' typology of action originates from Hegel and his Jena lectures and Marx's view of human practice. The distinction between action orientations that are called *purposive-rational* and *cultural* arises from these sources. Habermas divides purposive-rational action orientation into *instrumental* and *strategic action* types. On the level of cultural action orientation, Habermas divides it into *normatively regulated*, *communicative* and *discursive action* types. The word "action" has multiple meanings in this context. It covers both physical actions and verbal behavior. It also includes interlinked behavior sequences. (Lyytinen 1986, 138-139)

Habermas' action that is called *purposive-rational* i.e. *teleological* is directed towards achieving success in an objective and external world. Purposive-rational action poses a need to achieve a fit between a man's desires and states of affairs resulting from his interventions. From this it follows that a correct choice of means, effectiveness, and an application of valid technical knowledge and true beliefs predominate in purposive-rational action. (Lyytinen 1986, 139)

As stated before, purposive-rational action can be divided into instrumental and strategic actions. In *strategic* action, there are at least two goal-directed subjects who try to achieve their ends by way of an orientation. They also try to influence on the decisions of other actors. *Instrumental* action, then again, involves an intervention into an external world, which is usually physical. The success of instrumental action depends on valid empirical knowledge and goodness of technical directives. Both types of purposive-rational action presume one undivided, external world to which the actor adjusts himself. (Lyytinen 1986, 140)

Normatively regulated and *communicative* action types adapt to cultural and social world that is inter-subjectively shared. In these action types there are always at least two interacting subjects who recognize each other as free individuals. According to Lyytinen, these action types require distinctions between three worlds: an objective world, a social world and an individual's inner world. Normatively regulated action points to group behavior that demonstrates a re-occurring pattern. Action is regulated by norms that are mutually accepted. The central concept is in compliance of these norms. Communicative action uses language or other sign-types in order to reach an understanding. The central concept is interpretation, which refers to negotiations between people who want to achieve a consensus. (Lyytinen 1986, 141-142)

Also fifth action type can be identified. This action type is called *discursive* action. As Lyytinen puts it, instrumental, strategic, normatively regulated and communicative actions do not express how people can become conscious about the different nature of various action types. They also do not show in what way people can get understanding of why something is true in the world or efficient in action or why people have achieved a consensus. However, it is possible to find answers to the statements above when we understand how people use language to raise validity requirements about made expressions. These claims can then be accepted or denied in an argumentation. (Lyytinen 1986, 144)

Social action types can also be defined according to their basic orientation and domain of action. The instrumental action is success-oriented and a non-social action type. The strategic action is social and success-oriented. Communicative action is oriented towards agreement and is a social action type. Normatively regulated is oriented towards norms and is a social action type. (ibid.)

In table 2.2 we have gathered the basic features of four action types, instrumental, strategic, normatively regulated and communicative, to show the differences and similarities between them. In the table, we describe *purpose, co-ordinating elements, world-relation, number of actors* and *evaluative criteria* of each social action type. Discursive action type is left out because it will not be further discussed in our thesis. Table 2.2 is based on Lyytinen's Ph.D. thesis. This table and the four action types will be one basis for our evaluation of UMTS network management systems development in chapter 5.

Table 2.2: Basic features of four social action types (based on Lyytinen 1986, 140, 143)

Feature Covered	Instrumental Action	Strategic Action	Normatively Regulated Action	Communicative Action
Purpose	Technical control	Pursuit of individual interest	Compliance to norm	Mutual understanding
Co-ordinating elements	Rational decision procedures and law-like knowledge	Rational decision strategies and recognized behavior rules	Values and mutually binding expectations	Recognized linguistic rules based on reciprocity and consensus
World-relation	One external	One external	One world	System of worlds
Number of actors	One	Two or more	At least two	At least two
Evaluative criteria	Truth Efficacy	Maximization of interest	Conformism Justification	Agreement Validity claims

2.1.2 Theory of Cognitive Interests

Methodological problems of knowledge acquisition in the social sciences were the main reason why Habermas made an effort to construct a framework. This framework was supposed to be a conceptual framework upon which a critical social theory research could be based on. He outlines a program for the development of this methodology in "Knowledge and Human Interests" (1971). (Ngwenyama 1991, 270) Habermas claims that all theories of scientific knowledge are simultaneously theories of human behavior, i.e. theories of socially acting and knowing subjects. Therefore, a view of social action is needed in every theory of scientific knowledge. Knowledge is always tied to action although this relationship is not necessarily a direct one. The connection between action and knowledge can be shown by relating the three *cognitive interests - technical, practical* and *emancipatory,* to the social action types. (Lyytinen 1986, 150-151) The relationship is shown in columns one and two of table 2.3. Habermas believes that the three cognitive interests drive all human inquiry. Each type of cognitive interest is believed to represent a frame of reference through which researchers understand and make sense of the world as they seek to obtain knowledge about it. (Ngwenyama 1991, 270-271)

Table 2.3: Fundamental human knowledge interests (based on Ngwenyama 1991, 270 and Lyytinen 1986, 153)

Knowledge Interest	Social Action	Object of Interest	Orientation	Knowledge Products
Technical	Purposive-Rational	Natural world Social structures	Prediction control	Scientific knowledge Technology
Practical	Normatively-Regulated Communicative	Social relations Tradition	Mutual understanding	Social consciousness Humanity
Emancipatory	Discursive	Technology Social relations	Social criticism	Norms for justice Freedom

Technical cognitive interest's concern is the human need for prediction and control of the natural and social world. It focuses on defining means for achieving given ends. Its major product is technology, for instance, management procedures. Technical cognitive interest is validated by its effectiveness. (Ngwenyama 1991, 270) Sciences that follow the technical cognitive interest are natural sciences such as physics and chemistry, engineering, and economics and operations research (Lyytinen 1986, 155).

Practical cognitive interest is concerned with our need for self-understanding. The focus is on the understanding of social forms of life, traditions, social behavior and relations. It offers an improved social consciousness and humanity as its products. (Ngwenyama 1991, 270) Historical-hermeneutic sciences such as history, anthropology and law are concerned with this type of knowledge (Lyytinen 1986, 155).

Emancipatory cognitive interest is related to our need for freedom from physical and mental restrictions and social distortions. Emancipatory cognitive interest focuses on the establishment of norms for justice and enrichment of human freedom. (Ngwenyama 1991, 270) It is the most fundamental of the cognitive interests because it deals with the substantive and normative aspects of human life. Science that focuses on emancipatory cognitive interest is critical social science. (Lyytinen 1986, 156) In our thesis, we will not discuss the emancipatory cognitive interest further. For instance, Lyytinen and Hirschheim have studied the emancipatory cognitive interest and discursive action type (see Lyytinen & Hirschheim 1989). They have especially studied the paradox related to the emancipation. According to them, an information system is both a promise and a threat to it (Lyytinen & Hirschheim 1989, 116).

2.2 Technical and Practical Cognitive Interests

In our thesis, we have chosen to concentrate on two of Habermas' three cognitive interests: the technical cognitive interest and the practical cognitive interest. In the following, we rationalize the reasons why we have chosen these two.

Technical cognitive interest is relevant to our thesis because it is concerned in predicting and controlling the external world. Network management has traditionally been most of all that, prediction and control. According to Ngwenyama (1991, 270), technical cognitive interest focuses on defining means for achieving given ends. The development of a network management system for UMTS can be seen as related to technical cognitive interest in the sense that one of the development's goals is an effective system. As we can see in table 2.2, instrumental actions, which belong to the technical cognitive interest, can be evaluated by their effectiveness.

Practical cognitive interest is concerned with understanding the social forms of life as well as the social relations and behavior. The orientation is towards mutual understanding. Practical cognitive interest has aspects that are relevant when considering the new services UMTS offers. UMTS contributes to different interest groups who have to agree on the offered services. Practical cognitive interest also emphasizes the understanding of social forms. In our thesis, we try to understand and evaluate the social form and behavior of UMTS network management.

The relevance of these two cognitive interests and the social action types they refer to is further discussed in chapter 5. Next, we will define the development methodology concept based on Lyytinen's (1986) definition.

2.2.1 Information Systems Development Methodology

In our thesis, we will define information systems development methodology as Lyytinen (1986, 105) has defined it. According to Lyytinen, in ISDM can be found both practical and scientific meanings. Practical meaning occurs in the sense that methodologies try to educe the best practices. ISDMs have scientific meaning because ISDMs contain normative principles used in explaining and understanding object systems and their change. From this follows the definition Lyytinen presents in his Ph.D. thesis.

> "Development methodology is an organized collection of concepts, beliefs, values and normative principles supported by material resources that helps a development group to perceive, to generate, to asses, and to carry out in a systematic manner changes in object systems over a set of contexts". (ibid.)

According to Lyytinen, his definition refers to existing methodologies because all existing methodologies apply concepts, values and beliefs to perceive, generate, assess and carry out changes in object systems. Methodologies include also the languages by which change actions are represented, selected and implemented. Most methodologies are also limited in the sense that they do not take all three object system contexts into account. Most methodologies also assume that change occurs predominantly in the technology context. (ibid.)

Lyytinen presents five requirements for ISDMs that are found in the information system literature. First, ISDMs should be complete, meaning they should cover the whole development process. Secondly, ISDMs should be usable in many different development environments. Third, ISDMs should be suitable for supporting systematic analysis, suggesting, selection, understanding and implementation of change actions. Fourth, ISDMs should allow freedom during development. Finally, ISDMs should help continuous learning. From these requirements, we can derive following properties that ISDMs usually have. ISDMs are pre-specified and documented in writing, they provide a systematic way of proceeding in the process, and they are not complete but allow situational adaptation and interpretation. (Lyytinen 1986, 108-109)

ISDM has to be seen as more than a method but less than a theory. It can be seen as more than a method because a methodology does not have to be as precise as a method. A methodology can be modified and adapted to different development situations. It is neither as limited to specific tasks as method is. A methodology is also less powerful than a theory because it does not cover all cases and is not descriptive. (Lyytinen 1986, 108)

Within the ISDM three components can be found, which, according to Lyytinen (1986, 109), form the anatomy of an ISDM. The three components identified are the *paradigm*, *normative* and *resource components*. The relationships between the components form a hierarchical structure. The paradigm component *enables* the use of the normative component. The paradigm component is also *dependent* on the evolution of the resource component. The normative component also *consumes* the resource component and is *supported by* the paradigm component. (Lyytinen 1986, 110) We will describe the three components in ISDMs in the following sections.

2.2.2 Paradigm Component in Teleological and Cultural ISDMs

In this section, we describe the paradigm component in teleological and cultural ISDMs. As was mentioned in section 2.1.1, social action types in technical cognitive interest (i.e. instrumental and strategic social action types) belong to the teleological ISDMs. Social action types in practical cognitive interest (i.e. normatively regulated and communicative action types) belong to cultural ISDMs. Thus, we will describe the paradigm component in technical and practical cognitive interests.

To quote Lyytinen's definition (1986, 110), "the paradigm component in the methodology suggests something about the purpose, environments and content of change and the basic ways of carrying out the change". It provides, for example, conceptual schemes, metaphors and theories that the development group uses to organize all the information about the development, to suggest change actions and to make sense of them (ibid.).

This section is based on table 2.4, which represents six methodology paradigm components: *ontology, epistemology, rationality, metarules, theories* and *exemplars*. These methodology paradigm components are the basis of the description of paradigm component in teleological and cultural ISDMs. *Ontology* is concerned with the assumptions of the fundamental reality met during the systems work. Significant in ontology is whether it is based on empirical facts or is it socially constructed. *Epistemology* is concerned in the methods of obtaining knowledge needed for ISD. Object systems can be observed and measured objectively and subjectively. *Rationality* is concerned with the reasons why the developers should behave in a specific, maybe preferred way. The degree of free will is meaningful. *Metarules* help to select methodology. Usually there is a choice that has to be made between several methodologies. *Theories* help to explain and understand phenomena in contexts covered by a methodology. Theories do not necessarily have to be scientific. *Exemplars* are examples of successful problem solutions, which have been achieved by using the methodology. (Lyytinen 1986, 113-115)

Table 2.4: Technical and practical cognitive interest and the paradigm component of the ISDM (based on Lyytinen 1986, 177, 243)

Methodology Paradigm Component	Content in the Technical Interest	Content in the Practical Interest
Ontology	Realism, objectivating attitude, one external world	Intersubjective meanings and values, constructed social world different from nature
Epistemology	Empirical-analytic inquiring strategy	Hermeneutic inquiring strategies
Rationality	Truth, effectiveness	Alternative values and norms internal to a life-form
Metarules	Effectiveness of the ISD methodology	Not fixed beforehand
Theories	Prediction of states, selection of means	Better understanding, criticism
Exemplars	Technical advice	Interpretive clues

In technical cognitive interest, the ontological question is solved with realism and with objective attitude. The target is one external world. In practical cognitive interest, the ontological content is based on intersubjective meanings and values. Intersubjectivity in ISD means that goals in ISD are subjective; during the change process, the objectives and means must be found by negotiations and social interaction. This also means that the target is interpreted subjectively and is seen as a socially constructed world.

Epistemology in technical cognitive interest has empirical and analytic inquiring strategies. In practical cognitive interest, epistemology includes hermeneutic inquiring strategies. To quote Lyytinen (1986, 244), hermeneutic refers to "the tools and techniques by which the meaning of texts is interpreted or reinterpreted". Text in this context can be almost any expression of human life, for instance, a sequence of organizational action (ibid.).

Rationality in technical cognitive interest refers to the truthfulness of descriptions the actor has of the preconditions of the states he wishes to change. The correspondence between desired and achieved state is also evaluated (Lyytinen 1986, 178). In practical cognitive interest, values and norms are specific to every life-form. This means that every life-form comprises its own standards and principles of the ways the developer should behave and select means to proceed in the development process.

Metarules in technical cognitive interest are connected to the rationality and measuring of desired and achieved states. The actor is interested in the used resources: has the desired state been achieved with minimal resources? Lyytinen uses the term "economy of means" here (Lyytinen 1986, 178). In practical cognitive interest, metarules are not set beforehand.

In technical cognitive interest, the prediction of forthcoming states is emphasized. Like in rationality paradigm component, the selection of means is important. In practical cognitive interest, criticism, that is, critical theory is emphasized.

2.2.3 Normative Component in Teleological and Cultural ISDMs

The normative component of a methodology consists of a group of norms. There are three types of norms: *rules, prescriptions* and *directives.* Every methodology provides representation methods to identify the states, that is, directives and prescriptions, of object system. The directives indicate the order in which the changes are generated, and how they are selected and carried out. The prescriptions indicate how the development group is to be organized and how the argumentation of desirable changes is to be carried out. (Lyytinen 1986, 110, 117)

This section is based on table 2.5. In the table, we present the *root-definitions, working norms* and *organizing prescriptions* of instrumental, strategic, normatively regulated and communicative action types. *Choice directives* define methods to select between alternative information system designs, that is, change actions. Different alternatives can be evaluated by value-maxims and rules of calculus. *The sequencing and ordering directives* have a twofold purpose in a methodology: they determine the order in which change actions are selected in the same and different contexts and also specify necessary and/or possible ordering relationships between change actions. Change norms, that is, *procedural directives* and *grammar*, which is divided in *semantics* and *syntax*, help to identify, carry out and implement changes. *Performance prescriptions* indicate the purpose of stating standards of satisfactory performance for the development group, persuasions to perform better and sanctions to give, if performance falls below the standards. *Organizing prescriptions* consist of guidelines to arrange, shape and institutionalize interactions between developers. Without prescriptions, development activity would not be an institutionalized one, that is, organization. (Lyytinen 1986, 121-125)

In instrumental action, root-definition comes from technical, model and behavioral information systems. In strategic action, root-definition is based on technical, model, behavioral and strategic information systems. Normatively regulated action does not have any information system root definition. In communicative action root-definition information system is seen as text, as language game or as a set speech acts.

Choice directives in instrumental action express explicit rational choice and base on objective calculation of optimal design. Strategic action expresses explicit political choice and calculation of self-benefit based on group work. The choice directives of normatively regulated action adapt and conform to norms that realize accepted values. Choice directives in communicative action are open to interpretation and continuous learning.

Table 2.5: The normative component in teleological and cultural ISDMs (Lyytinen 1986, 181, 249)

Methodology Component	**Instrumental Action**	**Strategic Action**	**Normatively regulated Action**	**Communicative Action**
Root-definition	Technical IS, model IS, behavioral IS	Technical IS, model IS, behavioral IS, strategic IS	No IS definition	IS as text, IS as language game, IS as a set speech acts
Choice directives	Manifest rational choice, objective calculation of optimal design	Manifest political choice, group based calculation of self-benefit	Conformism to norms that realize accepted values	Open to interpretation, continual learning
Ordering directives	Specify ideal linear sequence of tasks	Subject to negotiation, vehicles in power play	n/a	n/a
Grammar: semantics	Restricted to objective facts, closed and fixed, consistent reasoning	Selected facts, open, conflicting, inconsistent reasoning	Norms, values and roles, conflict allowed consistent	Represent any world constituion of the world, open conflicting and ambiguous, inconsistent
syntax	Conciseness, efficient manipulation	Depends on culture, purposeful manipulation	Irrelevant	Open
Sequencing directives	n/a	n/a	Help to identify, analyze and change norms and values	Help to change and interpret language uses in a context
Procedural directives	Well defined, algorithmic	Ambiguous, subject to contextual interpretation	Well-defined	Both ambiguous and systematic
Performance prescriptions	Increase efficiency	Manifest power	Mediate accepted values	Irrelevant
Organizing prescriptions	Serve efficient intervention	Serve political influence	Maintain and define rolepatterns	Serve communication situations

Ordering directives in instrumental action specify ideal linear sequence of tasks. In strategic action, ordering directives are subject to negotiation as they can be seen as vehicles in a power play. Sequencing directives in normatively regulated action help to identify, analyze and change values and norms. In communicative action, they help to change and interpret language uses in a context.

The grammars of the four action types differ a lot. The semantics in instrumental action are restricted to objective facts. They are also closed and fixed and are placed to consistent reasoning. Syntax in instrumental action is concise and efficiently manipulated. Semantics in strategic action are selected facts. They are open, conflicting and are placed to inconsistent reasoning. Syntax in strategic action depends on culture and is purposefully manipulated. Norms, values and roles form semantics in normatively regulated action. They also conflict permitted consistent. Syntax in normatively regulated action is irrelevant. Semantics in communicative action represent any world constitution of the world. They are open, conflicting, ambiguous and also inconsistent. Syntax in communicative action is open.

Procedural directives are well defined and algorithmic in instrumental action. However, in strategic action they are ambiguous and are as a subject to contextual interpretation. Normatively regulated procedural directives can also been seen as well defined. In communicative action, they can be seen as both ambiguous and systematic.

Performance prescriptions increase efficiency in instrumental action. In strategic action, they demonstrate power. In normatively regulated action, performance prescriptions mediate accepted values. Performance prescriptions are irrelevant in communicative action.

Organizing prescriptions in instrumental action serve efficient intervention. In strategic action, they serve political influence. In normatively regulated action organizing prescriptions maintain and define role patterns, and in communicative action they serve communication situations.

2.2.4 Resource Component in Teleological and Cultural ISDMs

According to Lyytinen (1986, 127), development resources are needed to support, maintain and carry out change in information systems. This is why all methodologies must be able to identify resources and keep track of their consumption. Because resources need to be controlled, there is a need for organizing prescriptions.

Within the resource component can be found two kinds of resources: resources in the target object system and resources in the development organizations. Lyytinen concentrates on the resources in the development organizations. They can be divided into following five resources: *people, computer systems, money, tools* and *other resources*. Resources that relate to *people* mean the capabilities and skills needed to carry out ISD according to a methodology. *Computer system* resources refer to the technology needed to implement the change. *Money* resources refer to the investment needed to carry out the change actions. *Tools* resources are means that support and aid in performing tasks. Tools can be methodology independent and be further classified into manual and computer aided tools. *Other* resources mean, for instance, the physical locations and physical communication media and usually have a minor importance. (Lyytinen 1986, 127-128)

In the instrumental social action, the resources improve ISDs by making them more efficient. Increased efficiency can be measured, for example, by financial performance or software quality. In the strategic action type only few resources that are specific to it can be found. The only resources Lyytinen identifies (1986, 235) are the tools used only by some party to influence the system's development process. Also some operating systems that allow the monitoring of their users' actions can be seen as strategic resource components.

The resource component in the normatively regulated action has very little significance. According to Lyytinen (1986, 286), there are no specific tools that can support the normatively regulated action type. Similarly only a little attention is paid to the resource component in the communicative action.

2.3 Social Action Types and Object System Contexts

In this section, we describe the four social action types related to technical and practical cognitive interests and the three object system contexts identified in ISDs. The three object system contexts are the three areas of human activity an ISD effects on. Human activity in this context can be seen as development activity or process. Object system contexts identified in ISDs are the technology, the language and the organization contexts. Different social action types have different emphasis on different object system contexts. In this section, we examine and describe those emphases. First, we describe the three object system contexts in instrumental action type, which belong to the technical cognitive interest. The social action types and development activities in different object system contexts are gathered in table 2.6. The entries in the table are explained in more detail in the following sections.

Table 2.6: Classification of development activities by object system contexts and action type (Lyytinen 1986, 161)

Action Type	Technology Context	Language Context	Organization Context
Instrumental	Life-cycle phase model	Information elicitation process, data administration management	Implementation models, management science intervention, organizational rationalization
Strategic	Technology bargaining	Politics of language, bargaining about acceptable definitions and uses of words	Organizational politics, power play, change of transaction governance
Normatively regulated	Policies for technology adoption	Adherence to prescribed uses of language	Evolution of organizational roles and norms
Communicative	Interpretations of new technology	Change of language games, differentiation of symbolic cultures	Construction and sense-making of organizational reality

2.3.1 Instrumental Action Type

The development process or activity in instrumental ISDMs can be understood in *functional terms*. This is because information system and organization are external, in some sense independent objects that can be designed separately. Both the organization and the information system can also be divided into smaller parts and each component designed or changed separately. The solution between the information system and the organization is then obtained by recombining the components. The functional strategy is guided by procedural and consecutive directives, which means that the development process in instrumental information systems can be represented as life-cycle model based on a linear change that proceeds in systematic fashion. (Lyytinen 1986, 206)

Development activities in the three object system contexts in instrumental ISDs are: organizational change in the organization context, information elicitation task in the language context and systems engineering in the technology context. In the *organization context* the change is divided analytically into phases as suggested in some organizational change process models. In each phase, the designer uses the contingencies in the development process to direct the organization towards the desired organizational state. This is made possible by increasing the acceptance of new information systems by using implementation strategies. (Lyytinen 1986, 207)

In the *language context*, the information elicitation methods come into significance. The developer tries to find the most favorable information acquiring methods. The functional strategy is also important in data administration. (ibid.)

In the *technology context*, the functional strategy is characteristic of the life-cycle models. The functional application or system idea needs to be implemented into an operational system. This is where the number, type and sequence of transformations between abstraction levels determine development processes. Prototyping approach is an example of modified functional strategies. (Lyytinen 1986, 208)

2.3.2 Strategic Action Type

The strategic action type characterizes development processes by such terms as negotiation and contracting. The negotiation process is dependent on the underlying theory. Three different background theories can be found in strategic action. (Lyytinen 1986, 231) They are not further discussed in our thesis.

All three object system contexts place emphasis on the strategic action type. The *technology context* has nevertheless least relevance because information technology is usually regarded as politically neutral and non-strategic. Technology can be used as a competitive weapon or tool when strategic actions take place in an organization. (Lyytinen 1986, 216) Lyytinen states (1986, 230) that a technological change is a result of political struggle. It is also a "result of capitalistic rationalization that attempts to increase surplus value" (ibid.). Technologies that serve interest of all legitimate parties are adopted. This refers to the technology bargaining in table 2.6.

In the *language context*, the strategic use of language is supposed to help the developer to achieve the goals he has. The expressions used can be true or false, comprehensible or incomprehensible and legitimate or illegitimate. According to Lyytinen (1986, 217), information systems provide a means to persuade because they involve the use of signs. (ibid.) The control over data is also important in the language context. The organizational politics concentrate mainly on the competition over the control of data. (Lyytinen 1986, 230)

In the *organization context* in strategic action, it is assumed that organizations form their own type of reality. There are no deterministic laws they follow. Organizations are formed by its members' actions. From this follows, that the whole organization and its processes are characterized by the tensions between conflict and cooperation. (Lyytinen 1986, 223)

2.3.3 Normatively Regulated Action Type

In the normatively regulated action type, the development process changes the developers' norms, behavior-related expectations and role patterns. The change is controlled by norms. These norms guide the appropriate interactions between developers. Thus, the methodology is a collection of metanorms, rules that guide the identifying, analyzing and changing other norms. (Lyytinen 1986, 252)

Although the normatively regulated action type is not exposed to functional strategy, the development process can still be stated in a systematic manner. In the process, roles, needs, norms and values are identified, analyzed and changed (ibid.). According to Courbon and Bourgeois (ibid.), the change expresses features of the hermeneutic circle that leads to internationalization of new norms and role-expectations.

Normatively regulated action type emphasizes only the organization context. It sees an organization as a group of role-playing subjects. Important in an organization are the legitimate interactions. The members in the organization are identified by the roles they play and possess. The interactions between the roles are also further regulated by norms. (Lyytinen 1986, 248) Therefore, the evolution of organizational roles and norms become significant.

2.3.4 Communicative Action Type

Interpretive inquiry is what describes the development process in communicative action type. Both the language and organizational change are seen as learning processes, which comprehend a change in meanings. Like cultural inquiry, an ISD is a context dependent, circular sense-making process. In the process, the developer tries to make sense of the emerged issues and to build a sense of coherence to the development process. Studying ISD as a communicative action also reveals its inevitable ambiguity. (Lyytinen 1986, 282) This is according to Boland and Day (1982, 42-43), due to the symbolic nature of the development process.

Communicative action emphasizes the language and organization contexts. However, the mapping between the two contexts is not clear because in communicative action one object system context entails the other. On the one hand, language constructs the social reality, and on the other hand language is only significant as a part of the reality of organization context. The organization context emphasizes the construction and interpretation of the organizational reality. The language context emphasizes the differentiation of symbolic cultures. (Lyytinen 1986, 284)

The most significant outcome of the study of communicative action and ISD is that no system intervention can proceed without a discourse. The discourse has to happen in both object system contexts. The interpretation has to be seen as background process that affects all types of information systems development process activities (Lyytinen 1986, 283).

2.4 A General Framework of Information Systems Development

Lyytinen presents in his Ph.D. thesis a general framework for information systems development. In this section, we try to describe it and point out the essential parts of it concerning our thesis. Lyytinen's framework is based on Habermas' interest theory and theory of communicative action. Therefore, the beginning of this chapter can be seen as a basis for this framework. The first part of this chapter is also a basis for the examination of the framework.

In our thesis, we will use Davis and Olson's definition of information system and Lyytinen's definition of information systems development. Lyytinen presents these definitions in his Ph.D. thesis.

"Information system is an integrated, user-machine system for providing information to support operations, management, and decision-making function in the organization. The system utilizes computer hardware and software; manual procedures; models for analysis, planning and decision-making; and a database" (Lyytinen 1986, 182)

"Information systems development is a change process taken with respect to object systems (target) in a set of environments by a development group to achieve and/or to maintain some objectives" (Lyytinen 1986, 74).

The essential components of this definition can be shown as a figure (see figure 2.1), which indicates the relationships between the development group, object systems, objectives and environment. It can be also said that on the most general level, Lyytinen's framework consists of *development group*, which has *objectives.* Objectives are properties of *object system.* Object systems have *environments*. Development group carries out *change process*, which is guided by objectives. In the following sections, we will describe each element shown in the figure.

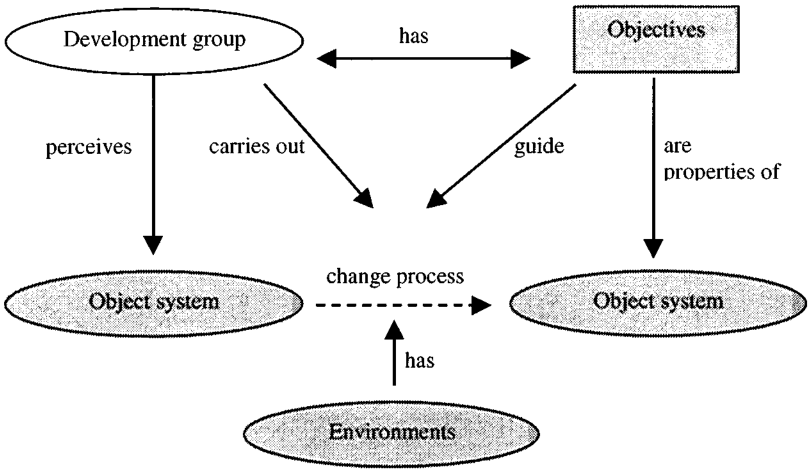


Figure 2.1: Information systems development (based on Lyytinen 1986, 75)

2.4.1 Object Systems and Object System Contexts

An object system can be seen, for instance, as a computer system, a network management system or an organization. The members of the development group perceive the elements and the relationships between them. Usually the object system perceived specifies the target of change. The perceptions of the developers are not, and do not even need to be, always congruent. In fact, they are often conflicting. As Lyytinen puts it, the convergence of perceptions is a condition for successful systems design. (Lyytinen 1986, 76)

Object systems need to be analyzed with the help of several concepts. These are the *object system contexts*, *super-context*, *underlying concepts*, *representation form*, *ontology* and *epistemology*. These concepts are important in helping to manage the complexity in perceiving the object systems. (ibid.) These concepts are clarified in the following.

The viewpoints, which generate object systems, are named *object system contexts* based on Welke and Konnsynski (Lyytinen 1986, 76). Object system contexts create a set of phenomena, which are feasible for that context and therefore belong to that context. (ibid.) As stated before, the object system contexts identified in ISD are the technology context, the language context and the organization context. Object system contexts are important also for the reason that they emphasize the significance of perception in information systems development. Thus, depending on the selected object system context (i.e. technology, language or organization), different parts of reality become visible. Underlying, there is questions of how an ISD environment should be perceived as "object systems" in order to obtain an acceptable change. (Lyytinen 1986, 77)

The object system contexts must be coupled to help predicting and analyzing a change in one context and how it affects the others. These couplings Lyytinen (1986, 77) calls *super-contexts.* In the case of super-contexts, the distinct object system contexts are called *sub-contexts.* Several, sometimes overlapping object systems can be within an object system context. These are identified as *overlapping concepts* or *theories.* They are used to clarify the levels in which the object system can be perceived.

2.4.2 Environment

Information systems development is not separate from other organization. According to Kling and Scacchi (Lyytinen 1986, 95), ISD is integrated in a web of surrounding conditions and factors, which affect what contexts are applied, how the process is organized and structured, how change actions are carried out, and what changes are desirable. Lyytinen calls these conditions and factors *environments* of information systems development. (Lyytinen 1986, 94-95)

Lyytinen divides environments into six types after Mathiassen (Lyytinen 1986, 95). In the *labor environment,* the keen interest is in the supply of labor to develop information systems, and in the labor's qualifications and experience. The *economy environment* is interested in economic possibilities of ISD. The *technology environment* is interested in information technology that is available, possibilities for launching new technology and constrains that existing technology sets. In the *application environment,* the keen interest lies on features of organizational area, in which ISD takes place such as its formal structure, social structure and base technology and also skills and attitudes of people. In the *external environment,* the key aspects are in organized bodies that operate in the surroundings of the application, in their objectives and strategies, and influence they can apply on ISD and its results. Laws and governmental regulations are taken into account during the ISD, including formal agreements and privacy in the *normative environment.* (ibid.)

2.4.3 Development Group

Developing an information system has to be seen as a collective process. Thus, information systems development is done within a formally organized group. Lyytinen calls the members of the development group *social agents.* The systematic relationships and interactions between the social agents are determined within the group. The social agents in a development group differ from a general social group in the sense that a development group must have structured relationships. According to Lyytinen (1986, 96), these relationships specify a set of differentiated roles, normative rules for each role, and lines of authority between roles and their communication channels. (Lyytinen 1986, 95-96)

2.4.4 Objectives

Information systems development is a process that proceeds intentionally. This is because ISD is regulated by *objectives.* This regulation can be implicit or explicit. Usually, information system development success is evaluated by achieved objectives and intended objectives. (Lyytinen 1986, 97)

According to Klein (ibid.), objectives are based on human value-orientations embedded in the act of development and can be reconstructed into design-ideals. Design-ideals identify the "good" design activities as well as the design activities that should be done during ISD. Several, fundamentally different approaches exist of how to choose design-ideals. Lyytinen identifies four aspects that need to be taken into consideration when dealing with the development objectives. According to these aspects, ideals and objectives can be partly imposed *explicitly* or *implicitly*, *clear* or *vague*, *uni-* or *multifunctional* and *conflictual* or *a-conflictual*. (Lyytinen 1986, 98)

2.5 Summary

In this chapter, we introduced Jürgen Habermas' critical social theory, its social action types, and its technical and practical cognitive interests. Technical cognitive interest was chosen for our thesis for the reason that it is concerned with predicting and controlling the external world. Network management's fundamental function is to predict and control the network's functionality and therefore technical cognitive interest suits evaluating network management well. We also examined the practical cognitive interest because it is concerned in things that are important when considering the increased amount of the services and interest groups in UMTS. We described all five social action types, which relate to the three cognitive interests. We also examined the four social action types in three object system contexts: the technology context, the language context and the organization context, which Lyytinen identifies in ISDMs. The four social action types and their basic features will be one basis for our comparison between XML and RDF schemas in chapter 5. Finally, we described Lyytinen's general framework of information systems development.

In the next chapter, we introduce ITUs Telecommunications Management Network (TMN) concept, which presents a framework for network management in general. Network management functions and benefits are also discussed. We will also introduce the Universal Mobile Telecommunications System, which is based on TMN framework and can be seen as a special case or example of TMN. It is however wider and especially applied properties and concepts are introduced in the next chapter. UMTS is known as the third generation mobile network and is one of the first mobile systems that enables the use of XML and RDF (schemas) especially in handling and presenting the performance measurement data. Thus, more detailed approach to networks, network management and especially performance management will be taken with the consideration of UMTS.

3 PRINCIPLES OF NETWORK MANAGEMENT

In this chapter, we will define basic concepts, functions and benefits of network management, describe ITUs Telecommunications Management Network concept and its relationship to a telecommunications network. Thus, network management will be examined in the TMN context. TMN provides a good framework for understanding the network management concept and overall architecture. We will also introduce Universal Mobile Telecommunications System UMTS and performance management in UMTS. UMTS is based on TMN. It can be seen as a specialization of TMN. TMN is not however entirely sufficient for UMTS for the reasons that UMTS system will have some impact on network management concept in general, for example on service management. Especially those applied concepts are discussed in this chapter.

3.1 Network Management Functions and Benefits

Network management can be understood as a group of activities that are necessary in identifying conditions that may affect network performance and service to the customer, and the application of network controls to minimize their impact. Network management includes, for example, monitoring the status and performance of the network on a real-time basis, which includes collecting and analyzing relevant data, detecting abnormal network conditions and initiating corrective action and/or control. (ITU-T 1998, 3)

Network operator, which can be considered also as service provider, benefits from efficient network management in many ways. One immediate benefit is the increased income due to an increase in successful use of services, like telephone calls. Improved service stimulates customers to use more services and that way the customer acceptance of new services increases. More efficient use of network results in an increased return on the capital invested in the network and an improvement in the ratio of effective to ineffective calls. Network management functions also lead into greater awareness of the status and performance of the network. This awareness helps the operator to prioritize the maintenance tasks. It also helps making decisions on further improvements. (ibid.)

3.2 Telecommunications Management Network, TMN

The purpose of the ITU-T TMN specification is to provide a framework for telecommunication management. Specification introduces the TMN concept, defines interfaces and reference points and blocks, which describe the functional, information and physical architecture. These architectures are discussed in detail in section 3.3. In the following sections, we describe the basic concepts in TMN and the relationship of a TMN to a telecommunications network.

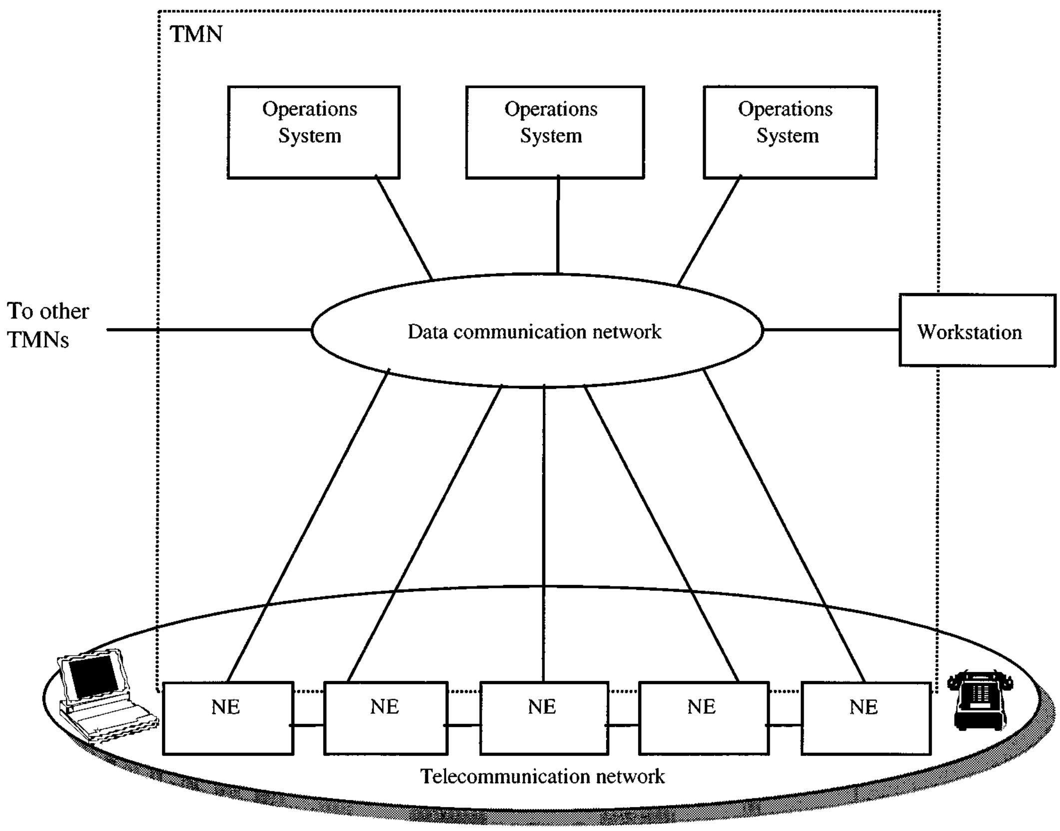
In the TMN concept, *management* refers to a set of capabilities that allow the exchange and processing of management information. This is meant to assist *administrations* in conducting their business efficiently. The term *administration* used in TMN concept includes public and private (customer and third party) administrations and/or other organizations that operate or use a TMN. TMN is logically kept distinct from the managed networks and services. This means that operators can perform management of many distributed equipment, networks and services from a number of network management systems. (ITU-T 1996, 1, 3) Separating TMN from the telecommunication network prevents problems that may occur in fault management; even in the case of failure in the telecommunication network, management will still be able to access the failing components. Thus, TMN has better fault management capabilities than OSI . The separate management network requires additional equipment, which raises the costs. Since failures may also occur in the management network, it will be necessary to manage the management network too. This meta management also introduces additional costs. (Pras, van Beijnum & Sprenkels 1999, 21)

3.2.1 Relationship of a TMN to a Telecommunications Network

In the TMN context, *a telecommunication network* is assumed to consist of both analogue and digital telecommunications and associated support equipment, such as, transmission systems, signaling terminals, front-end processors and file servers. When managed, such equipment is generically referred to as network elements (NEs). *A telecommunication service* consists of a range of capabilities provided to customers.

Examples of the networks, telecommunication services and types of equipment that can be managed by the TMN are TMN itself, public and private networks, digital and analogue transmission systems, tele services and software provided by or associated with telecommunications services, for example, short message databases. Conceptually a TMN is a separate network that interfaces a telecommunications network at several points to control its operations and to send or receive information to or from it. (ITU-T 1996, 2-3)

Figure 3.1 shows the relationship between a telecommunications network and a TMN. As the figure shows, a TMN can be connected to other TMNs and can provide a user interface via workstation. All managed objects are shown as NEs.



Note: The TMN boundary may extend to and manage customer/user services and equipment.

Figure 3.1: General relationship of a TMN to a telecommunication network (ITU-T 1996, 2)

3.3 TMN Architecture

Within general TMN architecture, three basic aspects can be considered separately when planning and designing a TMN. These three aspects are:

- TMN functional architecture,
- TMN information architecture, and
- TMN physical architecture.

All aspects are described in detail in the following sections. Also the five functional areas of network management will be discussed in section 3.3.2.

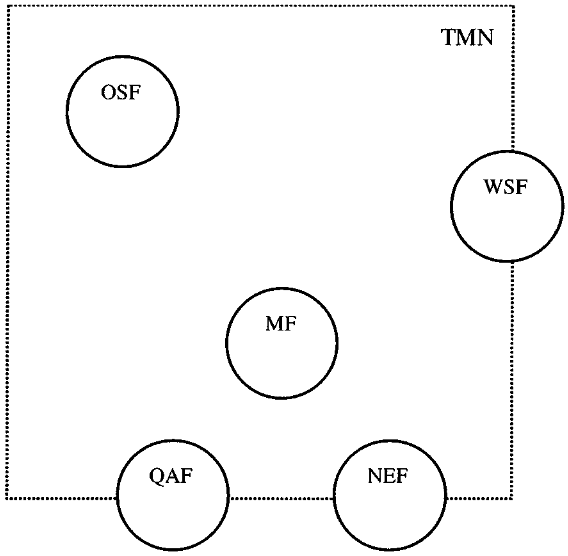
3.3.1 TMN Functional Architecture

A TMN provides the means to transport and process information related to the management of telecommunications network (ITU-T 1996, 6). The TMN functional architecture is defined in terms of five *function blocks* that allow the TMN to perform management functions. *Reference points* separate function blocks that exchange management information. Information is transferred between function blocks through a *data communications function* (DCF). Figure 3.2 shows the five function blocks. Two function blocks, Operations System Function (OSF) and Mediation Function (MF) are completely drawn within the box that represents TMN. This way of drawing indicates that these function blocks are completely specified by the TMN recommendations. The other three function blocks are drawn at the edge of the box to indicate that only parts of these function blocks are specified by TMN.

The most important of the five function blocks for our thesis is OSF block. OSF is responsible of processing telecommunications management related information for such purposes as monitoring, coordinating and/or controlling telecommunication functions including management functions.

Function blocks consist of *functional components.* They are identified but not standardized within the TMN area. The most important functional components to our thesis are the Operations System Function – *Management Application Function* (OSF-

MAF) and the *Information Conversion Function* (ICF). MAF function is, for example, adding value to raw information such as statistics and performance analysis. ICF is used in intermediate systems to provide translation mechanisms between the information models at both interfaces. (ITU-T 1996, 8)



OSF	Operations Systems functions	NEF	Network Element Function
MF	Mediation Function	QAF	Q Adaptor Function
WSF	Workstation Function	....	The TMN functional boundary

Figure 3.2: TMN function blocks (ITU-T 1996, 6)

Data Communication Function is used by the function blocks for exchanging information. Its main role is to provide information transport mechanisms. It may provide, for example, interworking, routing and relaying functions. When two different subnetworks are interconnected, the interworking functions will be part of the DCF. (ITU-T 1996, 11-12)

3.3.2 Functional Areas of Network Management

A TMN is intended to support a diversity of management areas, which cover the planning, administration, operations, installation, maintenance and provisioning of telecommunications networks and services. (ITU-T 1996, 4)

ITU-T recommendation X.700 (CCITT 1992, 3) categorizes telecommunications network management into five broad management functional areas. These areas provide a framework within which the appropriate applications can be placed to support the administration's business needs. Monitoring of the system and monitoring data is needed in all five functional areas. The five functional areas are *performance management, fault management, configuration management, accounting management* and *security management.* Management functions within these functional areas are provided by OSI management mechanisms. Several of those mechanisms are general in the sense that they are used to fulfill requirements in more than one functional area. In addition, managed objects are general in the sense that they may be common to more than one functional area (ibid.). These five functional areas are restricted in the sense that they disregard the aspect where outside response or comments should be able to affect to the behavior of the network.

Performance management is responsible for monitoring and controlling overall network performance, both within and across network services. It coordinates the actions of the lower level applications to recognize and resolve network performance problems. Performance monitoring is a part of performance management that includes reporting on network functioning. Performance management includes functions to, for instance, gather statistical information, maintain and examine logs of system state histories and alter system modes of operation for conducting performance management activities. (CCITT 1992, 4) Our thesis concentrates on the performance management area. It is further discussed in this chapter.

Fault management is responsible for managing network failures. Its purpose is to ensure smooth operation of the network and rapid correction of all problems that are detected. Fault management includes functions to, for example, maintain and examine error logs, trace and identify and correct faults. (ibid.)

Configuration management maintains up-to-date information about the operation and configuration status of the NEs in the network. It includes the management of the radio network, software and hardware management of the network elements and time synchronization operations. Configuration management includes, for instance, functions to change the configuration of the open system and initialize and close down managed objects. (ibid.)

Accounting management handles functions involved with charging. Accounting management includes, for instance, functions to inform users of costs incurred or resources consumed, and enable setting accounting limits and tariff schedules to be associated with the use of resources. (ibid.)

Security management handles minimizing of unauthorized use of the network. It covers the access to actual services and access to management operations. Security management includes functions, such as, the reporting of security-relevant events and the distribution of security-relevant information. (ibid.)

Based on ITU-T recommendation, TMN functionality can be described as following eight abilities:

- *the ability to exchange management information across the boundary between the telecommunications environment and the TMN environment;*
- the ability to exchange management information across the boundary between TMN environments;
- *the ability to convert management information from one format to another so that management information flowing within the TMN environment has a consistent nature;*
- the ability to transfer management information between locations within the TMN environment;
- *the ability to analyze and react appropriately to management information;*
- *the ability to manipulate management information into a form which is useful and/or meaningful to the management information user;*
- the ability to deliver management information to the management information user and to present it with the appropriate representation;
- the ability to ensure secure access to management information by authorized management information users (ITU-T 1996, 4).

We believe that the abilities in italics are most significant to our thesis because they especially refer to the management information and to possible metadata and schema usage areas. The abilities are discussed again in the next chapter.

3.3.3 TMN Information Architecture

TMN information architecture is the most significant aspect of the general TMN architecture when considering the possibility of using meta languages and schemas in network management.

Before introducing the TMN information architecture, we will define management information. According to ITU-T recommendation (ITU-T 1996, 18), the management information is considered from two perspectives: the *management information model* and the *management information exchange*. Management information model presents an abstraction of the resources being managed. It determines the scope of the information that can be exchanged in a standardized manner. The activity that supports the information model takes place at the application level and includes, for example, storing, retrieving and processing information. Management information model would be the basis of a schema. The other management information perspective, management information exchange, involves the DCFs and Message Communication Functions (MCFs), which are one type of TMN functional architecture functional components. The activity only involves communication mechanisms.

The information architecture is based on the same agent/manager concepts as OSI systems management. The information architecture uses an object-oriented information model. TMNs information architecture is in fact based on OSIs Management Information Model. The information that is exchanged in network management systems is modeled in terms of *managed objects*, which are conceptual views of the managed resources. A managed object is defined by the visible attributes and its boundary; the management operations, which can be applied to it; its response to management operations or other types of stimuli; and the notifications emitted by it. (ITU-T 1996, 19)

For understanding the agent/manager concept, it is necessary to understand that network management is a distributed information processing application because of the distributed nature of the managed environment. Information processing includes the exchange of management information between management processes. For one specific management association, the management processes take one of two possible roles, the agent role or the manager role. Figure 3.3 shows the interaction between manager, agent and objects.

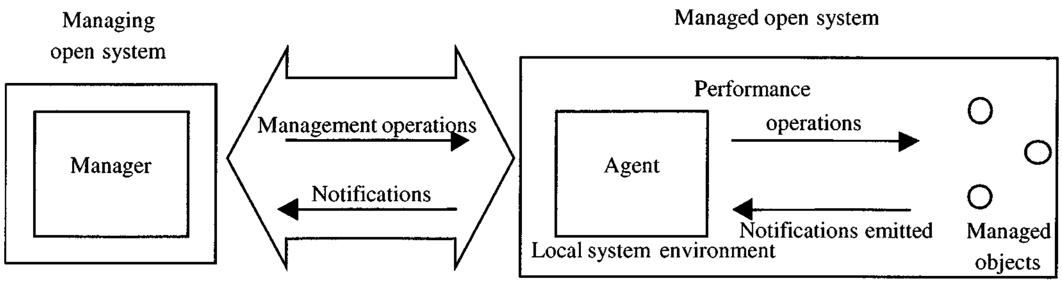


Figure 3.3: Interaction between manager, agent and objects (ITU-T 1996, 20)

Many-to-many relationship exists typically between managers and agents in the sense that one manager may be involved in an information exchange with several agents, and one agent may be involved in an information exchange with several managers. An agent may deny a manager's directive, for instance, for security reasons. A manager will therefore have to be prepared to handle negative responses form an agent. (ITU-T 1996, 20)

3.3.4 TMN Physical Architecture

Underlying the TMN are the physical entities through which the TMN is implemented. Thus, functional architecture defines the various TMN management functions and physical architecture defines how the various management functions can be implemented into physical equipment. Physical architecture is thus defined at a lower abstraction level than TMNs functional architecture (Pras et al. 1999, 10). Physical entities range from the *operations systems* (OS) to the *data communications network* (DCN), which carries the management information around the network. The TMN architecture focuses around key architectural components such as: operations systems, network elements, data communication network, and standardized interfaces. Management systems, which are referred to as operations systems (OSs), are responsible for controlling and managing network elements, for example, network switching nodes. These OSs and network elements exchange management information via a DCN through standardized interfaces, known as Q3 interfaces. Figure 3.4 shows an example of a simplified physical architecture for a TMN. A Q Adaptor is provided in case the entity does not offer the management interface of its own.

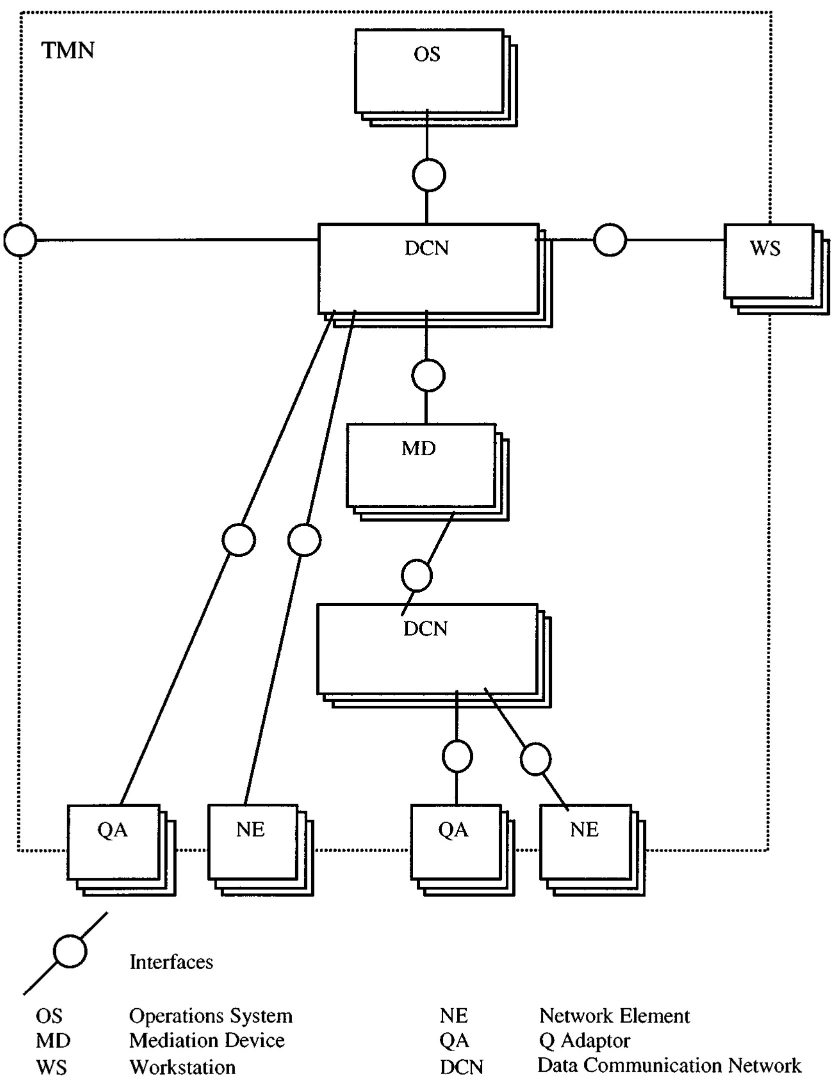


Figure 3.4: An example of a simplified physical architecture for a TMN (ITU-T 1996, 25) (See also figures 3.1 and 3.2)

3.4 Logical Layered Architecture

TMN recognizes that as in human societies, there is a hierarchy of management responsibilities. This hierarchy can be described in terms of *management layers*. The architecture that describes this layering is called the *Logical Layered Architecture*. It is described in this section. The concept of management layers has become the most important concept of TMN. The ideas behind this architecture were described first in 1989 as part of the Open Network Architecture (ONA). To decrease the complexity, the management functionality with its associated information can be decomposed into logical layers. A usual decomposition into layers is as follows (see also figure 3.8):

- element management layer,
- network management layer,
- service management layer, and
- business management layer. (Pras et al. 1999, 14)

In the *element management layer,* the functions of individual network elements are managed by OSFs. The management functions in this layer are vendor specific. Functions are hidden from the network management layer above. Examples of functions performed at the element management layer are measuring the resources that are being used (CPU-time, queue length), measuring of power consumption and detection of equipment errors. (ibid.)

Network management layer is responsible for managing the functions related to the interaction between multiple pieces of equipment. To the network management layer, the internal structure of the network elements is not visible. This is why the element management layer is needed; for instance, power consumption cannot be directly managed at this level. Examples of performed functions at the network management layer are creation of the complete network view, optimizing the network performance and detection of faults. (Pras et al. 1999, 15)

Service management is concerned with management of those aspects that users of the telecommunication network may directly observe. The concept "user" refers here to all interest groups: end-users, network operators, as well as service providers. Service management is built upon the management information provided by the network management layer. Still, the service management layer is not aware of the internal structure of the network. Examples of objects that can be directly managed at service management layer are routers, switchers and links. (Pras et al. 1999, 16) According to Pras et al. (ibid.), the notion of service management can be regarded as the most valuable contribution of TMN. Service management layer is emphasized in UMTS network management. In second generation networks services have not had the same significance as they have in UMTS networks. Services in UMTS are further discussed later in this chapter.

Business management layer is responsible for the management of the whole enterprise. Rather than seeing business management as goal achieving, it can be seen as *goal setting*. Thus, business management can better be related to strategical and tactical management instead of operational management, like the other management layers of TMN. (Pras et al. 1999, 18)

3.5 TMN and OSI Management

Between TMN and OSI management exists a strong relationship. This relationship is described in the following. TMNs functional architecture that is defined in terms of function blocks can, for example, be explained in terms of OSI concepts. Function blocks, containing functional components, can be compared to OSI protocol entities. Reference points that interconnect function blocks are in OSI terminology comparable to underlying service provides. (Pras et al. 1999, 19) Figure 3.5 illustrates the correlation.

In the beginning, there was only some collaboration between the management groups of CCITT (Consultative Committee on International Telephone and Telegraphy, nowadays known as ITU-T) and ISO/IEC (International Electrotechnical Commission of International Organization for Standardization). ISO/IEC standards had only some impact on the 1988 version of recommendation M.30. After M.30 was published, the collaboration between CCITT and ISO/IEC improved and many OSI management ideals were incorporated into TMN. The most important changes to TMN were the adaptation of the manager-agent concept and copying the ISO/IECs object-oriented approach. (Pras et al. 1999, 20)

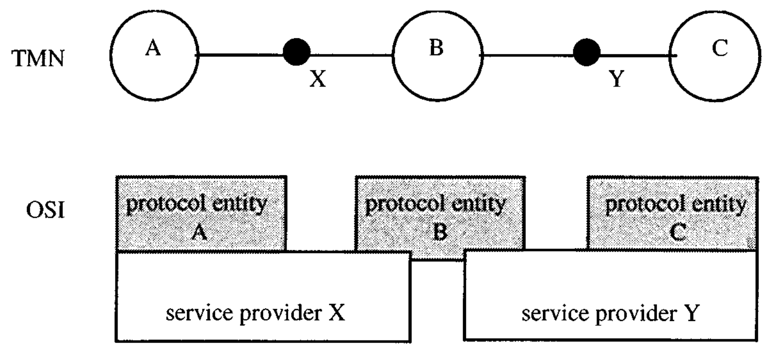


Figure 3.5: Relation between TMN concepts and OSI concepts (ibid.)

Regardless of the cooperation, at least three fundamental differences in philosophy still exist between ITU-T and OSI management groups. The first difference is that OSI has defined single management architecture while TMN has defined multiple architectures at different levels of abstraction. The second difference between TMN and OSI management is that TMN defines a structure for the multiple levels of management responsibility that exist in real networks. OSI management does not provide such structure. The advantage of such structure is the easier understanding and distinguishing of the various management responsibilities. The third difference is that as opposed to OSI, TMN suggests a conceptual separation between the managed network, that is, the telecommunication network, and the network that transfers the management information, that is, the data communication network, DCN. (Pras et al. 1999, 21)

3.6 Network Management in Universal Mobile Telecommunications System

The TMN framework is a good framework for all second generation network systems like GSM (Groupe Speciale Mobile). New, third generation systems like UMTS need however wider management concepts and changes to the management architecture.

Thus, the TMN model does not totally cover the UMTS management. For instance, the amount of provided services grows so rapidly that TMN does not have facilities to the UMTS service management. We see that the lack of complete service management in TMN shows that in TMN the social issues have not been taken into consideration properly. We believe that in UMTS the social issues can have emphasized importance. In the following sections, we introduce concepts that are applied and emphasized in UMTS. We also describe the overall UMTS standardization, development and architecture. Performance management and performance management measurements in UMTS are described last.

3.6.1 General Features of UMTS

Universal Mobile Telecommunications System is one of the major new third generation mobile communications systems being developed within the framework which has been defined by the ITU and known as IMT-2000. UMTS standardization started in European Telecommunications Standards Institute (ETSI) several years ago. The standardization goals were various: the new system should be able to provide existing services more efficient way, take the integration of fixed and mobile networks into account, harmonize and use synergy between the many kinds of wireless systems for different environments, and take advantage of the progress in technologies (Weiler & Klas 1997, 281). The overall objective of UMTS is to give roaming users the same functionality that they have at their desktop; data transport capabilities, security and account management facilities as well as the same speech quality. Thus, UMTS will deliver pictures, graphics, video communications and other wide-band information, as well as voice and data, direct to the people who can be on the move.

Like was said earlier, UMTS has many different interest groups. Also the UMTS standardization process is influenced by several factors. Both fixed and mobile network operators are such factors. Some of them have already invested in current systems while some are still waiting to make an investment to the new future system. Other factors are emerging technologies and their maturity, and standardization of all technical areas, which are expected to play a role in the UMTS system architecture. (ibid.)

UMTS will meet the standardization goals both revolutionary and evolutionary ways. Revolution is carried out in access network and evolution in the evolution of GSM core network. This way, operators and manufacturers who have already invested in GSM system do not have to create the whole UMTS system. Also, the reuse of several proven network concepts of GSM reduces risks - both operators' and investors' risks (Weiler & Klas 1997, 282). In UMTS, the evolution of current systems plays a significant role. In fact, according to ETSI (1997, 4), the vision of UMTS is based on the evolution of GSM. In the paradigm component of ISDMs two types of change can be recognized: evolutionary and revolutionary changes (Lyytinen 1986, 113). These match the notion of meeting standardization goals in UMTS both evolutionary and revolutionary ways.

UMTS success will be built on some essential concepts. According to Weiler and Klas (1997, 283), these key concepts are: self-adaptive and reprogrammamble mobile terminals; automatic methods for on-demand roaming negotiations, as a large number of subscribers and operators are expected; virtual home environment, which enables same service independent of location and operating environment; and new addressing principles. For a smooth transition for both end-users and operators from second to third generation systems, UMTS will use a phased approach.

3.7 UMTS Architectural Framework

In the following sections, we will describe the UMTS architectural framework. We will also give an example of an application protocol in section 3.7.1.

Architecturally, a UMTS system consists of the following components:

- 1 or more *Access Networks*, using different types of access techniques (e.g. GSM, UTRA and ISDN),
- 1 or more *Core Networks*, service specific or not (e.g. GSM, UMTS and ISDN),
- 1 or more *Intelligent Node Networks*, service logic and mobility management (e.g. IN and GSM), and
- 1 or more *transmission networks.* (3GPP 1999, 8)

Also, a mobile terminal (i.e. mobile phone), terminal equipment (i.e. personal computer) and terminal adapter can be seen as parts of UMTS. Terminal adapter is a converter, which enables terminal equipment to adapt to the mobile termination.

Every UMTS organization has its own Management Infrastructure. Each Management Infrastructure will contain different functionality depending on the role played and the equipment used by that UMTS Entity. However, the core management architecture of the UMTS organization is similar. Every UMTS organization provides services to its customers, needs an infrastructure to fulfill them (e.g. advertise and ordering), assures them (e.g. operation and quality of service), and bills them (e.g. rating and discounting). (3GPP 1999, 8)

Not every UMTS organization will implement the complete management architecture and related processes. Processes not implemented by a particular UMTS organization are accessed via interconnections to other UMTS organizations, which have implemented these processes. The UMTS Management Architecture is based on TMN (see section 3.3), and it will reuse functions, methods and interfaces that are suitable to its management needs. However, TMN does not cover the whole UMTS management since UMTS incorporates other technologies to which TMN is not fully applied, and UMTS faces new challenges that TMN does not address today. 3GPP (3G Partnership Project) expects that the eventual evolution of TMN will cover this ground. (3GPP 1999, 8-9)

Figure 3.6 describes the UMTS Management Reference Model, which shows the interfaces to other systems. Enterprise systems are information systems that are used in telecommunications organizations but are not directly related to the telecommunications aspects. Interface to other management systems can be one of two types: interface to other UMTS management system, that is, from one UMTS operator to other UMTS operator, or interface to other management systems, that is, to non-UMTS operators. (3GPP 1999, 9-11)

According to the 3GPP Technical Specification (3GPP 1999, 30-31), all management interfaces including previous and interfaces, within the management system of a single UMTS organization, can be studied from four different perspectives or levels. These are *logical, application protocol, networking protocol* and *physical level.* They are described in the following.

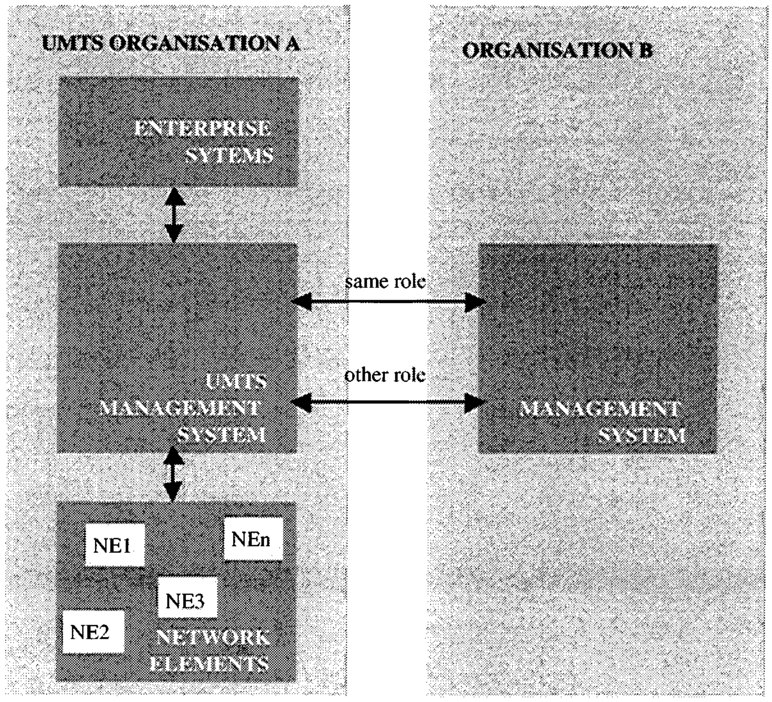


Figure 3.6: UMTS management system interactions (3GPP 1999, 10)

Logical level covers the mutual and conceptual knowledge of entities being connected by a given interface. All interactions at this level are totally standardized by 3GPP in terms of Information Models and Information Flows.

Application Protocol Level defines the primitives used to pass information across a given interface. It also defines the means to establish associations between the applications. Valid UMTS Management Application Protocol Suites are CORBA IIOP, SNMP and CMIP. Protocol suites are optional although in some cases one of those protocol suites will be recommended as a preferred one. The relevance of CORBA in the use of metadata is discussed on section 4.2.1. In addition, an example of one application protocol suite is presented in section 3.7.1.

Networking Protocol Level can use all standardized protocol suites that are capable of meeting the functional and operational requirements of the Logical and Application Protocol levels. 3GPP will not implement further standards in order to allow the influence of market.

Physical Level enables the use of Q-Adapter concept of TMN Architecture as physical implementation. It is however left to be the implementers' choice. The 3GPP specification does not prevent the use of Q-Adapters at other interfaces of the UMTS Management.

From the previous four levels follows, that for being accommodating to a given UMTS Management Interface, an UMTS entity must:

- implement the management functionality following the Information Model and flows specified by the relevant 3GPP UMTS Management Interface Specifications applicable to that interface,
- provide at least one of the valid Application Protocols, that is, CORBA IIOP, SNMP or CMIP,
- provide at least one standard networking protocol, and
- provide a Q-Adapter in case the entity does not offer the management interface of its own. Q-Adapter must be provided independently of any other UMTS NE and/or UMTS Management System. (3GPP 1999, 31)

3.7.1 Example of an Application Protocol: CORBA

Common Object Request Broker Architecture (CORBA) is one of the valid Management Application Protocol Suites in UMTS. It is an open standard developed by Object Management Group (OMG). In the following, we briefly introduce CORBA architecture and requesting. We chose CORBA as an example, because many of the upcoming systems will be built with object-oriented principles. At the moment, CORBA supports programming languages such as C++, Smalltalk and Java. Thus, CORBA permits more than one implementation language. New systems will also operate in a heterogeneous hardware and operating system environment. CORBA

Interface Definition Language (IDL), which is used in object's interface definitions, enables the CORBA requesting in diverse environment.

A CORBA Object Request Broker (ORB) is the middleware that enables the establishment of the client/server relationships between different objects. Objects can be either on the same machine, in the same process or across a network. A client object can transparently invoke a method on a server object, when using an ORB. The ORB intercepts the call. The ORB is also responsible for finding an object that can implement the request, pass it the parameters, invoke its method, and return the results. The client does not have to be aware of the object's location, its programming language and its operating system, only an object's interface is important, as all interfaces are defined with IDL. The client/server roles are used to co-ordinate the interactions between two objects. Thus, objects on the ORB can act as either client or server, depending on the event. (Orfali, Harkey & Edwards 1996, 68)

Figure 3.7 shows the client and server sides of a CORBA ORB. In the following, we briefly introduce CORBA architecture based on the figure below. At the moment, CORBA provides both *static* and *dynamic interfaces* to its services. *The client IDL stubs* provide the static interfaces to object services. These precompiled stubs define how clients invoke corresponding services on the servers. *The Dynamic Invocation Interface* (DII) enables the method to be invoked at run time. *The Interface Repository Application Programming Interfaces (APIs)* allow obtaining and modifying the descriptions of all the registered component interfaces, the methods they support, and the parameters they require. The Interface Repository is a run-time database that contains machine-readable versions of the IDL-defined interfaces. The APIs allow components to dynamically access, store and update metadata information. *The ORB Interface* consists of a couple of APIs to local services. (Orfali et al. 1996, 69-70)

The server IDL skeletons provide static interfaces to each service exported by the server. *The Dynamic Skeleton Interface (DSI)* provides a run-time binding mechanism for servers that need to handle incoming method calls for components that do not have IDL-based compiled skeletons. *The Object Adapter* sits on top of the ORBs core communication services and receives requests for service on behalf of the objects of the server. *The Implementation Repository* provides a run-time repository of information about the classes a server supports, the objects that are instantiated, and their identifiers (IDs). *The ORB Interface* consists of a couple of APIs to local services that are identical to those provided on the client side. (Orfali et al. 1996, 71)

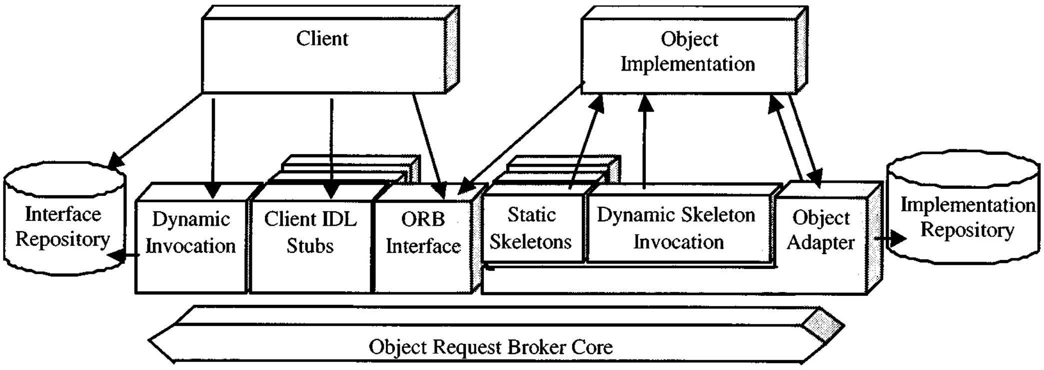


Figure 3.7: The structure of a CORBA ORB (Orfali et al. 1996, 69)

3.8 Management Concept in UMTS

UMTS management focuses on two layers of the TMN pyramid (see also section 3.4) shown in figure 3.8. These layers are network management and service management layers. They are described in following sections.

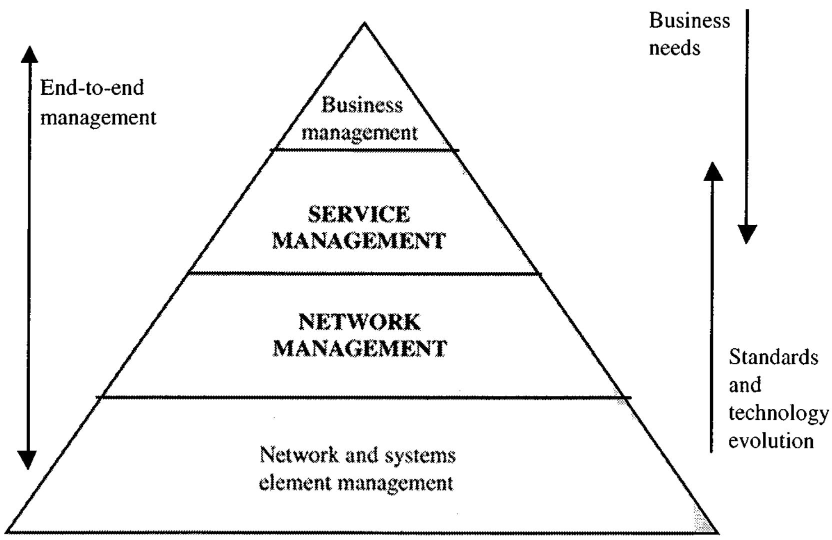


Figure 3.8: UMTS management pyramid (ETSI 1997, 9)

In the figure, element management is the set of functions that provide network element specific information to the network management layer and vice versa. In the UMTS Technical Report 21.06 (ETSI 1997, 8), and therefore in our thesis, element management is considered part of network management.

Network management has the difficult task of processing the requirements imposed by the service management. This is because network management has to provide the customer with an appropriate service onto the capabilities provided by the technology of its subnetworks. Due to this network management has to integrate, correlate and often summarize the information received from its subnetworks in order to pass on the relevant information to service management systems. (ibid.)

3.8.1 Network Management

Network management is in principal the same in UMTS and second generation networks. However, some matters need to be taken into consideration when designing a UMTS network management system. 3GPP has listed the requirements of network management for UMTS. We have chosen the following requirements because they are significant to performance management aspects. A network management for UMTS should:

- be capable of managing equipment supplied by different vendors including the management systems themselves.
- minimize the complexity of UMTS management.
- provide the communication between UMTS network elements and UMTS management systems or between UMTS management systems themselves via standardized interfaces (e.g. Q3, CORBA and SNMP) as appropriate and necessary.
- provide UMTS configuration capabilities that are flexible enough to allow rapid deployment of services.
- allow interoperability between network operators/service providers for the exchange of management/charging information. This includes interoperability with other networks and services (e.g. ISDN/B-ISDN, PSTN and UPT) as well as other UMTS networks.

- enable the support and control of a growing number of resources. This would allow the system to grow both in size and in complexity.
- address the management and assessment of system performance and operation through the use of common measurements, etc. This would enable a network operator/service provider to assess actual performance against planned targets.
- expose any information only once. (3GPP 1999, 7-8)

According to UMTS Technical Report 21.06 (ETSI 1997, 10), network management shall support the use of measurements from, for example, handover and user registration, for performance management, maintenance and planning purposes. The measurement data produced by the network may be used for traffic measurement evaluation of current network configuration and evaluation of possible fault situations around the UMTS network.

3.8.2 Service Management

The function of service management is to apply management principles to the services supported by UMTS (ETSI 1997, 13). The UMTS services do not directly map to identifiable network elements and therefore need a distinguished management. This changes the concept of management. The increased significance of services and service management emphasizes the meaning of performance management, which has to also be able to manage the performance of the services. This is why service management has special relevance also to our study.

One of the goals for UMTS is the ability to roam seamlessly between networks. Thus, users' applications need to negotiate to establish communications paths that have the required characteristics of bandwidth, delay and quality. The need to provide non-standardized services, created independently in a competitive, multi-operator environment, places new requirements on network elements. Network elements do not have fixed parameters as in second generation systems. The parameters can be selected, negotiated, mixed and matched by requirements of the required service. These need to be managed by service management. Service management takes data from network management as its input. (ibid.)

Some service management requirements are identical to those of network management. Service management requirements however emphasize the minimizing of costs, both operating and managing costs, the introduction and removal of new services, controlling access to UMTS services, and allowing management of services over shared interface. UMTS Technical Report 21.06 (ETSI 1997, 15) introduces service management life cycle, which is shown in figure 3.9. It is discussed in the following from the performance monitoring viewpoint.

In performance monitoring, statistical information will be gathered and maintained throughout the service life cycle. At the *service creation* level, this information could relate to the performance of the service creation process as a whole, for example, average response time. Information may also relate to response times or success rates in the creation of specific services. Performance data collected during *service provision* could be concerned with measuring response times and/or success rates on the provision of individual services. *Service invocation* performance statistics could be concerned with invocation success rates and breakdowns of causes of invocation failure. Performance data might be required during *service operation* by interoperating service providers for, for instance, charging purposes. At the *service deletion* stage, performance data could relate to average times take to respond to deletion requests and failure causes. (ibid.) This service management information is increasingly important due to the increased amount of managed services. We see that this poses a need to social activity because services and their management need to be agreed on. Thus, mutual understanding on these issues needs to be achieved.

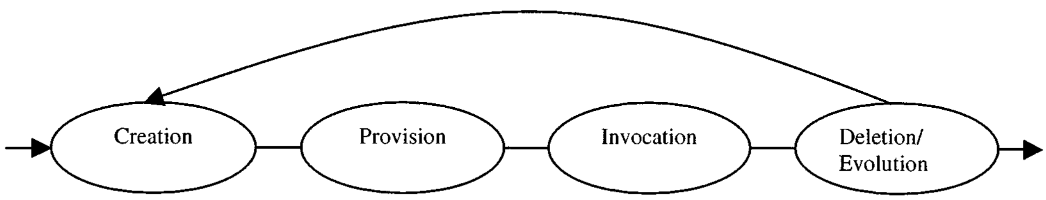


Figure 3.9: Service management life cycle (based on ETSI 1997, 15)

3.9 Performance Management

One of ITU-Ts five telecommunications network management functional areas is performance management. In our thesis, we have decided to concentrate on performance management because performance management functional area provides a possible area for the use of metadata. Like was said earlier, performance management is also responsible of managing the performance of the services (cf. fig. 3.9), which was the basis for the use of metadata application. Other possible functional area for the use of metadata is configuration management, but it does not provide as many or wide possibilities as performance management area does. Performance management and performance measurements and measurement concepts in UMTS are described in the following sections.

Performance management enables the behavior of resources in the network and the effectiveness of communication activities to be evaluated. According to ITU, performance management includes functions for statistical information gathering, maintaining and examining logs, determining system performance under natural and artificial conditions and altering system modes of operation for the purpose of conducting performance management activities (CCITT 1992, 4).

3.9.1 Performance Measurements in UMTS

In order to locate the faults and fault ratios in network and to control the network's capacity, an operator needs to perform performance measurements. Performance measurements are used mainly for maintenance and operation purposes. The collection of performance measurements is also done to support network planning and to examine the quality of service provided by the network. For effective performance monitoring, managed objects, that is, the network elements, need to be well known.

Network elements produce data in order to support several areas of performance evaluation. These areas are *traffic measurements*, *verification of the network* *configuration*, *quality of service* (QoS), *resource availability,* and *resource access.* (ETSI 1996, 22)

Traffic measurements are needed in controlling the network in real-time. In a usual case, the performance indicators are monitored, for example, in periods of 15 minutes. In case a shorter period is needed, the measurements can be received also online. Traffic measurements characterize the load of the network, for example, as blocked call attempts. Examples of what need to be measured include failed call attempt, handovers per base station controller per hour and pages per location area. *Network configuration evaluation* comprises the evaluation of the effectiveness of network's configuration. *Quality of service measurements* will be implemented according to the UMTS quality of service classes specification. Because the quality of service is a measurement area that is visible to the customer, it has to cover the whole UMTS service area. The customer is not aware of the network architecture and cannot therefore understand the possible differences in quality in different parts of network. The quality of service extends the concept of five functional areas (see 3.3.2) in a way that it takes into account the user's possibility to give feedback on the network's functioning. *Resource availability measurements* give an operator an indication of the "bottlenecks" in the network. Traffic measuring partly covers this area; resource access measurements can be seen as more NE basis. (ibid.)

3.9.2 Measurement Concepts in UMTS

When a measurement is carried out, it has the following components: *measurement object*, *measurement setup* and *counters*. *Measurement object* is the NE or a component within the NE whose behavior was measured. Measured object can be abstract, for example, a time slot in the radio interface. *Measurement setup* is the start time of the measurement and the duration of the measurement. *Counters* represent a piece of information about the behavior of the measured object. Usually the counters are integers. There could be one counter for the total number of attempted calls and other counters for successful and failed calls. Also, there could be one counter for each reason a call could fail. (Pohja 1999, 3)

Measurement administration is a function that sets the schedule for carrying out the measurements. At the measurement setup, that is, the start time, the counters are zeroed and for the duration time incremented, when something that matches with the definition of the counter happens. For the purposes of measurement administration, the counters are grouped into *measurement types*. Measurement type restricts the measuring in a sense that either all or none of the counters in a given measurement type are measured at a given time. (ibid.)

Observations are reports of the behavior of the NE. The main difference between the measurements and observations is that measurements have a predetermined schedule while observations are asynchronous. Measurements also have predetermined content and duration, whereas observations have several content alternatives and an atomic nature. Observations can be stored into a database like measurements. *Traces* are special cases of observations. Traces contain information about a specific user or user equipment. Tracing is done over a number of NEs, for example, in the area of one switching center. Tracing produces events, for instance, on every location update and handover made during the call. Observations and traces have to be turned on explicitly. This is because a continuous observing or tracing would load a TMN, both the transmission links and the database, enormously. (Pohja 1999, 4) Observations can be received either as events, for example, CORBA messages or in a file. The former is called online and the latter offline observing. Observation (tracing) reports may later be displayed (and thrown away), stored in a database or exported to some external system. (Pohja 1999, 13)

3.10 Summary

In this chapter, we introduced the functions and benefits of network management and described the TMN concept. We also briefly introduced the five functional areas of network management. As was said earlier, in our thesis we will concentrate on the performance management, which provides one possible area for the use of metadata and schemas. We also introduced the new third generation system, UMTS, which has some significant impacts, like emphasized services, on network management and network management systems. In this chapter, we briefly introduced the UMTS development and standardization process, overall architecture and the management concepts emphasized in UMTS compared to those of TMNs presented earlier in this chapter. We stated that services and thus service management had emphasized in UMTS. We saw that especially due to the several interest groups this posed a need to social activity.

The next chapter presents the concepts of metadata and meta languages. Those concepts and their definitions will be the basis for examples presented later in the chapter. As examples, we will introduce a meta language called Extensible Markup Language, XML and a metadata application Resource Description Framework, RDF.

4 METADATA AND METADATA APPLICATIONS

In this chapter, we will introduce different types of metadata and characteristics of metadata. Metadata has relevance to this study because we see metadata and metadata applications as a solution for information overload and complexity issues caused by ever-growing performance measurements in UMTS. Most of all, we see that metadata could solve the problem that arises from the increased amount of services. We see metadata and metadata applications as solutions that take social issues into account because they offer better visibility to the system and thus offer better possibilities to provide the services different parties are interested of.

We will shortly discuss the importance of metadata to organizations and the roles of metadata in organizations. We will also discuss the distributed nature of network management and the importance of metadata in this kind of distributed environment. We will also specify the usage area of metadata and files that support metadata in performance management. Meta models, meta languages and schema theory will also be briefly discussed.

We will also introduce as examples one meta language and one metadata application. Extensible Markup Language can be defined as a meta language because it defines how to describe a collection of data. It is a streamlined and simplified subset of Standard Generalized Markup Language (SGML). We will also introduce World Wide Web Consortium's (W3C) Data Format Architecture, the basic structures of XML recommendation and XML schema. As an example of metadata application, we introduce Resource Description Framework. Its development and RDF schema will also be discussed. In both XML and RDF sections, we will emphasize the importance of schemas. The schemas are expected to bring more extensibility to these metadata applications.

Finally, we will present an example of XML and RDF schema usage in UMTS network management. As one possible solution, we will introduce Nokia's Open Measurement

Standard (OMeS). Some examples of OMeS usage with both XML and RDF will also be presented.

4.1 Relevance of Metadata in Network Management

In the previous chapter, we briefly pointed out four possible metadata usage areas (see 3.3.2). In the following section, with *management information* we refer to performance management information produced by several different network elements in UMTS.

First one of the possible metadata usage areas is the ability to exchange management information across the boundary between the telecommunications environment and the TMN environment, for example, in our case from network elements to performance management applications in TMN. We claim that the exchange of management information needs metadata, which helps to identify the real sources of management information.

The second area identified is the ability to convert management information from one format to another so that management information flowing within the TMN environment has a consistent nature. We suggest that with a meta language like XML it could be possible to create a XML file already in the network element, so that the management information would not need to be converted within the TMN environment at all. In this, we also argue that the use of structured languages and structured documents gives the possibility to interpret the management information.

The third area is the ability to analyze and react appropriately to management information. We suggest that analyzing the management information, including service management information, eases with appropriate metadata. The fourth area is the ability to manipulate management information into a form that is useful and meaningful to the management information user. In this case, we also see structured documents as a solution. The use of appropriate schemas makes the management information more useful and meaningful by the way that schemas make it possible to interpret the management information. The schemas also introduce the extensibility so that it is easier to show the users the information they are especially interested in.

In the previous chapter, we also listed requirements of UMTS network management (see 3.8.1). In the following, we again refer to performance management with management. We claim that with the help of metadata usage it is possible for UMTS network management to minimize the complexity of UMTS management. For instance, the possibility of presenting measurement data in XML format during creation, transferring and preservation reduces the complexity. This is because the data does not have to be converted from one file format to other during its lifetime in the system.

The requirement of providing the communication between UMTS network elements and UMTS management systems or between UMTS management systems themselves via standardized interfaces could be fulfilled by structured documents and middleware that enables distribution. One such middleware is CORBA, which was introduced in section 3.7.1. We claim that the requirement of enabling the support and control of the growing number of resources could be fulfilled by the use of metadata, which would allow the system to grow in both size and complexity.

The requirement of addressing the management and assessment of system performance and operation through the use of common measurements could also be fulfilled by appropriate metadata usage. Metadata could help a network operator to evaluate actual performance against planned targets. By improved measurement files, the actual interpretation of the measurements would become easier.

4.2 Outline and Approach

In the following sections, we define metadata and metadata application and present different phases of information objects, which contain metadata. We also concentrate on different types of metadata and their usage areas. We try to clarify the importance of metadata in organizations and especially in organizations related to network management.

There are several ways to define metadata. Howe (1997) defines "metadata as data that characterizes other data in a reflexive way, i.e., data about data. In data processing, definitional data provides information about or documentation of other data managed within an application or environment". Metadata may include descriptive information about the context, quality and condition, or characteristics of the data. In our thesis, by metadata application we mean a computer application or software modules that is able to produce and/or present metadata.

In general, the purpose of metadata is to ease information search, information management and information location. Metadata can be seen as data with structure, typically this is understood as at least attribute/value pairs, structure, and defined meaning. According to Gilliland-Swetland (1998, 5), metadata relates to more than just the description of an object. Metadata can also indicate the context, management, processing, preservation, and/or use of the resources being described. It can come from a variety of sources and be supplied by a human, created automatically by a computer, or inferred through a relationship to another resource, for example, a hyperlink. Metadata continues to accrue during the life of an information object or system. By information object Gilliland-Swetland (1998, 3) means "a digital item or group of items, regardless of type or format that can be addressed or manipulated as a single object by a computer". Metadata can be created, modified, and sometimes even disposed of at many points during the life of a resource. (Gilliland-Swetland 1998, 3-6)

In figure 4.1, we present the different phases through which information objects typically move during their life in a digital environment. We have modified the original figure (see Gilliland-Swetland 1998, 5) to fit to the network management system environment, and more precisely to performance management, better.

In the network management system objects enter this digital and distributed information system by being created digitally by network elements. These objects, that is, in our case performance measurement data files, are files whose syntaxes could be defined by XML schema or RDF schema. In the organization phase, files are automatically organized as XML files are transferred through middleware. Stored and distributed files are subjects to search and retrieval by users, who are mostly operators, but can be also other users of the network. Retrieved files are utilized for different purposes. Files may be preserved or disposed.

In each of these five parts of the life cycle it is possible to add descriptive information, that is, metadata to information objects. In our thesis, we mainly concentrate on organization part of the information object life cycle. We restrict the use of metadata to the transfer of performance measurement files and to the insertion of the files into the database that supports metadata.

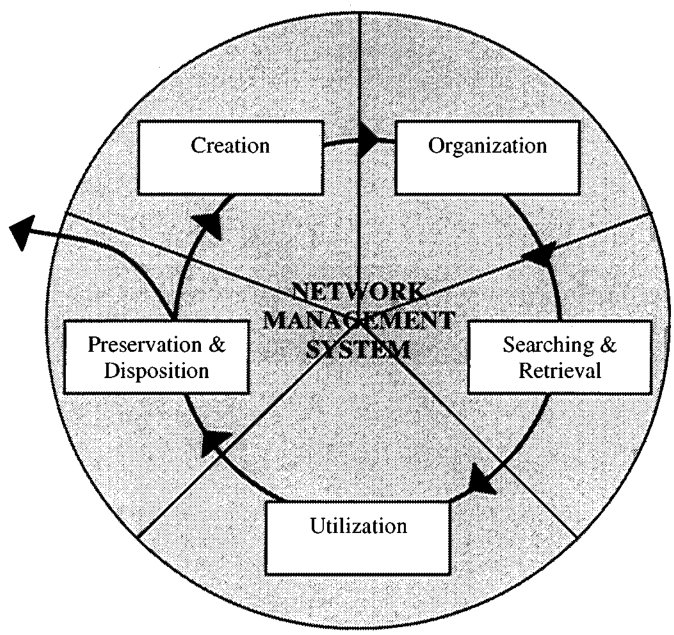


Figure 4.1: Different phases of information objects in a digital environment (based on Gilliland-Swetland 1998, 5)

As Gilliland-Swetland (1998, 2) states, metadata not only identifies and describes an information object, it also documents how that object behaves, its function and use, its relationship to other information objects, and how it should be managed. Different perspectives on metadata become important in the development of networked digital information systems. This in turn leads to a very broad conception of metadata.

To understand this conception better, Gilliland-Swetland (1998, 2) has broken it down into distinct categories - *administrative*, *descriptive*, *preservation*, *use*, and *technical* metadata - that reflect important aspects of metadata functionality. These types of metadata are presented in table 4.1. This table defines each of these metadata categories and gives examples of common functions that each of them might perform in a digital information system. In chapter 5, we will evaluate how Habermas' four social action types and different types of metadata according to Gilliland-Swetland relate. We will also use this table as one basis for our comparison between XML and RDF schemas.

Table 4.1: Different types of metadata and their functions (Gilliland-Swetland 1998, 3)

Type	Definition	Examples
Administrative	Metadata used in managing and administering information resources	– Acquisition information – Rights and reproduction tracking – Documentation of legal access requirements – Location information – Selection criteria for digitization – Version control
Descriptive	Metadata used to describe or identify information resources Manual metadata created by people	– Cataloging records – Finding aids – Specialized indexes – Hyperlinked relationships between resources – Annotations by users
Preservation	Metadata related to the preservation management of information resources	– Documentation of physical condition of resources – Documentation of actions taken to preserve physical and digital versions of resources, e.g., data refreshing and migration
Technical	Metadata related to how a system functions or metadata behaves	– Hardware and software documentation – Digitization information, e.g., formats, compression ratios, scaling routines – Tracking of system response times – Authentication and security data, e.g., encryption keys, passwords
Use	Metadata related to the level and type of user of information resources	– Exhibition records – Use and user tracking – Content re-use and multi-versioning information

In addition to there being different types of metadata and metadata functions, metadata also exhibits many different characteristics. Appendix A indicates some of the important attributes of metadata and examples of them. In our thesis and in network management, metadata can be characterized as structured and controlled metadata.

4.2.1 Metadata in Organizations

The importance of metadata to organizations has not been widely studied. According to Murphy (1998, 267), metadata can appear in very different roles in organizations, for instance, organizational memory, visibility, and network management. There is not much guidance available to organizations about when and how to develop and provide metadata for their needs.

However, according to Murphy (1998, 268) digital metadata has existed in organizations for many years, playing three major roles. First, metadata has been supporting organization's information search and retrieval, for example, querying a database or retrieving an archival record. Secondly, metadata has been used to define relationships between information elements and organizational entities, for example, in database design or systems integration. Finally, metadata has also served evidence, for example, for legal or audit purposes.

Today, an increasing amount of important information in organizations is created and maintained as both informal and formal digital documents (Murphy 1998, 267). This is the case also in network management and performance management where measurement data files can be seen as formal documents. For organizations to gain value from these digital documents, users must be able to discover them, to assess their value for a particular information need, and to access and use them. In addition, as the number of documents in digital form increases, metadata becomes more important for the effective use of the documents. In performance management, UMTS will add the number of documents radically as it introduces a huge number of new network elements that produce performance measurement data.

Murphy (1998, 269) also introduces roles for document metadata in organizations. These roles are introduced in the following table 4.2. In network and performance management, metadata has most of all a very traditional role in managing resources, that is, managing performance measurement data files.

Future network systems like UMTS will be based on distributed objects. New applications will be built of multiple autonomous components specialized for particular tasks. For example, CORBA is one proposed framework for managing interactions between distributed objects. CORBA architecture and CORBA requesting were already discussed in section 3.7.1. Metadata is an important feature of the CORBA architecture. All objects in CORBA compliant system are self-describing, which is why CORBA can be called an agile system (Orfali & Harkey 1997, 417).

Table 4.2: Roles for document metadata in organizations (Murphy 1998, 269)

Traditional Roles	New Roles
Discovery	Interpretation for re-use
Evidence	Individual memory aid
Use limitations and cost	Organizational memory
Retention and destruction	Persuasion or visibility
Managing resources	To describe accessibility

In a distributed environment, metadata can be described as the ingredient that allows creating agile client/server systems. An agile system is self-describing, dynamic and reconfigurable. The pervasive use of metadata, which means the consistent description of all available services, components and data, differentiates an agile system from a traditional client/server system. Metadata allows independently developed components to dynamically discover each other's existence and to collaborate over the network. From the previous follows, that the pervasive distribution of metadata is an important ingredient in a distributed component infrastructure. (ibid.)

4.3 Meta Languages, Meta Models and Schema Theory

In the following sections, we define meta languages, meta models and the concept of schema. We also give a short overview of some of the W3C data format specifications and the relationships between them.

According to the denotational theory, it is possible to describe meta languages in the following way. The object system and the sign-system, that is, language form together the Universe of Discourse for conceptual schema languages, called *meta languages*. Universe of Discourse is the external world information system models. It contains both the states of affairs in the part of the external world being of interest called an object system, and abstract concepts, by which signs in the information system can be related to their meaning in the object system. The conceptual schema language provides means to describe meaningful relationships. (Lyytinen 1986, 191-193)

Metamodeling has had an impact on the development of RDFs data model that will be discussed in section 4.6.2. *Meta models* can be based on semantic data models, which can differ greatly based on their purpose. They can also be further divided into different types depending on what type of a object is modeled. According to van Gigch (Tolvanen 1998, 82), metamodeling can be defined as "a modeling process that takes place one level of abstraction and logic higher than the standard modeling process". According to Tolvanen (1998, 85), most of the metadata modeling languages rely on some existing semantic data model. Semantic data models are easy to use, support communication, and yet are powerful and formal enough in describing objects.

The RDF data model can be described in a schema with, for example, XML or some other meta language. In the same way, XML schema can be constructed based on extensible datatypes. According to Arbib et al. (1987, 7), all humans have "internal model of the world". Our actions are directed to interacting with the environment and also to updating this internal model. It is also reasonable to assume that our internal model of the world is constructed from units. According to Arbib et al. (ibid.), *a schema* is the unit on which the internal model of world is built on. Each schema corresponds to an area of interaction, which can be an object in the usual sense, a detail of an object, or some area of social interaction. Like computer programs are usually built of components, schemas can be combined to form new schemas, too. Arbib et al. (ibid.) sees the schema theory as a theory that offers a style of programming. The theory allows schemas (or programs) to be *instantiated*, *deinstantiated* and *formed into groups* of concurrently active schemas that modify their activity based on the messages passed back and forth between them. Instantiating means that multiple copies may be activated with separate parameters. Deinstantiating means that individual instantiations may be turned off. Schema is both a process and a representation. It combines the declarative information of the resources and the program for action. (ibid.)

In the sections 4.4 and 4.6, we will introduce a meta language XML and a metadata application RDF as examples, which we have considered as advantageous ones. We will also present XML and RDF schemas and discuss their comparability to the schema theory and the concept of schema.

4.3.1 Relationships between W3Cs Data Format Specifications

Standard Generalized Markup Language can be seen as a basis for other W3Cs data format specifications. The SGML standard has given text-processing applications a common way of expressing the structure of data. Later, W3C has introduced several recommendations and specifications, which are more or less based on SGML. W3C's data format architecture is introduced in figure 4.2, where HTML is introduced as one particular application based on SGML. XML provides the same functionality as SGML in a simpler and more powerful way. XML is further discussed in section 4.4. W3C expects that future text markup will be built on XML rather than SGML (W3C 1997a, 2). RDF, a layer on top of XML, provides a common basis for expressing semantics. RDF is further discussed in section 4.6.

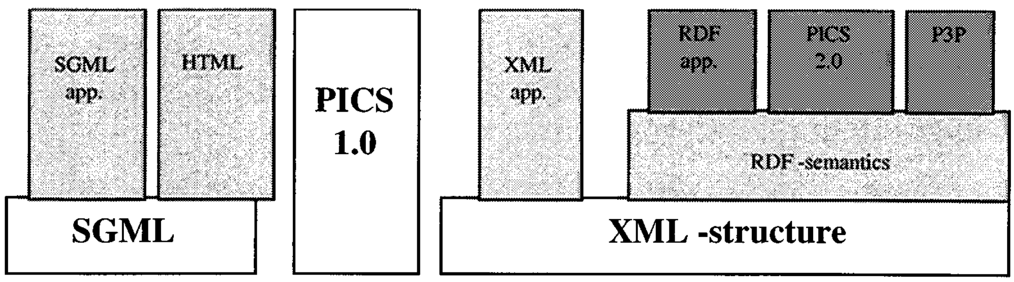


Figure 4.2: W3C's data format architecture (W3C 1997a, 1)

W3C plans that existing specifications such as PICS (Platform for Internet Content Selection) 1.0, written before XML and RDF should make the transition to a 2.0 version defined in terms of RDF and therefore XML. According to W3C, this will make it possible to mix PICS labels with information about privacy, for example, from the P3P project (Platform for Privacy Preferences Project), which will use RDF. W3C believes that many new applications, like P3P, will be directly built on RDF. (W3C 1997a, 3) W3C considers XML and RDF mostly as Internet applications. We however make an effort to show that they are feasible also in other application areas and especially in network management.

4.4 Example of Meta Language: Extensible Markup Language

XML was developed by an XML Working Group that was supported by the World Wide Web Consortium in 1996. XML is in fact based on three specifications: the XML language specification, the specification for XML links, and the proposal for XML stylesheets. Together these three parts make up XML. In our thesis, we describe only the language part of XML. Nowadays the XML language recommendation is based on a stable document.

4.4.1 SGML and XML

Actually, the development of XML started as early as 1969, when the IBM team developed a language that could implement their vision of markup that was not specific to any particular system. They named it the Generalized Markup Language. In 1974, Charles Goldfarb developed a validating parser that could read document type definition and check the accuracy of markup, without actually processing a document. However, the SGML standard took a long time to develop. It was not standardized until 1986. By the time it was standardized it had become large, detailed, powerful and also complex. (Goldfarb & Prescod 1998, 13-14)

As was said earlier, SGML is a language for the mark up of structured documents. The marking of document structure in SGML is not based on fixed fields of traditional separators. It is based on meta-tags that mark the meaning of each data element and which are added in ASCII text. Flexibility and the richness of features are strengths of SGML. However, the definition of SGML has been found hard to understand and learn. The strengths may also be seen as drawbacks for SGML, as its definition offers many alternative solutions for even simple tasks. (Malmi & Lantonen 1999, 22)

XML has the same flexibility of structures as SGML, since XML structures may also contain hierarchical elements. Structures may contain inclusions from other documents, allowing the distributed definition of data elements and cross-reference among data elements. XML documents are said to be highly understandable because of their clear structures and easily readable tags. (Malmi & Lantonen 1999, 23)

4.4.2 The Goals and Advantages of XML

According to the W3C (1998, 3-4), there are ten design goals for XML. They are introduced in the following.

- XML shall be straightforwardly usable over the Internet,
- XML shall support a wide variety of applications,
- XML shall be compatible with SGML,
- it shall be easy to write programs which process XML documents,
- the number of optional features in XML is to be kept to the absolute minimum, ideally zero,
- XML documents should be human-legible and reasonably clear,
- the XML design should be prepared quickly,
- the design of XML shall be formal and concise,
- XML documents shall be easy to create, and
- terseness in XML markup is of minimal importance.

There is a need for machine-understandable document content. The need is especially high in large and distributed systems like network management systems, where a lot of complex data is produced daily. There is a thread of data losing its relevance. XML is believed to be one possible solution for these problems because of its flexibility and clear structure.

XML provides a common syntax for a large number of new data formats. Many of these applications have more in common than simply being structured data. The data represents machine-understandable declarations about objects. (W3C 1997a, 2) XML also breaks up the information into smaller information components. The smaller and more specific the information component is, the more addressable and reusable it is. Components simplify complexity and increase flexibility for adapting to change. (Goldfarb & Prescod 1998, 147)

4.4.3 XML Recommendation

In this section, we describe some of the main concepts of the XML language specification. XML recommendation is based on a stable document (W3C 1998, 3). The recommendation was released on 10 February in 1998. The formal grammar of XML is given in the specification using a simple Extended Backus-Naur Form (EBNF) notation.

XML language specification describes a class of information objects called *XML documents* and also partially describes the behavior of computer programs that process them. XML documents are made up of storage units called *entities*. They contain either *parsed* or *unparsed data*. Parsed data is made up of characters, some of which form *character data*, and some of which form *markup*. A character in XML is an atomic unit of text as specified by ISO/IEC 10646 (see ISO 1993). Legal characters in XML are tab, carriage return, line feed, and the legal graphic characters of Unicode and ISO/IEC 10646. The formal structure of an XML document consists of XML markup, which specifies how the data in the document is stored and structured. (W3C 1998, 3-6)

A software module called an XML parser is used to read XML documents and provide access to their content and structure. XML parser does its work on behalf of another module called the application. (W3C 1998, 3) Parsing an XML document breaks it up into its component elements and enables interpreting of documents.

4.4.4 XML Documents

According to W3C (1998, 5), an information object is an XML document if it is *well-formed.* A well-formed XML document may in addition be *valid* if it meets certain further constraints. A textual object is a well-formed XML document if it contains one or more elements. There is exactly one element that is called the root of the document. No part of this element appears in the content of any other element. All the other elements that are delimited by start- and end-tags fit together properly within each other. A textual object has to meet all the well-formedness constraints given in XML language specification. Each of the parsed entities that is referenced directly or indirectly within the document is well-formed. (W3C 1998, 5-6)

In XML, text consists of mixed character data and markup. Markup takes the form of *start-tags, end-tags, empty-element tags, entity* and *character references, comments, CDATA section* delimiters, *document type declarations* and *processing instructions.* All text that is not markup is considered the character data of the document. The character data is made up of what is usually considered the real data in an XML document. (W3C 1998, 7)

Each XML document has both a *physical* and a *logical structure.* Physically, the document is composed of units called entities. Logically, the document is composed of declarations, elements, comments, character references and processing instructions, all of which are indicated in the document by explicit markup. The physical and logical structures must nest properly. This means that not any of the markup forms, which were introduced above, can begin in one entity and end in another entity. (W3C 1998, 5, 25)

Datatypes in XML are identified by URIs (Uniform Resource Identifier). The URI is simply a reference to a section of a document that defines the appropriate parser and storage format of the element. A URI is, for all practical purposes, a URL (Uniform Resource Locator). Today's most important form of URI is an extended form of the URL. URLs are uniform, in that they have the same basic syntax no matter what specific type of resource is being addressed or what mechanism is described to fetch it. (Goldfarb & Prescod 1998, 512)

XML namespaces provide a simple method for qualifying element and attribute names used in XML documents. They associate element and attribute names with namespaces identified by URI references. An *XML namespace* is a collection of names, identified by a URI reference, which are used in XML documents as element types and attribute names. XML namespaces differ from the conventional "namespaces" in computing disciplines in that the XML version has internal structure and is not a set. URI references that identify namespaces are considered identical when they are exactly the same character-for-character. (W3C 1999a, 1-2)

4.5 XML Schema

The XML language specification (W3C 1998, 5-6) defines the concepts of well-formedness and validity, which is very simple to check. Validation requires more work but allows the user to define more powerful constraints on the structure of the documents. To quote W3C (1999f, 2), "XML validity requires that a document follow the constraints expressed in its document type definition, which provides the rough equivalent of a context-free grammar for a document type". Sometimes, applications may need more informative definitions of markup constructs, or tighter, looser or different constraints on document structure that can be expressed using document type definition as defined in XML language specification. According to W3C (1999f, 2), there is also a widespread desire to allow markup constructs and constraints to be specified in an XML-based syntax to allow tools for XML documents to be used on the specifications. That is why XML schema development has begun.

The purpose of a schema is to define and describe a class of XML documents by using constructs to constrain and document the meaning, usage and relationships of their component parts: datatypes, elements and their content, attributes and their values, entities as well as their contents and notations. Schema constructs may also provide for the specification of implicit information, for example, default values. Schemas document their own meaning, usage, and function. The XML schema language is also more expressive than XML DTDs (Document Type Definitions), it is always described in XML and it is self-describing. (W3C 1999f, 2-4)

XML schemas offer several significant advantages over DTDs. For example, content models in XML schema are "open" by default, enable additional tags to be present within an element without having to declare every element in the XML schema. This provides an extensibility mechanism not present when using a DTD. XML schema also allows the specification of a datatype for an element or attribute. (XML Schema Developer's Guide 1999)

4.5.1 Usage Scenarios

XML Schema Working Group has defined seven different usage scenarios for XML use. According to W3C (W3C 1999f, 3), these scenarios are used during the development of XML schemas as design cases. In our thesis, performance management falls into the category of supervisory control and data acquisition.

- Publishing and syndication
- Electronic commerce transaction processing
- *Supervisory control and data acquisition*
- Traditional document authoring/editing governed by schema constraints
- Use schema to help query formulation and optimization
- Open and uniform transfer of data between applications, including databases
- Metadata interchange

XML schemas are used for different purposes in different design cases. Distribution of information through *publishing and syndication* services involves collections of XML documents with complex relations among them. In this case, structural schemas describe, for example, the properties of headlines, news stories, thumbnail images, and cross-references. In *electronic commerce*, libraries of XML schemas define business transactions within markets and between parties. The *management and use of network devices* involves the exchange of data and control messages. (W3C 1999f, 3-4)

In *traditional document authoring*, one important class of applications uses a schema definition to guide an author in the development of documents, for instance, a simple memo. The application can ensure that the author always knows whether to enter a date or a part number, and might even ensure that the data entered is valid. A query interface inspects XML schemas to guide a user in the *formulation of queries.* Any given database can emit a schema of itself to inform other systems what are the legitimate and useful queries. (ibid.)

According to W3C, XML has become a widely used format for encoding data, including metadata and control data, for *exchange between loosely coupled applications.* Such exchange needs expressive XML schema definitions to perform such interactions in a simplified way. There is also a growing interest in the *interchange of metadata*, especially for databases. (ibid.)

4.5.2 W3C XML Schema

The W3C XML schema work continues. The W3Cs schema specification is nowadays presented in two working drafts, named XML Schema Part 1: Structures and XML Schema Part 2: Datatypes (W3C 1999d and 1999e). In general, a schema specifies the structure of an XML document and constraints on its content. XML schemas are themselves also specified as XML documents. As said earlier, XML schema specification not only reconstructs the DTD constraints of XML language specification, it also adds the ability to define new kinds of constraints. (W3C 1999d, 6-7)

The XML schema specification supports the expression of specific constraints by including in the mechanism for the declaration of element types the option of specifying that its contents must consist of a valid string expression of a particular datatype. A number of other mechanisms are added that improve the expressive power, usability and maintainability of schemas as a means of defining the structure of XML documents. (W3C 1999d, 7-8)

According to W3C (1999e, 4), the limited datatyping facilities in XML have prevented validating XML parsers from supplying the exact type checking, which is required in many situations. XML schema specification addresses the needs of document authors and applications writers for a robust, extensible datatype system for XML, which could also be incorporated into XML parsers. (W3C 1999e, 4)

In appendix B is represented a normative schema for W3Cs XML schemas. The schema describes the structures part of the W3Cs XML schema specification. It is not discussed further here. It is appended for providing more specific information about W3Cs schema structures and for showing the structure of XML documents.

In section 4.3, we presented Arbib et al.'s schema definition. In W3C's XML schema we can find many similarities to this schema definition. XML schema is constructed of units called component parts. It is a representation because it has constructs to document and constrain the meaning, usage and relationships of component parts of the document. It can be seen as a process because it defines processing instructions for the parsers of the documents. XML schemas can be combined to form new schemas, for example W3C's XML schema is especially targeted to be used as a basis for other XML schemas. The XML schema is also extensible; it is possible to add new elements to a schema or to remove old elements from the defined schema. The units, representation, process and new schema combinations described above were also specified in Arbib et al.'s schema definition.

In the next section we will concentrate on one meta application of XML; RDF. As we could see in figure 4.2, RDF is built on XML structures.

4.6 Example of Metadata Application: Resource Description Framework

The following sections of this chapter describe the basic structure of the Resource Description Framework, which is a foundation for processing metadata. RDF schemas are described in section 4.7. Before reading more about RDF, it is essential to understand that RDF specification is still only a W3C recommendation, not a standard of any kind. The RDF references used for this thesis are W3C *recommendations* (Model and Syntax Specification) and *proposed recommendations* (Schema Specification). Any changes to RDF structure are still possible and therefore only references that are mentioned in the references chapter are valid concerning this thesis. Our thesis does not describe the whole RDF Syntax and Model specification or the Schema Specification, only the essential parts.

The syntax used in the W3C specification uses XML for the reason that one of RDFs goals is to make it possible to specify semantics for data based on XML in a standardized, interoperable way. XML syntax is however only one possible syntax for RDF. RDF data model is possible to represent in any other (future) syntax.

4.6.1 RDF Development

RDF is a collaborative design effort; any individual or organization did not invent it. Several W3C Member companies have contributed intellectual resources to its development (Miller, 1998). Although RDF started as an extension of the PICS content description technology, HTML metadata, SGML and most of all XML have also influenced it. Thus, RDF is a further layer on top of XML (see figure 4.3). Also object-oriented programming and modeling languages have had an influence. Other documents, such as Microsoft's XML-Data paper and the Dublin Core Framework have also had an effect on the RDF design.

RDF is a great step forward from specifications such as HTML and Hyper Text Transfer Protocol (HTTP). These specifications have been extended significantly during their evolution. The process has been one of the experimental additions of new tags.

The significance or importance of new tags has never been evident to software not involved in the experiment, with resulting danger of serious incompatibility. RDF is intended to allow documents to be written in both old standard vocabularies and in specific new experimental or proprietary vocabularies. This has to be however done in a well-defined way of knowing what is important, what can be ignored and how old software can deduce or download an understanding of a new vocabulary, which will hopefully allow powerful combinations. By allowing experimental vocabularies to intermix with standard vocabularies, RDF will give a faster deployment path for new ideas (W3C, 1997a).

RDF does not have a long history yet. First published note from W3C was a press release in October 1997. It announced "the first public draft of a work-in-progress of the Resource Description Framework (RDF), providing interoperability between applications that exchange machine-understandable information on the Web" (W3C 1997). According to the press release, the RDF Working Group was one of the earliest phases of a major effort to build a vendor-neutral and operating system-independent system of metadata. One of the other first documents concerning RDF is a W3C note dated 13 November 1997. It was made for discussion only, and includes a brief introduction to the RDF and the concept of metadata. First public draft of the syntax and model specification was released in February 1998. First public draft of RDF schema specification was released in April 1998. The current version of the RDF Model and Syntax Recommendation is dated 22 February 1999, current RDF Schema Proposed Recommendation 3 March 1999.

4.6.2 RDF Model

The foundation of RDF is a model for representing named properties and property values (W3C 1999b, 3). It therefore resembles an entity-relationship diagram. In object-oriented design terminology, resources correspond to objects and properties correspond to instance variables.

As W3C expresses (W3C 1999b, 3), the RDF data model is a syntax-neutral way of representing RDF expressions. The basic data model consists of three object types: *resources*, *properties* and *statements*. All objects described by RDF expressions are called *resources*. A resource may be, for example, a Web page or an XML element. *A property* is a specific aspect, characteristic, attribute or relation used to describe a resource (W3C 1999b, 4). Each property has a specific meaning. *A statement* is a specific resource combined with a named property value of that property for that resource. These three parts of a statement are called *the subject*, *the object* and *the predicate*. An example of their relationship is shown in figure 4.3. The figure shows the sentence *"John Doe is the creator of the resource "http://www.johndoe.com"* in a diagram. The sentence has the following parts:

Subject (Resource):	http://www.johndoe.com
Predicate (Property):	creator
Object (Literal):	"John Doe"

In the figure, nodes (ovals) represent resources and arcs represent named properties. The node that represents string literal is drawn as rectangle.

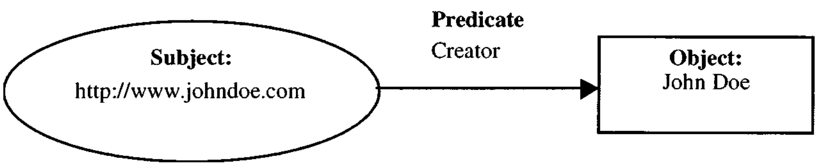


Figure 4.3: An RDF statement expressed in node and arc diagram (based on W3C 1999a, 5)

It is notable that the direction of the arrow is important. The arc always starts at the subject and points to the object of the statement. The diagram above may also be read "http://www.johndoe.com has creator John Doe", or in general "<subject> HAS <predicate> <object>".

As was mentioned above, also object-oriented programming languages and design have had an influence on the development of RDF. Figure 4.4 shows the mapping between RDF and object-orientation. RDF model subjects can be seen as object-oriented modeling objects, predicates as attributes and RDF objects as attribute values.

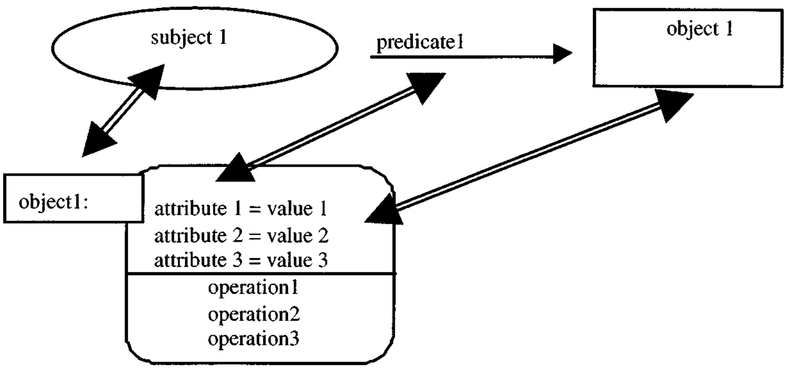


Figure 4.4: Mapping between RDF and object-orientation

4.6.3 RDF Syntax

As we indicated in the previous section, RDF data model provides an abstract and conceptual framework for defining and using metadata. Syntax is needed in RDF for creating and exchanging the metadata defined in RDF data model. The syntax presented in this chapter is XML serialization syntax (see W3C 1999, 7). XML has also an abbreviated syntax, and RDF interpreters are expected to implement them both. The XML serialization syntax is selected here because according to the RDF Model and Syntax Specification, it expresses the full capabilities of the data model in a very regular fashion. In the following the basic elements of RDF/XML syntax are introduced.

It is common in RDF that several properties of a resource need to be given together. In RDF/XML syntax, this can be done by grouping multiple statements for the same resource into a `Description` element. Thus, typically there will be more than one statement made about the resource. The `Description` element names in an `about` attribute the resource to which each of the statements apply. The `RDF` element simply marks the boundaries in an XML document. The example sentence and figure 4.3 in the previous chapter is represented in RDF/XML as:

```
<rdf:RDF>
  <rdf:Description about="http://www.johndoe.com">
    <s:Creator>John Doe</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

Property names must be associated with a schema. Association can be done by adding a namespace prefix or by declaring a default namespace. Here the namespace prefix "s" refers to a specific namespace prefix chosen by the author of this RDF expression. It is defined in an XML namespace declaration. RDF uses the XML namespace facility in order to avoid confusion and conflict in the use of same term. The complete XML document of the same example would be:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org.schema/">
  <rdf:Description about="http://www.johndoe.com">
    <s:Creator>John Doe</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

It is often necessary to refer to a collection of resources, for instance, software modules in a package. RDF containers are used in handling such lists of resources or literals. RDF defines three types of container objects: a bag, a sequence and an alternative. Object-oriented programming language Smalltalk has similar kind of selection of container classes. Smalltalk has also a class called `set`, which is same as `bag` but with no duplicates. In the future, RDF Model and Syntax Specification may define a class like `set`.

Table 4.3: Container types in RDF (W3C 1999b, 13)

Container	Definition
Bag	An unordered list of literals. Bags are used to declare that a property has multiple values and that there is no significance in their order. Duplicate values are permitted.
Sequence	An ordered list of resources or literals. Sequence is used to declare that a property has multiple values and that their order is significant. Duplicate values are permitted.
Alternative	A list of resources or literals that represent alternatives for the (single) value or a property. An application using a property whose value is an Alternative is aware that it can choose any one of the items in the list as appropriate.

Containers are commonly used in expressing the value of a property. The benefit of using containers in such a way is that a statement still has a single statement object regardless of the number of members in the container. The container resource itself represents the object of the statement. Thus, when representing a collection of resources, RDF uses an additional resource that identifies the specific collection, that is, an instance of a collection. This resource must be declared an instance of one of the container types (a bag, a sequence or an alternative). Figure 4.5 shows the sentence *"John, Mary and Tim are the creators of the resource "http://www.johndoe.com"* in a node and arc diagram.

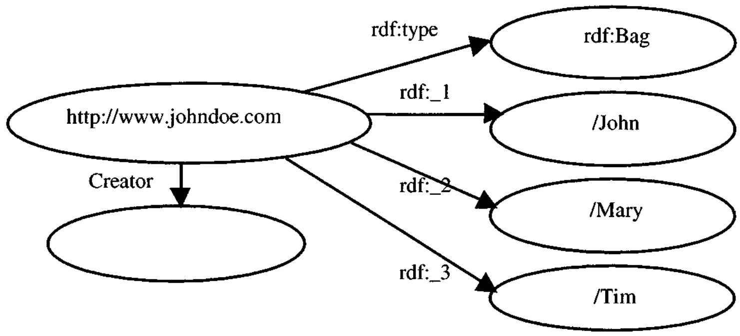


Figure 4.5: Simple bag container (based on W3C 1999, 14)

4.7 RDF Schema

Like in all human written documents, also in RDF applications it is important that both the reader and the writer understand the statement or statements the same way. Meaning in RDF is expressed through reference to a schema. RDF schema is described in this section. Schema can be seen as a dictionary that defines the terms that will be used in RDF statements and gives specific meanings to them (W3C 1999b, 12). Many different schemas can be used with RDF, including the schema defined in RDF Schema Specification (W3C 1999c). It has some characteristics that help automating tasks when using RDF. The RDF Schema Specification schema is included in this thesis in appendix C. Therefore; the details in that schema are not further discussed here.

RDF schemas might be contrasted to XML DTDs. RDF schema is however wider and not as restricted as a DTD, which gives specific constraints on the structure of a document. RDF data model does not provide a mechanism for declaring the properties nor defining the relationships between the properties and other resources. RDF schema is aimed at these purposes. Thus, RDF schema provides information of the interpretation of the statements given in a RDF data model (W3C 1999c, 3).

The RDF Schema Specification defines the RDF schema that can be used in describing properties of other RDF resources, which define application specific RDF vocabularies. The core schema vocabulary defined in the RDF Schema Specification uses a namespace informally called '`rdfs`' because a formal URI has not been defined yet. A formal URI will be given in the final version of the specification.

To quote W3C, their RDF Schema Specification is "not aimed at theoretical issues, but at solving a small number of immediate problems" (ibid.). The RDF Schema specification was directly influenced by consideration of problems concerning the PICS, description of Web pages, sitemaps and the P3P project.

The RDF Schema corresponds well to the concept of schema presented earlier in section 4.3. As in the definition of Arbib et al., each part of RDF schema corresponds to an object in a real world. RDF schemas can also be combined to form new schemas. One example of such use is presented later in this chapter in section 4.8. In network management, all network elements have to be separately defined in a schema. This means also, that the *instantiating* and *deinstantiating* Arbib et al. mentions are possible in RDF schemas. Instantiating in RDF schemas can be seen as having multiple copies of schemas for multiple purposes. In network management, this can mean, for instance, different schemas for processing different network element's management information. Deinstantiating correspondingly refers to the fact that schemas or parts of schemas can be "turned off", that is, parts of schemas can be taken off the schema. This can have significance, for example, in a situation where some network element does not exist anymore. Schemas can also in some sense be formed into groups of active schemas, that is, some schemas may not be in use all the time. RDF schema is also both the process and the representation: it contains both the declaration of the resources it refers to, and the instructions of what to do with the resource related data.

4.8 Example of XML and RDF Schema Usage in Performance Management

We assume that XML files can be generated in network elements. The transferring of performance data files from network elements to the consumer in network management could be done, for instance, by FTP (File Transfer Protocol). The usage of FTP in file transfer is possible because the data communication network that connects network elements to operations systems (see figures 3.1 and 3.4) is based on TCP/IP (Transmission Control Protocol/Internet Protocol) techniques. Binary measurement files that are used in GSM networks nowadays are very compact but it is nearly impossible to interpret the contents of measurement files. However, structured documents allow the interpretation of the document content.

One possible solution for files is Open Measurement Standard (OMeS), which is a format that has been studied by Nokia. OMeS can be used to represent performance data in most or all telecom technologies as it is passed through data interfaces of a NMS (Network Management System). OMeS is an effort to harmonizing the presentation of measurement results from different mediators, that is, network elements. OMeS comprises of three components: *data model, service model* and *data format.* (Pohja, Muurinen & Lassila 1999, 1-2) The actual implementation of OMeS files, that is, the data format could be done in XML and the RDF or XML schema could be used as validating schema.

The OMeS *data model* restricts the contents of the performance measurement file by defining the possible fields that are allowed when constructing the OMeS file. In this way the appropriate content of the file can be ascertained in the decoding end. (Pohja et al. 1999, 2) Actually, the same kind of data model (see 4.6.2) has been defined in RDF.

A performance measurement is described as a five-tuple: *context, target, property, setup* and *result* in the abstract data model. Context identifies the semantics needed to interpret and to verify the other four parts. Target identifies an object, when given the context, for instance, a network element. Property identifies which attribute of the target was measured, when given the context. Attribute can be, for example, a counter of a network element. Setup identifies and describes the measurement act, when given the context. Setup can identify, for instance, the time when the particular measurement was measured. Result is the outcome of the measurement. The first four elements identify the performance measurement and each performance measurement may have only one outcome. (ibid.)

An OMeS producer transfers performance measurement results to an OMeS consumer. A mediator acts as the producer. The consumer is in the manager, which is the NMS. The three-layer model describes the producer and the consumer. The layers are *application layer, semantics layer* and *coding layer* in the order from the top down. This OMeS *service model* is also described in figure 4.6.

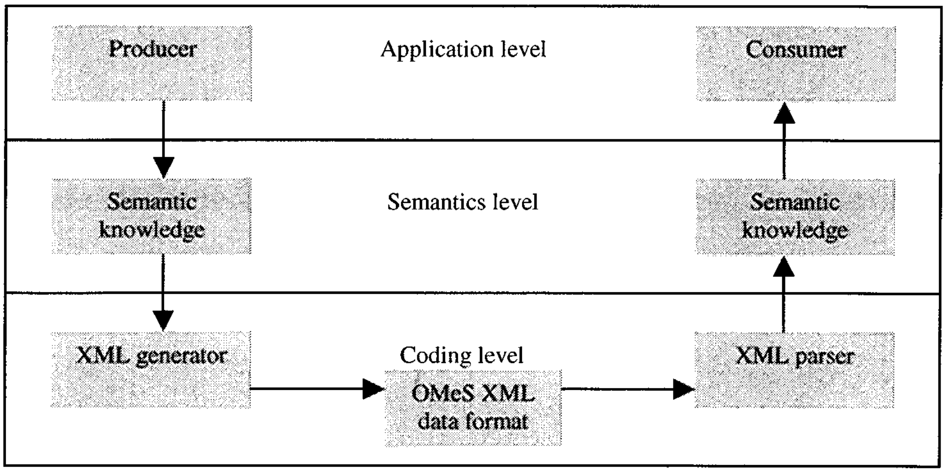


Figure 4.6: OMeS service model

The *application layer* of the producer communicates measurement results to the corresponding layer of the consumer by using the services of the semantics layer. There is no direct relationship between the presentation of data in the application layer of the producer and the consumer, because the functions they perform should not be fixed. The *semantics layer* of the producer communicates measurement results to the corresponding layer of the consumer by using the services of the coding layer. The semantics layers of the producer and consumer both have the knowledge about the semantics identified by the context element in the 5-tuples in the OMeS data model. The semantics layer validates the data against the semantics. The semantics can be represented in XML or RDF schema. The *coding layer* writes the validated data onto a medium. The OMeS/XML data format can be used as the medium. The coding layer of the consumer reads the data from the medium. The semantics layer parsers the data into data structures according to the semantics.

XML files can be used also in "upper levels" of performance management. By upper levels we mean the levels that have an interface to human users as well as CORBA interfaces and databases. It would be possible to provide XML data from the relational database to the users who are interested in performance measurement data. For example, Oracle8i database solution supports XML, because XML is believed to increase scalability and cost-effectiveness of applications. CORBA has also an important role in data transfer in upper levels. XML files could be easily transferred in CORBA environment.

Figure 4.7 presents OMeS in an object model. The figure consists of five classes. Class *FooPMTarget* is the subclass of class *PMTarget.* Class *BarFNE* is subclass of FooPMTarget. BarFNE is an example of network element and is not defined by the OMeS specification. Class *PMSetup* consists of one PMTarget. Classes PMSetup and PMTarget are abstract classes. Class *FooPMSetup* is a subclass of PMSetup.

We can directly see from figure 4.7 some requirements for the schema. First, the figure shows a clear hierarchical structure, which allows inheritance in OMeS. With an appropriate schema it should be possible to represent sub-classes with inheritance of attributes and elements from the upper to lower classes, like from FooPMTarget to BarFNE. However, sub-classes should have their own additional attributes and elements, for instance, because new network elements in UMTS can have more built-in intelligence. This means that they can control their own functionality more extensively than second generation network elements. The own attributes in the sub-classes produce extensibility.

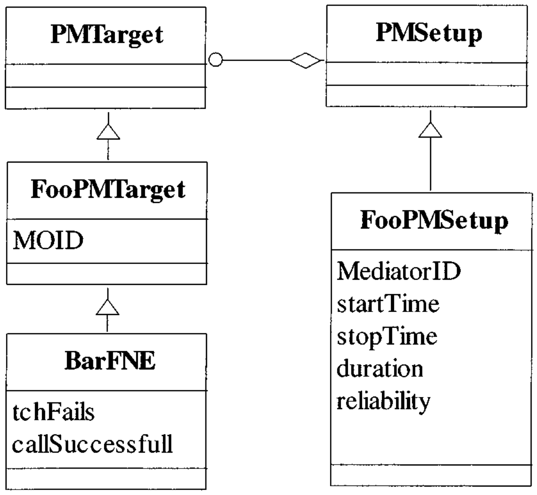


Figure 4.7: OMeS object model (based on Nokia Networks 1999, 7)

Next, the appropriate schema should have means to constrain classes, attributes and cardinalities. It must be possible to constrain classes to have only specific attributes, especially in upper levels of hierarchy. The schema should also constrain the values of attributes to certain data types, at least primitive data types such as integers should be included. It should also have a possibility to represent cardinality within attributes. It should be possible to define that an attribute can have zero, one or multiple values, for example, that startTime can have only one value.

The schema requirements presented above have correspondence to Lyytinen's language context and denotational theory. The denotational theory also focuses on the specification of the conceptual schema. The conceptual schema in the denotational theory models the abstraction system. The conceptual schema presents formally the contents of the abstraction system and also defines the states of affairs that can occur in the object system. (Lyytinen 1986, 191)

Lyytinen (1986, 192-193) presents researchers' criteria for evaluating conceptual schema, for instance, stability, simplicity, uniformity, completeness and uniqueness. He also states that schema has to be able to describe logical relationships between elements. This corresponds to our requirements of hierarchy and inheritance. Other requirement Lyytinen (1986, 194) mentions is one of canonical structure. By the canonical structure he means a structure that is internally consistent and reflects correctly the Universe of Discourse. From these it follows that two sentences in the conceptual schema can in no circumstances logically contradict each other. This consistency requirement concerns all schemas, including XML and RDF schemas.

In the following example is figure 4.7 presented in XML and RDF/XML. The class BarFNE is not presented in this example because it is only an example of network element and is not defined by the OMeS specification. OMeS specification gives however guidelines on how to create definitions of such classes. All network element types, which produce performance management data to a network management system, have their own classes. The class BarFNE is therefore presented in appendix D in both XML and RDF/XML. There are also example measurement files in appendix E that are based on the object model in figure 4.7. in both XML and RDF/XML.

Example 1: OMeS object model presented in both XML and RDF/XML

OMeS object model in XML:

```
<!—Abstract class for PMSetup -->
<archetype name= 'PMSETUP' model='refinable'/>

<!-- Base class for FooPMSetups -->
<archetype name='FooPMSetup' model='refinable'>
  <refines name='PMSetup'/>
  <!-- The network element identification -->
  <attribute name='MediatorID' required='false' type='MediatorID'/>
  <!-- The time when the interval started -->
  <attribute name='startTime' type='dataTimeUTCDiff'/>
  <!-- The length of the interval in minutes -->
  <attribute name='duration' type='minutes'/>
  <!-- The startTime added with duration -->
  <attribute name='stopTime' required='false' type='dateTimeUTCDiff'/>
  <!-- The reliability value of measurement data -->
  <attribute name='reliability' required='true' type='boolean'/>
</archetype>

<!-- Abstract class for PMTarget -->
<archetype name='PMTARGET' model='refinable'/>

<!-- Base class for FooPMTargets -->
<archetype name='FooPMTarget' model='refinable'>
  <refines name='PMTarget'/>
    <!-- The managed object identification -->
    <attribute name='MOID' type='MOID'/>
</archetype>
```

(based on Nokia Networks 1999, 8)

OMeS object model in RDF/XML:

```
<Class rdf:ID="FooPMTarget"
  comment="Base class for Foo OmeDM Targets"/>
  <subClassOf rdf:resource="&om;PMTarget"/>

  <Property rdf:ID="MOID"
      comment="The managed object identification">
      pri:minCardinality="1"
      pri=maxCardinality="1"
      <domain rdf:resource="#FooPMSetup"/>
      <range rdf resource="&om;MOID"/>
  </Property>

</Class>

<Class rdf:ID="FooPMSetup"
  comment="Base class for Foo OmeDM Setups">
  <subClassOf rdf:resource="&om;Setup"/>

      <Property rdf:ID="MediatorID"
        comment="The network element identification">
        pri:minCardinality="1"
        pri=maxCardinality="1"
        <domain rdf:resource="#FooPMSetup"/>
        <range rdf resource="&om;MediatorID"/>
      </Property>

      <Property rdf:ID="startTime"
        comment="The time when the interval started">
        pri:minCardinality="1"
        pri=maxCardinality="1"
        <domain rdf:resource="#FooPMSetup"/>
        <range rdf resource="&pr;Date"/>
      </Property>

      <Property rdf:ID="duration"
        comment="The length of the interval in minutes">
        pri:minCardinality="1"
        pri=maxCardinality="1"
        pri:minimum="0"
        <domain rdf:resource="#FooPMSetup"/>
        <range rdf resource="&pr;Integer"/>
      </Property>

      <Property rdf:ID="stopTime"
        comment=" The startTime added with duration">
        pri:minCardinality="0"
        pri=maxCardinality="1"
        <domain rdf:resource="#FooPMSetup"/>
        <range rdf resource="&pr;Date"/>
      </Property>

      <Property rdf:ID="reliability"
        comment=" The reliability value of measurement data">
        pri:minCardinality="0"
        pri=maxCardinality="1"
        <domain rdf:resource="#FooPMSetup"/>
        <range rdf resource="&pr;Boolean"/>
      </Property>

</Class>
```

As a conclusion to the description of XML and RDF and schemas we can state that XML can be used in many different manners. Thus, XML can be seen as a versatile and powerful meta language. In XML schema, data has a hierarchical structure, which is in some sense a traditional way of presenting data. In RDF/XML documents, data and document structure have an object-oriented nature. This feature can have an advantage in object-oriented environments. The differences between XML and RDF schemas as well as their applicability to performance management information processing are discussed in the next chapter.

4.9 Summary

In this chapter, we introduced different characteristics, functions and types of metadata. We tried to clarify the relevance of metadata to the network management. We also discussed the importance of metadata to organizations. We stated that due to the increased amount of services and interest groups the need to social activity is emphasized in UMTS. We saw that metadata and metadata applications offer better visibility to the system and thus offer better possibilities to provide the services different parties are interested of. Metadata applications thus take social issues well into account.

We shortly described the development of XML. We presented basic structure of XML language specification and XML schema specification. In appendix B is presented a (normative) schema for XML schemas. We also briefly described the development of RDF and introduced the basic concepts of the RDF model and syntax according to the RDF Model and Syntax Specification. Also, the RDF schema principles were introduced according to the W3Cs RDF Schema Specification. We also evaluated, how the concept of schema based on Arbib et al's theory corresponded to XML and RDF schemas.

In the last section of this chapter, we introduced an example of XML and RDF schema usage in network management. As an example, we introduced Nokia's Open Measurement Standard, which can be used in most or all telecom technologies. We also showed, how OMeS object model could be presented in XML and RDF.

In the next chapter, we will evaluate all the issues introduced in our thesis. We will make an effort to answer all the research problems posed in the introduction. At first, we will evaluate the issues that have had an impact on the increased need of social activity in UMTS. We will also evaluate how UMTS network management systems allow the activity of the social action types and how different types of metadata and the social action types relate. At last, we will evaluate how XML and RDF schemas differ in presenting data in the performance management context.

5 EVALUATION

So far, we have represented all components of our research. We have introduced Habermas' critical social theory and its significance to information systems development. We have also introduced Telecommunications Management Network framework and a specialization of it: Universal Mobile Telecommunications System. In chapter 4, we described metadata and examples of a meta language - Extensible Markup Language, and a metadata application - Resource Description Framework.

In this chapter, all of these introduced concepts and issues are evaluated. We will show how developments of UMTS and UMTS network management can be evaluated with the help of the three object system contexts. After that, we will point out the issues that have had an impact on the increased need of social activity in UMTS, thus why CST can be used in the examination of UMTS. The function of UMTS network management systems with the consideration of the four social action types is discussed next. Different types of metadata and their relationship to the four social action types are also discussed. XML and RDF will be evaluated last with the help of requirements presented in section 4.8.

5.1 Critical Social Theory and UMTS Network Management System Development

In this section, we try to emphasize the parts of CST that have significance in UMTS development, and UMTS network management systems development. We will also discuss the three object system contexts and the super-contexts identified in ISDMs from the UMTS viewpoint. This section will also act as a motivation to our evaluation of XML and RDF schema usage in UMTS network management systems.

The change process in UMTS, and thereby in UMTS network management systems, has to be seen as a derivation from the development organization. In the case of UMTS, the development organization is the organization involved in the standardization of UMTS.

This organization refers to many parties like ETSI, 3GPP and ITU as well as independent manufacturers like Nokia and Ericsson. The reason why development organization is the organization behind the standardization of UMTS also in the case of UMTS network management systems development is that UMTS is highly standardized system. The development of network management systems for UMTS is abundantly guided by the standards of UMTS. It can be said, that the UMTS development organization *defines* the UMTS network management system. Therefore, the actual development process has been done within the organizations or parties that were involved in the UMTS development. There are also parts of UMTS that are not standardized, but are left to be the designers' and developers' choice. In our thesis, we have however decided to take an approach, which sees the UMTS development and standardization organization as the developing organization in UMTS network management system development.

Because we have pointed out that the development of UMTS network management system is derived from the organization, we can now explain the impacts on other two object system contexts. The change in organization context affects to the language context in a way that it creates the language used in the language context. In our case, this means that the organization context defines the way of modeling the system in the language context. To the technology context, the change in organization context means that the technology has to adapt to the changes made in the organization context (cf. Lyytinen 1986, 88). In our case this means that the standardization decisions that affect, for instance, applied technologies change the technology context in a way that the technology used has to adapt to the changes made. For example, changes in application protocol level and in the standardization of valid application protocol suites can be seen as changes that affect the technology context.

In table 5.1 we present the classification of development organizations by social action type. From the table, we can see that all types have actions carried out during the UMTS development process. We will next comment on some of the entries in the table. In the communicative action type, especially development by dialogue comes into significance. When developing a global system like UMTS, the dialogue and communication ways are in a very important role. Different interest groups have to especially discuss of the services and achieve understanding of them. Different interest groups have to also discuss and agree on their needs, which can be contradictory. In the normatively regulated action type, the legitimation of norms is important. In addition, the conformism to norms is important in the UMTS development. All developing parties have to accept and conform to the norms. In the strategic action type, significance of negotiation is emphasized. This means that different parties try to achieve their own, sometimes conflicting strategic objectives by negotiations. In the instrumental action type, the relationship between developers and clients is important. This can be interpreted as user participation during the development of UMTS. In UMTS, the users' needs have been taken into account better than in the second generation systems development. This is partly due to the fact that services and service management is emphasized in UMTS (cf. 3.8.2).

Table 5.1: Classification of development organizations by the action type (Lyytinen 1986, 163)

Action Type	Development organization
Instrumental	Division of labor, scientific management, expert-client relationship
Strategic	Administrative conflict resolution mechanisms, negotiation
Normatively Regulated	Organization based on hierarchical authority, conformism to norms, legitimation of values and norms by myths and rituals
Communicative	Forms of life and ethnomethodological accounting practices, discourse participation, understanding tradition, development by dialogue

Based on above information, we can also clarify the mappings between the super-contexts in the case of network management systems development (cf. 2.4.1). In the organization and language super-context exists an association, which comes from the organization context to the language context. The organization context enables the modeling of the system in the language context. This can mean, for example, a data model, that models and describes the system and its architecture. In the language and technology super-context the language context, and the data model in it, enables the use of XML and RDF in a reasonable way. In the organization and technology super-context, the standardization enables, for example, the use of CORBA or other middleware that enable distribution. Consequently, CORBA enables and supports the use of XML and RDF so that it makes it possible to the transfer XML files in a distributed environment.

5.2 The Function of UMTS Network Management and Social Action Types

In this section, we will briefly recap the purpose of network management and discuss the issues that have had an impact on the increased need of social activity in UMTS. We will also evaluate how different social action types have been taken into consideration in UMTS network management systems and especially in performance management, and how the UMTS network management related standards enable the use of metadata. According to Hirschheim, Klein and Lyytinen (1996, 12), actors change their orientations during the ISD and that all action types are present is ISDs. This means that we will evaluate how UMTS network management system enables different kinds of actions, for instance, strategic actions.

According to the traditional definition, the purpose of a network management system is to control and monitor the status and performance of the network. This includes measuring the network's performance, and gathering and analyzing the measurement data. Network management also includes the applications, which are supposed to minimize the impacts of possible faults in the network. Network operator benefits from efficient network management in many ways. One benefit is better quality in the services and consequently growing amount of users, which in turn increases income (cf. 3.1). The improved quality in the services thus directly increases the income. In UMTS, the significance of the services is emphasized. Service management as a part of performance management is thereby a very important part of network management as a whole.

As we stated before, the need of social activity is emphasized in UMTS due to the increased amount of services and interest groups. The different interest groups can have various needs concerning especially the services. The achievement of agreement needs communication, which has to be seen as social activity. Traditionally the function of network management system has been technical control whose only functionality criteria has been efficiency. In UMTS this situation is changed both because of the various interest groups and the increased amount of services. We see that the social aspects will thus be taken into consideration more extensively. The functionality criteria can be based on social aspects instead of efficiency.

The mutual understanding of the various needs and functionality criteria of the different interest groups can only be achieved by negotiation, that is, communicative activity. For example, functionality criteria can be based on how well the system serves the operator's or service provider's information needs about failed calls or handovers. Then the functionality criteria would not at all be based on the network management system's efficiency or information processing performance, which has been the traditional way of measuring the system's overall performance. Traditionally, information system's functionality has been tested by performance tests and other possible criteria have been ignored. For example, measurement files that are based on OMeS/XML data format may not have competitive transfer rate compared to the binary files in second generation systems, but OMeS/XML data format offers improved possibilities in the information usage. Thus, in the case of OMeS/XML, there can be a contradiction between the absolute efficiency of the system and the quality and value of the information.

In UMTS, the dominant functionality criterion, for example algorithmic efficiency versus the quality of the provided services, have to be agreed on by negotiations. The network management system is then developed to measure up the selected criterion. This may result a system, whose performance is not improved compared to the existing systems, but who has improved services and service quality to offer. A network management system is never developed based on only one criterion but some criteria are contradictory in a way that they demand a selection between them.

For the reason that systems, including also other systems than network management systems, have always been evaluated based on their effectiveness, it is not at all self-evident that all interest groups achieve agreement effortlessly. The interest groups pursue for their individual and possibly conflicting interests. During the development, there can be continuous negotiations of the purposes of the system. In these negotiations, the agreement has to be achieved. The more interest groups there are, the more conflicting opinions can occur and thus the more communication is needed. The same applies to the amount of the services.

In network management systems and in performance management's functional area, communicative actions have been taken into consideration in a way that communicative actions are needed in communication situations (cf. table 2.5). Communicative situations are various in network management systems and circumstances related to it. If we for instance consider the agreement on the services we can see that it requires achievement of mutual understanding, which is communicative activity. Also if we consider the system itself and the transferring and use of performance measurement files, we can say that the possible use of metadata is communicative action. The use of metadata as communicative action also implements the objective of communicative action: mutual understanding. Metadata is aimed at better understanding and compatibility in UMTS network management system, which has to be seen as a distributed and a large system. Metadata and metadata applications are supposed to help the interpreting of measurement files and also ease the transferring of the files in the system.

Normatively regulated action type is essential in UMTS network management systems. By the norms, the system is partly specified and the possible use of the system is also restricted and guided. UMTS network management system allows normatively regulated action so that it allows different kinds of rules or constraints to be implemented in the system. Norms are needed everywhere in the system, for example, in defining valid application protocols and different roles in the system. Different roles are especially important in the system, as well as the interactions between them. In UMTS, there are different actors, who have some kind of access to the network management system. Network operators are one, manufacturers are another and also the end-users of the services have access to the system. A matter of importance is who gets the information network management system processes.

The control over data comes into special significance in the strategic action type. By the use of norms, some strategic use of the system is restricted. Some parts of UMTS network management system are advisedly left un-standardized to allow the influence of markets (see e.g. 3.7). The normatively regulated action is however made possible in UMTS network management systems; operators can norm the system almost the way they want. The parties in UMTS development have already done some normatively regulated action considering the network management system. Those parts are defined by standards. One standardized part is valid management application protocol suites, one of which is CORBA. As was discussed earlier in sections 3.7.1 and 4.2.1, CORBA makes the use of metadata profitable as it enables the transferring of XML files in a distributed environment.

Strategic action type typically dominates in the beginning of any ISD. By normatively regulated actions, the strategic use of the system is at least partly prevented. This does not mean that systems cannot be strategic or that they cannot have strategic purposes. In a global system like UMTS, the norms still have however a special importance. By the norms, standards and standardized parts, the system is meant to become really global and universal. If much influence and room is left for the strategic action, the system may not function globally but as a system of some individual factor.

In strategic actions, the control over data is important (see 2.3.2). Laws regulate some of this control. Network management system's operator or manufacturer cannot thus control all the use of the data. Although the owner of a network management system owns the collected data, he cannot use the data totally as he wishes. We have to however remember that the data collected in performance management is mostly statistics and does not concern any individual network user. Thus, who gets the data does not have to be strictly regulated. As we can see in table 2.2, the purpose of a strategic action is pursuit of individual interest. With metadata and metadata applications, this pursuit becomes easier as with appropriate schema, operator can collect data that interests him more easily. The data is also easier to interpret as XML or any other meta language can comprise and present also the meaning of the data.

Instrumental action type is what characterizes all information systems. Traditionally, information systems are seen as technical systems, whose purpose is technical control. These kinds of systems are also typically evaluated by their effectiveness. With the impact of some of the UMTS standards, the systems are intended to become (more) effective. Also, by not standardizing some parts, it is assumed that new implementations may develop the effectiveness even more. It is obvious that the ultimate purpose of a UMTS network management system is technical control. This is not however the only purpose of the system in the way it was in second generation systems. Thus, it is evident that instrumental actions are needed in the development and use of the system. We also assume that the possibility of using metadata and metadata applications in UMTS network management system improves the quality and effectiveness of the technical control.

The features of social action types in UMTS network management and especially performance management are gathered in table 5.2. Activities of all four social action types and features that characterize the actions are presented in the table.

Table 5.2: Social action types in UMTS performance management

Feature Covered	Instrumental Action	Strategic Action	Normatively Regulated Action	Communicative Action
Activity in UMTS performance management	Gaining improved technical control by the use of metadata	Gaining strategic advantage or information with appropriate schema	Defining the system, constraining the system's use	The use of metadata and schemas
Characteristic feature	Characterizes all ISs	Restricted by the norms, standards and standardized parts	Restricts strategic use	Aims at better understanding with the use of metadata

5.3 Social Action Types and Different Types of Metadata

We have now demonstrated how UMTS network management system and performance management enables activity of different social action types and how the possible use of metadata relates to these actions. In this section, we will evaluate the different types of metadata based on Gilliland-Swetland's division presented in table 4.1 and the social action types and how they relate. The different types of metadata are administrative, descriptive, preservation, technical and use. Each of them is described separately in the following section.

In our thesis, administrative metadata type means metadata that is used in managing and administrating the network. Administrative information is important to network operators because they need to know the actual performance of the network compared to the set objectives. Administrative metadata refers to the information in schema that relates, for example, to successful and failed calls in particular network element. In schema, the collected information can be defined and thus administrative information is easy to obtain with appropriate metadata in the schema. We suggest that administrative metadata relates to three social action types: strategic, normatively regulated and communicative. Normatively regulated actions are needed because the norms according to which the data is collected have to be decided upon. This means that with the help of norms, the important information is collected from the network. Not all information can be considered as relevant to administrative needs and thus the information has to be validated by norms. This is also where strategic actions take place. By strategic actions, different parties try to achieve their own, sometimes conflicting administrative needs. These strategic actions have to be however restricted by the norms to achieve mutual agreement on the administrative purposes. Mutual agreement again refers to communicative actions.

With descriptive metadata type we refer to metadata that describes and identifies network elements and their functions. It is distinctively made by people. An example of descriptive metadata in performance management and in our thesis is the object model in figure 4.7. The schemas, which can be considered as descriptive metadata, are made based on the object model (see examples in 4.8). We suggest that descriptive metadata relates to two social action types: communicative and instrumental. Descriptive metadata is communicative because it describes the system and the information in it and is therefore communicative by nature. Instrumental activity relates to descriptive metadata in a way that well defined system and the information in it serves efficient intervention (cf. table 2.5). Well-defined and algorithmic descriptions of the system facilitate efficient use of the system.

Preservation type of metadata relates, according to Gilliland-Swetland (1998, 7), mainly to databases. In our thesis, we have not largely discussed the data storing in UMTS network management. However, XML files could be stored directly to databases. This would further increase the efficiency of the system. We claim that preservation relates to two social action types: communicative and strategic. Because the information in a database can be described through schemas, the preservation can also be seen as communicative action. Preservation is also a strategic activity for two reasons. First, by storing essential information in the database strategic advantages can be achieved. Secondly, if the information is easy to retrieve from the database, this can produce some strategic advantage. We assume that if the database consists of XML files, the retrieving becomes easier. Both of these strategic actions are made possible by preservation metadata.

In our thesis, technical metadata type relates to the functioning of the network management system. Technical metadata has similarities to administrative and descriptive metadata types. Its purposes are not however the same. It mostly describes the functions of subsystems and even individual software modules. It differs from the administrative metadata in the way that is does not select the information in a way administrative metadata does. Technical metadata is used to ensure that the whole system functions the way it is designed to. We see that technical metadata has a strong relationship to instrumental action type. They both aim at an effective and well-functioning system. Technical metadata is the only type of metadata, which has no relationship to communicative action. This is because technical metadata is non-social by nature as instrumental action.

Use type of metadata refers to who gets the performance management information. In this, the different roles of users come into significance. As we stated in the previous section 5.2, in network management and performance management several users have access to the system. This access has to be controlled. In our thesis, we have not largely considered this area. We however assume that by metadata these accesses can be controlled. We suggest that use type of metadata relates to three social action types: normatively regulated, strategic and communicative. The relationship to the normatively regulated action is obvious based on the previous section. Normatively regulated action type also relates to the different roles of actors and to who gets the information from the system. The control over data is also important in the strategic action type. By controlling different user's accesses to the system, strategic advantages can be gained. Communicative action type relates to the use type of metadata in a way that the levels and types of users and the metadata descriptions of them are communicative descriptions of the organization. The use type of metadata thus describes the organizational reality (cf. table 2.5).

In table 5.3, we have gathered the characteristics of different types of metadata in performance management and the social action types they relate to.

Table 5.3: Social action types and different types of metadata

Feature Covered	Administrative	Descriptive	Preservation	Technical	Use
Character-istics of the metadata	Metadata used in managing and administrating the network. E.g. specific information in schemas.	Metadata that describes and identifies network elements and their functions. E.g. object models.	Metadata in database descriptions.	Metadata related to the functioning of the network management system. E.g. descriptions of subsystems.	Metadata related to the use of the system.
Related action types	Strategic Normatively regulated Communicative	Communicative Instrumental	Communicative Strategic	Instrumental	Normatively regulated Strategic Communicative

As we stated in section 5.2, the use of metadata in performance management is most of all communicative action. From the previous discussion, we can conclude that the communicative action type is clearly dominating in metadata types. The technical type of metadata is the only one in whose definition and functions we do not see communicative action as important part. Traditionally, performance management in network management systems is seen most of all as a technical and normative system.

Based on the evaluation above we can state that with appropriate metadata solutions, performance management becomes very communicative by nature. It is with metadata able to define and describe itself in all levels. We can even say that its technical parts become invisible. Only the broker objects interact with each other. They do not have to know each other's implementation or structure. Performance management is communicative by nature also because it serves better all the parties that are interested in its functions and in the information in it. This interest is emphasized in the services and in the service related information.

As was stated in 2.3.4, the communicative action type emphasizes language and organization contexts and that no system management can proceed without communication. Because the language context explains how and why the sign-relation is established and what effects it can have, it has special relevance in performance management. According to Lyytinen (1986, 81), the fundamental problem in the language context is the problem of meaning: what it is, how it is created and interpreted. We claim that this problem of meaning can be at least partly solved by metadata solutions. In the next section, we discuss these solutions: XML and RDF schemas.

5.4 XML and RDF Schemas in Performance Management

As we defined in section 4.3, XML is a meta language that corresponds to the definition of a meta language in the denotational theory. According to the definition, the object system and the language together form the Universe of Discourse for the meta language. In the last section, we also concluded that different metadata types relate to the communicative action type. From this it follows that schemas used in network management are communicative by nature. Schemas enable developers to interpret new technology, for instance, by helping them to model new network elements. With the help of schemas it is possible to construct and make sense of organizational reality. Schemas also direct to evolution of symbolic culture in the system (cf. table 2.6). With XML and XML applications, in which RDF also belongs, the system can be described in a way that includes also the meanings of the descriptions. Thus, the fundamental problem in the language context can be at least partly solved.

In the following, we will evaluate how well the two solutions – the XML and RDF schemas, solve this problem. The following requirements are based on section 4.8, in which we presented some requirements for an appropriate schema for performance management purposes. We will evaluate both schemas and how they are able to fulfill these requirements. The evaluation is based on schemas presented in appendixes B and C. Further information on the schemas is available in the following W3Cs documents: XML Schema Part 1: Structures (1999d), XML Schema Part 2: Datatypes (1999e), XML Schema Requirements (1999f) and Resource Description Framework (RDF) Schema Specification (1999c).

5.4.1 XML Schema and Schema Requirements

In the XML schema, inheritance is supported in elements, which corresponds to classes in RDF, attributes and datatype definitions.

All elements in the XML schema can have additional tags, that is, attributes. This involves both the elements and their sub-elements. This means that additional attributes are allowed in elements. Additional attributes offer powerful extensibility mechanism. In XML schemas, extensibility is made especially possible.

XML schema specification offers the means to constrain the meaning, usage and relationships of their component parts. Component parts refer to datatypes, elements and their content, attributes and their values, entities and their contents, and notations. From this it follows that the XML schema is able to constrain elements (classes) to have only specific attributes.

The XML schema is intended to be wider than XML DTD especially in two specific ways. First, it is supposed to be more extensible than DTD is. It also offers data typing facility, which includes built-in datatypes and generated datatypes. Generated datatypes offer a possibility for each programmer to build their own datatypes for specific purposes. Datatypes facility offers thus a possibility of constraining the values of attributes to certain datatypes. This property is of significant advantage in the XML schema compared to other schemas.

The cardinalities within attributes are possible to describe in the XML schema in a way that an attribute can have either one of two “sets”. It can have 0 or 1 value from that “set”, or it can have multiple values from the multiple “set”. The ability of expressing attribute cardinalities is advantageous in situations, where a value is required, that is, an attribute can have only one value. In our example in section 4.8 we used startTime as an example, where it is required that there must be exactly one value.

Stability is still a problem in XML. As the XML schema specifications are still working drafts, some changes are expected. These changes may have an impact on future XML applications and thus on applications that are implemented today. The possible changes threaten especially the compatibility of today’s XML applications and future applications. Compatibility is also extremely important in global systems like UMTS performance management. The use of XML schema in UMTS network management and in performance management has to be seen as contingent solution. For the unstable nature of XML specifications the possible use of XML schema has to be evaluated with the consideration that XML is a new and still developing technology.

As we stated in the beginning of chapter 4, XML is a simplified subset of SGML. SGML has been criticized for its complexity. The development of XML was aimed at simpler mark-up language that would still be expressive. XML schema is written in XML and can thus be made simple. The possibility of designers to make their own schemas threatens this simplicity. We can state that extensibility has also its drawbacks, which can lead into complexity.

The XML schema is not based on any model like the RDF schema is. The uniformity cannot thus be evaluated on the basis of how all schemas are based on some model. All XML schemas are however structured the same way: they all are based on the same structures, datatypes and language. The same problem of extensibility comes into significance also in the requirement of uniformity as it did in the requirement of simplicity. Because every designer can create, for example, his own datatypes, the uniformity of all XML schemas decreases. XML schema is not even meant to be one strictly defined schema, but a schema that allows individual interpretations and implementations.

Completeness of XML schema has to be evaluated by how well the XML schema describes the Universe of Discourse. The Universe of Discourse refers to the network whose management the schema is supposed to help. XML schema does not have serious limitations that would affect on the completeness of the description of the network. Extensible nature of XML is the biggest reason why we can assume that XML is able to describe the network completely. The development of datatypes is however one factor that can affect on the extensible nature of XML. Datatypes are not totally standardized by the time of writing of this thesis and their development cannot be completely predicted.

All XML schemas are unique since W3C recommends that all versions of all schemas are named with a unique name. It is expected that there will be new network elements, which can have new measurement capabilities and new measurement types. These new properties must be added into the schema. This new schema should be named with a new name, and thus all schemas would have unique identifier.

The requirement that XML schema has to be able to describe logical relationships was covered with the requirement of inheritance. Inheritance also describes logical relations between elements.

5.4.2 RDF Schema and Schema Requirements

RDF schema has several advantages and some limitations in performance measurement data presenting. At first, it is able to present the inheritance between objects. Inheritance is supported through sub-classes and sub-properties. This produces easy extensibility and reuse of code. If network operator wishes to create a new, slightly different schema, this is possible through incremental modifications to the original schema. Sub-properties correspond to the requirement of sub-classes having their own additional attributes. Sub-properties are properties that belong only to the sub-classes, not to the inherited class.

Also relationships can be defined in the RDF schema in a very suitable way. All types of relationships can be presented through the use of properties. Relationships have to however be one-to-one, that is, only a single range can be specified. Multiple range-domain pairs are not allowed. This means that for each domain-range pair, there has to be a separate property defined. This causes unnecessary repetition in the information.

In the RDF schema, the data type attribute value has to be cannot be specified yet. Data typing is limited in the RDF schema. In the future, this property is expected to improve. Cardinality in the sense that attribute can have 0, 1 or multiple values, is not possible to specify in the RDF schema. This is a clear limitation in RDFs power of expression. If cardinalities are needed, some external functionality is needed in addition to the RDF processor.

Lyytinen's requirement of stability is a special problem in RDF schemas. As was stated in section 4.6, the RDF schema specification is still only a proposed recommendation and not a standard of any kind. RDF cannot be seen as a stable metadata application. It has not also been widely tested or even used. In a global system, like the UMTS, the requirement of stability is however important. From this follows, that the use of RDF in UMTS performance management is still somewhat risky. The possible changes in the RDF specifications can have an impact on the system and its functionality. If the properties and possibilities of RDF are however researched well, it can be feasible to network management systems even in this early phase of its development.

Simplicity in RDF schemas can be evaluated partly by readability. Since the RDF schema written in XML consists of distinct components, it is easy to interpret and simple to understand. The RDF schema does not have too many complicated features that would decrease the simplicity. RDF schema (specification) is still controllable entity.

The requirement of uniformity in RDF schemas is well fulfilled if we evaluate how uniform all schemas are by nature. All RDF schemas are based on the RDF data model and are thus uniform. The basic structure is also always the same although the syntax may differ. In our thesis, we have presented RDF schemas in XML, but RDF is intended to be possible to present in any, possible future syntax. All RDF schemas are also uniform by structure. This eases the human interpretation and readability.

Completeness can be evaluated the same way as in XML schema, that is, by how completely the schema describes the Universe of Discourse. As was stated above, RDF schema has some shortcomings concerning, for example, cardinality rules and data typing. In some cases, the target system may be described completely by RDF schemas and in some other case RDF may not have sufficient capabilities to do that. Thus, we cannot say that RDF schema fulfills totally the requirement of completeness.

The uniqueness requirement is fulfilled the same way as in XML schemas. Also RDF schemas have versioning that relates to the namespaces. All versions of schemas are supposed to be named uniquely. Lyytinen's requirement of schemas ability to describe the logical relationships was discussed above. The requirement of inheritance concerns also the logical relationships.

5.4.3 Differences Between XML and RDF Schemas

In table 5.4 we have gathered the essential advantages and disadvantages of XML and RDF schemas. These advantages and disadvantages concern especially performance management related issues and how they affect on the presentation of the data in the system. We have also evaluated their applicability to performance management.

Based on the table we can conclude that compared to RDF, XML schema has a clear advantage in its extensibility mechanism. The XML schema is easy to modify to correspond the designer's needs. RDF has an advantage in its object-oriented nature. In object-oriented systems and especially in object-oriented design the way in which RDF model is converted into a schema is advantageous.

Both schemas are still unstable and this can cause problems with their use. In XML, the extensibility mechanism can produce decreased simplicity and controllability. If used right, extensibility is however a powerful property. RDF is limited especially because it has no mechanism to handle cardinalities. This lacking causes a need for additional parts to the checking mechanism.

The XML schema has no considerable obstacles for its use in performance management. Most of all for its extensible nature it is suitable for distributive and growing environment. RDF is also suitable but not complete and cannot therefore totally cover the function it is supposed to. Some additional mechanisms are needed to express the parts of the system that RDF schema is not able to perform. RDF fits however well with object-oriented systems and therefore these little shortcomings can be considered as minor ones.

Table 5.4: Differences between XML and RDF schemas in presenting data in performance management

Feature Covered	XML Schema	RDF Schema
Advantages	Extensibility mechanisms Data typing Inheritance	Easy to construct from the RDF model Object-oriented nature fits well to object-oriented systems Inheritance
Disadvantages	Unstable Extensibility can cause decreased controllability and simplicity	Unstable No mechanism for attribute cardinalities
Applicability to performance management	Suitable because of extensibility	Limitations in attribute cardinalities requires additional processing of the data

5.5 Summary

In this chapter we evaluated all the concepts of our research. We also made an effort to analyze the research problems listed in the introduction. We showed how the development of UMTS network management system could be analyzed with the help of the three object system contexts. We pointed out the issues that have had an impact on the increased need of social activity in UMTS. The function of UMTS network management systems and the different types of metadata were analyzed with the consideration of the four social action types. XML and RDF were evaluated last with the help of requirements presented in section 4.8.

In the next chapter, we will make conclusions of the significance of this study and the results gained. We will also make suggestions for future research issues.

6 CONCLUSION

In our thesis, we studied the use of metadata in network management in UMTS networks. In network management, we concentrated on performance management's functional area. The main research problem was: *How does Habermas' critical social theory and Kalle Lyytinen's Information Systems Development framework support the selection between two different kinds of schemas in UMTS network management?* The main research problem was divided into four sub-problems. We approached the main research problem by defining a telecommunication network and network management with the help of TMN framework, showing how network management system could be analyzed with CST and showing why metadata is needed in network management. As examples of two different kinds of schemas, we introduced XML and RDF schemas.

We also approached the main research problem by acknowledging that both the amount of the interest groups and the services are increased in UMTS. That posed a need for social activity, which clearly distinguishes UMTS from second generation networks. The sub-problems were solved with the help of Habermas' four social action types and different types of metadata based on Gilliland-Swetland's division. The differences between schemas were analyzed with schema requirements.

At first, we showed that the change process in UMTS was derived from the development organization, that is, organization context. The impacts on two other contexts were that language context had to model the system defined in the organization context. The technology context had to adapt to the changes made in the organization context. We also pointed out the issues that had had an impact on the increased need of social activity in UMTS. We showed that the functionality criterion could vary among the different interest groups. If the criteria were contradictory, one of these criteria had to be selected as dominant functionality criterion. The development process had to then be carried out based on the dominant criterion. The agreement on the selected criterion demanded social activity and especially communication.

Next, we evaluated how the four social action types were taken into consideration in UMTS network management and especially in performance management. We also showed how the use of metadata was made possible in UMTS. We found that the use of metadata in performance management was most of all communicative activity. The use of metadata as communicative action also improves mutual understanding, which is the objective of all communicative actions. We found that normatively regulated actions are essential in UMTS network management systems. Norms are needed everywhere in the system. Strategic actions typically dominate in the beginning of all ISDs. In UMTS, strategic actions are restricted by the norms. Instrumental actions characterize all information systems. Traditionally, information systems are seen as efficient, technical systems whose purpose is technical control. This is also the case in UMTS performance management, although technical control is not anymore the only purpose of the system. We suggested that the use of metadata could improve the quality and efficiency of the system.

We continued by analyzing the four social action types and different types of metadata. We showed that administrative and use types of metadata related to strategic, normatively regulated and communicative action types. Descriptive metadata related to communicative and instrumental action types. Descriptive metadata was communicative because it described the system and the information in it and was thus communicative by nature. Preservation type of metadata related to communicative and strategic action types. Technical metadata was the only type of metadata that was not related to communicative action type. It related strongly to instrumental action type. We indicated that three types of metadata: administrative, descriptive and technical metadata had most significance to this study. In our thesis, we have not largely considered data storing or user accesses that refer to preservation and use types of metadata. We also found that communicative action type dominated in metadata types. As we stated before, performance management in network management systems is traditionally seen as normative and technical system. We suggested that with appropriate metadata solutions performance management could become very communicative by nature.

As we showed that the use of metadata in performance management was most of all communicative action we could also conclude that language context had a special emphasis. Communicative action type emphasized organization and language contexts and accentuated that no system management could proceed without communication. According to Lyytinen, the fundamental problem in the language context was the problem of meaning. We suggested that XML and RDF could approach this fundamental problem properly. XMLs advantage was its extensibility mechanism. It had no significant obstacles for its use in performance management. RDFs advantage was its object-oriented nature. RDF was also suitable to performance management, but was not in all senses complete. Both XML and RDF schemas are still unstable and incomplete. This may affect on their usability.

To our knowledge, UMTS performance management has not before been studied with the consideration of CST. Also the relationship between different types of metadata and four social action types has not been analyzed before. Metadata applications like XML and RDF have not been used or studied in network management systems. XML and RDF are mostly considered as Internet solutions. We believe that our thesis can help to understand that XML and RDF can be used in large systems in distributed environments. Especially systems that manage some other system and collect information from it can have an advantage through the use of XML or RDF schemas. In our thesis, CST was applied to one specific area: UMTS network management and performance management's functional area. Therefore, the results gained cannot be directly generalized to other information systems.

In our thesis, we studied the presentation of metadata. In the future, the processing of metadata should also be studied. This could mean, for instance, a further study of metadata parsers. The differences between parsers and their capabilities of processing metadata could be analyzed. Other future research is needed in the area of data mining and data warehousing. In this area, metadata could help both the storing and retrieving of the data. Data mining would also take into consideration all types of metadata. In the area of UMTS network management, also other functional areas and the possible use of metadata in them should be examined. As we stated before, configuration management could be one such area.

REFERENCES

3GPP; Technical Specification Group Services and System Aspects (1999), 3G TS 32.101 V 1.0.0: 3G Telecom Management principles and high level requirements. 3rd Generation Partnership Project, 7-11, 30-31.

Arbib M. A., Conklin E. J. & Hill J. (1987), From Schema Theory to Language. Oxford University Press, New York, 7.

Boland R. J. jr. & Day W. (1982), The Process of System Design: A Phenomenological Approach. In Proceedings of the 3rd International Conference on Information Systems (Eds. Ginzberg M., Ross C.). Ann Arbor, Michigan, 31-45.

CCITT (1992), CCITT Recommendation X.700 - Management framework definition for open systems interconnection (OSI) for CCITT applications. The International Telegraph and Telephone Consultative Committee, 3-4.

ETSI (1996), European Telecommunication Standard GSM 12.04: Digital cellular telecommunications system (Phase 2); Performance data measurements. European Telecommunications Standards Institute, 22.

ETSI (1997), Technical Report UMTS 21.06, version 1.1.1: Universal Mobile Telecommunications System (UMTS); Network and Service Management Requirements for UMTS. European Telecommunications Standards Institute, 4-15.

Gilliland-Swetland A. J. (1998), Defining Metadata. In Introduction to Metadata: Pathways to Digital Information (Ed. Baca M.). Getty Information Institute, 1-8. Available also on WWW <URL: http://www.getty.edu/gri/standard/intrometadata/>.

Goldfarb C. F. & Prescod P. (1998), The XML Handbook. Prentice Hall PTR, Upper Saddle River, New Jersey, 13-14.

Hirschheim R., Klein H. K., & Lyytinen K. (1996), Exploring the Intellectual Structures of Information Systems Development: A Social Action Theoretic Analysis. Accounting, Management and Information Technologies, Vol. 6, No. ½, 1-64.

Howe D. (1997), Free on-line Dictionary of Computing (FOLDOC) [online]. [Referred 05.09.1999]. Available on WWW <URL: http://wombat.doc.ic.ac.uk>.

ISO (1993), ISO/IEC 10646-1993 (E). Information technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane. International Organization for Standardization, Geneva.

ITU (1996), ITU-T Recommendation M.3010: Principles for a Telecommunications management network. Telecommunication Standardization Sector of ITU 2-25.

ITU (1998), ITU-T Recommendation E.410: International Network Management – General Information. Telecommunication Standardization Sector of ITU, 3.

Lyytinen K. (1986), Information Systems Development as a Social Action: Framework and Critical Implications. Dissertation, University of Jyväskylä.

Lyytinen K. (1990), Information Systems and Critical Theory – A Critical Assessment. Working Paper WP-13, Department of Computer Science, University of Jyväskylä, 10.

Lyytinen, K. & Hirschheim, R. (1989), Information Systems and Emancipation: Promise or Threat? In Systems Development for Human Progress (Eds. Klein, H. & Kumar, K.). North-Holland, Amsterdam, 115-139.

Malmi, P., Lantonen, A. (1999), Advantages of XML/EDI in logistic data interchange. Liikenneministeriö, Helsinki, 22-23.

Miller, E. (1998), An Introduction to the Resource Description Framework. D-Lib Magazine, Vol. 4, No. 5 [online]. [Referred 06.09.1999]. Available on WWW <URL: http://www.dlib.org/dlib/may98/miller/05miller.html>.

Murphy L. (1998), Digital Document Metadata in Organizations: Roles, Analytical Approaches, and Future Research Directions. In Proceedings of the Thirty-First Hawaii International Conference on System Sciences Volume II. IEEE Computer Society, 267-276.

Ngwenyama O. K. (1991), The Critical Theory Perspective on Information Systems: Problems and Challenges. In Information Systems Research: Contemporary Approaches and Emergent Traditions (Eds. Nissen H.E., Klein H. K. & Hirschheim R.). Amsterdam: North Holland, 267-280.

Nokia Networks (1999), OMeS XML Schema Specification/Draft 4. Nokia Networks internal material, 7-8, 18.

Orfali, R., Harkey, D. & Edwards, J. (1996), The Essential Distributed Objects Survival Guide. John Wiley & Sons, Inc., New York, 68-71.

Orfali, R. & Harkey, D. (1997), Client/Server Programming with Java and CORBA. John Wiley & Sons, Inc., New York, 417.

Pohja, S. (1999), Introduction to PM for NMS Software Designers. Nokia Telecommunications internal training material.

Pohja, S., Muurinen K. & Lassila O. (1999), Unified Modeling and Representation of Measurement Results [unpublished].

Pras, A., van Beijnum, B-J. & Sprenkels, R. (1999), Introduction to TMN. CTIT Technical Report 99-09, University of Twente, 10-21.

Tolvanen, J-P. (1998), Incremental Method Engineering with Modeling Tools: Theoretical Principles and Empirical Evidence. Dissertation, University of Jyväskylä, 82, 85.

W3C (1997a), W3C Data Formats, W3C Note 29-October-1997 [online]. [Referred 27.07.1999]. Available on WWW <URL: http://www.w3.org/TR/NOTE-rdfarch/>.

W3C (1997b), W3C Press release: World Wide Web Consortium Publishes Public Draft of Resource Description Framework (RDF) 03-October-1997 [online]. [Referred 22.8.1999]. Available on WWW <URL: http://www.w3.org/Press/RDF/>.

W3C (1998), Extensible Markup Language (XML) 1.0, W3C Recommendation 10-February-1998 [online]. [Referred 24.06.1999]. Available on WWW <URL: http://www.w3.org/TR/REC-xml>.

W3C (1999a), Namespaces in XML, World Wide Web Consortium 14-January-1999 [online]. [Referred 09.10.1999]. Available on WWW <URL: http://www.w3.org/ TR/1999/REC-xml-names-19990114/>.

W3C (1999b), Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999 [online]. [Referred 22.08.1999]. Available on WWW <URL: http://www.w3.org/TR/REC-rdf-syntax/>.

W3C (1999c), Resource Description Framework (RDF) Schema Specification, W3C Proposed Recommendation 03 March 1999 [online]. [Referred 24.06.1999]. Available on WWW <URL: http://www.w3.org/TR/PR-rdf-schema/>.

W3C (1999d), XML Schema Part 1: Structures, W3C Working Draft 24 September 1999 [online]. [Referred 20.10.1999]. Available on WWW <URL: http://www.w3.org/TR/xmlschema-1/>.

W3C (1999e), XML Schema Part 2: Datatypes, W3C Working Draft 24 September 1999 [online]. [Referred 20.10.1999]. Available on WWW <URL: http://www.w3.org/TR/xmlschema-2/>.

W3C (1999f), XML Schema Requirements, W3C Note 15 February 1999 [online]. [Referred 22.07.1999]. Available on WWW <URL: http://www.w3.org/TR/NOTE-xml-schema-req>.

Weiler, D. & Klas, G. (1997), GSM on the Way to UMTS. In XVI World Telecom Congress Proceedings, Toronto, 281-288.

XML Schema Developer's Guide [online]. [Referred 05.09.1999]. Available on WWW <URL: http://msdn.microsoft.com/xml/XMLGuide/schema-overview.asp>.

APPENDIX A: ATTRIBUTES AND CHARACTERISTICS OF METADATA

Attribute	Characteristics	Examples
Source of metadata	Internal metadata generated by the creating agent for an information object at the time when it is first created or digitized	– File names and header information – Directory structures – File format and compression scheme
	External metadata relating to an information object that is created later, often by someone other than the original creator	– Registrarial and cataloging records – Rights and other legal information
Method of metadata creation	Automatic metadata generated by a computer	– Keyword indexes – User transaction logs
	Manual metadata created by people	– Descriptive surrogates such as catalog records and Dublin Core metadata
Nature of metadata	Lay metadata created by persons who are neither subject nor information specialists, often the original creator of the information object	– Metatags created for a personal Web page – Personal filing systems
	Expert metadata created by either subject or information specialists, often not the original creator of the information object	– Specialized subject headings – MARC records – Archival finding aids
Status	Static metadata that never changes once it has been created	– Title, provenance, and data of creation of an information resource
	Dynamic metadata that may change with use or manipulation of an information object	– Directory structure – User transaction logs
	Long-term metadata necessary to ensure that the information object continues to be accessible and usable	– Technical format and processing information – Rights information
	Short-term metadata, mainly of a transactional nature	– Preservation management documentation
Structure	Structured metadata that conforms to a predictable standardized or unstandardized structure	– MARC – TEI and EAD – local database formats
	Unstructured metadata that does not conform to any standardized vocabulary or authority	– Unstructured note fields and annotations
Semantics	Controlled metadata that conforms to a standardized vocabulary or authority form	– AAT – ULAN – AACR2
	Uncontrolled metadata that does not conform to any standardized vocabulary or authority form	– Free-text notes – HTML metatags
Level	Collection metadata relating to collections of information objects	– Collection-level record, e.g., MARC record or finding aid – Specialized index
	Item metadata relating to individual information objects, often contained within collections	– Transcribed images captions and dates – Format information

(Gilliland-Swetland 1998, 4)

APPENDIX B: A (NORMATIVE) SCHEMA FOR SCHEMAS

```
<?xml version='1.0'?>
<!-- $Id: Overview.html,v 1.3 1999/09/24 22:52:58 connolly Exp $ -->
<!DOCTYPE schema PUBLIC '-//W3C//DTD XMLSCHEMA 19990923//EN'
     '../structures/structures.dtd'>

<schema xmlns='http://www.w3.org/1999/09/23-xmlschema/'
  targetNS='http://www.w3.org/1999/09/23-xmlschema/'version='0.6'>

<!-- The datatype element and all of its members are defined
  in XML Schema: Part 2: Datatypes -->
<include
schemaName='http://www.w3.org/1999/09/23-xmlschema/datatypes/datatypes.xsd'/>

<!-- The NCName datatype is widely used for the names of components -->
<datatype name='NCName'><basetype name='NMTOKEN'/> </datatype>

<!-- The public datatype is used for entities and notations -->
<datatype name='public'><basetype name='string'/> </datatype>

<!-- schema element -->
<element name='schema'>
  <archetype>
    <element ref='import' minOccurs='0' maxOccurs='*'/>
    <element ref='include' minOccurs='0' maxOccurs='*'/>
    <element ref='export' minOccurs='0'/>
    <group order='choice' minOccurs='0'>
      <element ref='comment'/>
      <element ref='datatype'/>
      <element ref='archetype'/>
      <element ref='element'/>
      <element ref='attrGroup'/>
      <element ref='modelGroup'/>
      <element ref='textEntity'/>
      <element ref='externalEntity'/>
      <element ref='unparsedEntity'/>
      <element ref='notation'/>
    </group>
    <attribute name='targetNS' type='uri'/>
    <attribute name='version' type='string'/>
    <attribute name='xmlns' type='uri'
            default='http://www.w3.org/1999/09/23-xmlschema/'/>
    <attribute name='model' type='NCName' default='closed'>
      <enumeration>
        <literal>open</literal>
        <literal>refinable</literal>
        <literal>closed</literal>
      </enumeration>
    </attribute>
  </archetype>
</element>

<!-- comment element -->
<element name='comment' type='string'/>

<!-- ################################################## -->
<!-- Toplevel and named: Fundamental archetypes, used hereafter to build the
    schema. A toplevel specifies an export control in addition to a name. -->

<!-- named archetype -->
<archetype name='named' model='refinable'>
```

```
  <attribute name='name' type='NCName'/>
</archetype>

<!-- toplevel archetype  -->
<archetype name='toplevel' model='refinable'>
  <refines name='named'/>
  <attribute name='export' type='boolean'/>
</archetype>

<!-- ############################################### -->
<!-- ######### Toplevel elements #################### -->
<!-- The datatype definition element is defined in
    XML Schema Part 2: Datatypes.  -->

<!-- qualifiable archetype For references which may be to components of
    other schemas. reference and typeRef are sub-types of qualifiable -->
<archetype name='qualifiable' model='refinable'>
  <attribute name='schemaName' type='uri'/>
  <attribute name='schemaAbbrev' type='NCName'/>
</archetype>

<!-- reference archetype
    refines, modelGroupRef and attrGroupRef are all kinds of reference -->
<archetype name='reference' model='refinable'>
  <refines name='qualifiable'/>
  <attribute name='name' type='NCName'/>
</archetype>

<!-- typeRef archetype -->
<!-- 'element', 'archetype' and 'attribute' are all kinds of typeRef  -->
<archetype name='typeRef' model='refinable'>
  <refines name='qualifiable'/>
  <attribute name='type' type='NCName'/>
  <attribute name='default' type='string'/>
  <attribute name='fixed' type='string'/>
</archetype>

<!-- modelGroup archetype -->
<!-- modelGroup, group and archetype are all kinds of modelGroup -->
<archetype name='modelGroup' model='refinable'>
  <attribute name='order' default='seq'>
    <enumeration>
      <literal>choice</literal>
      <literal>seq</literal>
      <literal>all</literal>
    </enumeration>
  </attribute>
</archetype>

<!-- modelElt archetype -->
<!-- the abstract class of all model elements:
    groups, elements and modelGroupRefs -->
<archetype name='modelElt' model='refinable'>
  <attribute name='minOccurs' type='non-negative-integer' default='1'/>
  <attribute name='maxOccurs' type='string'/>
  <!-- allows '*', so integer won't do -->
</archetype>

<!-- The archetype element refines the toplevel, typeRef and modelGroup
    archetypes. It may include a refines element that specifies the
    archetype(s) that is is based on, and either a datatypeQual or a model,
    followed by any number of attribute and attrGroupRef elements.  -->
<!-- archetype element -->
<element name='archetype'>
  <archetype>
    <refines name='toplevel'/>
```

```
    <refines name='typeRef'/>
    <refines name='modelGroup'/>
    <element ref='refines' minOccurs='0' maxOccurs='*'/>
  <group order='choice'>
    <element archRef='modelElt' minOccurs='0' maxOccurs='*'/>
    <element ref='datatypeQual' minOccurs='0'/>
  </group>
  <group order='choice'  minOccurs='0' maxOccurs='*'>
    <element ref='attribute'/>
    <element ref='attrGroupRef'/>
  </group>
    <attribute name='content' default='elemOnly'>
      <enumeration>
        <literal>elemOnly</literal>
        <literal>textOnly</literal>
        <literal>mixed</literal>
        <literal>empty</literal>
        <literal>any</literal>
      </enumeration>
    </attribute>
    <attribute name='model' type='NCName'>
      <!-- default comes from schema model -->
      <enumeration>
        <literal>open</literal>
        <literal>refinable</literal>
        <literal>closed</literal>
      </enumeration>
    </attribute>
  </archetype>
</element>

<!-- refines element -->
<element name='refines'>
  <archetype content='empty'>
    <refines name='reference'/>
  </archetype>
</element>

<!-- The element element refines the toplevel and typeRef archetype.
     It can be used either at the toplevel to define an element-type binding
     globally,
     or within a content model to either reference a globally-defined
     element or archetype or declare an element-type binding locally.
     The ref/archRef forms are not allowed at the top level -->
<!-- element element -->
<element name='element'>
  <archetype order='choice'>
  <refines name='toplevel'/>
  <refines name='typeRef'/>
  <refines name='modelElt'/>
    <element ref='datatype'/>
    <element ref='archetype'/>
  </archetype>
</element>

<!-- The group element refines the modelElt and modelGroup archetypes. -->
<!-- group element -->
<element name='group'>
  <archetype>
    <refines name='modelElt'/>
    <refines name='modelGroup'/>
    <element archRef='modelElt' minOccurs='2' maxOccurs='*'/>
  </archetype>
</element>

<!-- The modelGroup element refines the toplevel & modelGroup archetypes. -->
```

```
<!-- modelGroup element -->
<element name='modelGroup'>
  <archetype>
    <refines name='toplevel'/>
    <refines name='modelGroup'/>
    <element archRef='modelElt' minOccurs='1' maxOccurs='*'/>
  </archetype>
</element>

<!-- The modelGroupRef element refines the reference & modelElt archetypes -->
<!-- modelGroupRef element -->
<element name='modelGroupRef'>
  <archetype content='empty'>
    <refines name='reference'/>
    <refines name='modelElt'/>
  </archetype>
</element>

<!-- The datatypeQual archetype provides for modifying datatypes referenced
    from attribute declarations and archetype definitions. It is realised by
    the datatypeQual element and refined by the attribute element -->
<archetype name='datatypeQual' order='all'>
  <element archRef='maxBound' minOccurs='0'/>
  <element archRef='minBound' minOccurs='0'/>
  <element ref='maxInclusive' minOccurs='0'/>
  <element ref='minInclusive' minOccurs='0'/>
  <element ref='precision' minOccurs='0'/>
  <element ref='scale' minOccurs='0'/>
  <element ref='length' minOccurs='0'/>
  <element ref='maxLength' minOccurs='0'/>
  <element ref='enumeration' minOccurs='0'/>
  <element ref='lexicalRepresentation' minOccurs='0'/>
</archetype>

<!-- The datatypeQual element realises the datatypeQual archetype -->
<element name='datatypeQual' type='datatypeQual'/>

<!-- the attribute element declares attributes -->
<element name='attribute'>
  <archetype>
    <refines name='datatypeQual'/>
    <refines name='typeRef'/>
    <refines name='named'/>
    <attribute name='minOccurs' type='non-negative-integer' default='0'>
      <enumeration>
        <literal>0</literal>
        <literal>1</literal>
      </enumeration>
    </attribute>
    <attribute name='maxOccurs' type='integer' fixed='1'/>
  </archetype>
</element>

<!-- attrGroup element -->
<element name='attrGroup'>
  <archetype>
  <refines name='toplevel'/>
    <group order='choice' minOccurs='1' maxOccurs='*'>
      <element ref='attribute'/>
      <element ref='attrGroupRef'/>
    </group>
  </archetype>
</element>

<!-- The attrGroupRef element refines the reference archetype.  -->
<!-- attrGroupRef element -->
```

```
<element name='attrGroupRef'>
  <archetype>
    <refines name='reference'/>
  </archetype>
</element>

<!-- The textEntity element refines the toplevel archetype.
     It provides for string content to specify the entity value.  -->
<!-- textEntity element -->
<element name='textEntity'>
  <archetype type='string'>
    <refines name='toplevel'/>
  </archetype>
</element>

<!-- The externalRef archetype provides for specification
     of a uri, an optional public identifier, and a notation attribute.
     It refines the toplevel archetype -->
<archetype name='externalRef' model='refinable' content='empty'>
  <refines name='toplevel'/>
  <attribute name='system' type='uri' minOccurs='1'/>
  <attribute name='public' type='public'/>
</archetype>

<!-- the typedExternalRef adds a required notation to an external ref -->
<archetype name='typedExternalRef' model='refinable'>
  <refines name='externalRef'/>
  <attribute name='notation' type='NOTATION' minOccurs='1'/>
</archetype>

<!-- The externalEntity and unparsedEntity elements are
     based on the typedExternalRef archetype. -->
<!-- externalEntity element -->
<element name='externalEntity'>
  <archetype>
    <refines name='typedExternalRef'/>
    <attribute name='notation' fixed='XML'/>
  </archetype>
</element>

<!-- unparsedEntity element -->
<element name='unparsedEntity'>
  <archetype>
    <refines name='typedExternalRef'/>
  </archetype>
</element>

<!-- The notation element refines the externalRef archetype. -->
<element name='notation'>
  <archetype>
    <refines name='externalRef'/>
  </archetype>
</element>

<!-- ############################################### -->
<!-- import, export and include -->
<!-- The import, export and include elements all refine the restrictions
     archetype, whose attributes can be used to enable or disable import and
     export restrictions.Within import and include elements, references to the
     components of foreign schemas control their importation or inclusion,
     respectively.  -->
<!-- The import and include elements both refine external -->

<archetype name='restrictions' model='refinable'>
  <attribute name='datatypes' type='boolean' default='true'/>
  <attribute name='archetypes' type='boolean' default='true'/>
```

```
  <attribute name='elements' type='boolean' default='true'/>
  <attribute name='attrGroups' type='boolean' default='true'/>
  <attribute name='modelGroups' type='boolean' default='true'/>
  <attribute name='entities' type='boolean' default='true'/>
  <attribute name='notations' type='boolean' default='true'/>
</archetype>

<archetype name='external' order='choice' model='refinable'>
  <refines name='restrictions'/>
  <element ref='component' minOccurs='0' maxOccurs='*'/>
  <attribute name='schemaName' minOccurs='1' type='uri'/>
</archetype>

<!-- component element, used in external -->
<element name='component'>
  <archetype content='empty'>
    <attribute name='name' type='NCName' minOccurs='1'/>
    <attribute name='type' minOccurs='1'>
      <enumeration>
        <literal>datatype</literal>
        <literal>archetype</literal>
        <literal>element</literal>
        <literal>attrGroup</literal>
        <literal>modelGroup</literal>
        <literal>entity</literal>
        <literal>notation</literal>
      </enumeration>
    </attribute>
  </archetype>
</element>

<!-- import element  -->
<element name='import'>
  <archetype>
    <refines name='external'/>
    <attribute name='schemaAbbrev' minOccurs='1' type='NCName'/>
  </archetype>
</element>

<!-- export element  -->
<element name='export'>
  <archetype content='empty'>
  <refines name='restrictions'/>
  </archetype>
</element>

<!-- include element  -->
<element name='include' type='external'/>

<!-- ################################################ -->
<!-- notations for use within XML Schema schemas      -->
<notation name='XMLSchemaStructures'
  public='structures'
  system='http://www.w3.org/1999/09/23-xmlschema/structures/structures.xsd'/>
<notation name='XML'
  public='REC-xml-19980210'
  system='http://www.w3.org/TR/1998/REC-xml-19980210'/>
</schema>
```

(based on W3C 1999d, 50-57)

APPENDIX C: RDF SPECIFICATION IN SERIALIZATION SYNTAX

Note that the namespace URIs listed here are only examples because formal identifiers have not yet been defined. The labels in French are left out from the original version.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/TR/1999/PR-rdf-schema-19990303#">

<rdf:Description ID="Resource">
  <rdf:type resource="#Class"/>
  <rdfs:label xml:lang="en">Resource</rdfs:label>
  <rdfs:comment>The most general class>/rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">type</rdfs:label>
  <rdfs:comment>Indicates membership of a class</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
</rdf:Description>

<rdf:Description ID="comment">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">comment</rdfs:label>
  <rdfs:domain rdf resource="#Resource"/>
  <rdfs:comment>Use this for descriptions</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                                   19990303#Literal"/>
</rdf:Description>

<rdf:Description ID="label">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">label</rdfs:label>
  <rdfs:domain rdf resource="#Resource"/>
  <rdfs:comment>Provides human-readable version of the resource
                  name.</rdfs:comment>
  <rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                            19990303#Literal"/>
</rdf:Description>

<rdf:Description ID="Class">
  <rdf:type resource="#Class"/>
  <rdfs:label xml:lang="en">Class</rdfs:label>
  <rdfs:comment>The concept of Class</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#Resource"/>
</rdf:Description>

<rdf:Description ID="subClassOf">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">subClassOf</rdfs:label>
  <rdfs:comment>Indicates membership of a class</rdfs:comment>
  <rdfs:range rdf resource="#Class"/>
  <rdfs:domain rdf resource="#Class"/>
</rdf:Description>

<rdf:Description ID="subPropertyOf">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
```

```
  <rdfs:label xml:lang="en">subPropertyOf</rdfs:label>
  <rdfs:comment>Indicates specialization of properties</rdfs:comment>
  <rdfs:range rdf:resource=" http://www.w3.org/1999/02/22-rdf-syntax-
                                    ns#Property"
  <rdfs:domain rdf:resource=" http://www.w3.org/1999/02/22-rdf-syntax-
                                    ns#Property"
</rdf:Description>

<rdf:Description ID="seeAlso">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">seeAlso</rdfs:label>
  <rdfs:comment>Indicates a resource that provides information about the
                  subject</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                                  19990303#Resource"/>
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                                  19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="isDefinedBy">
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:label xml:lang="en">isDefinedBy</rdfs:label>
  <rdfs:comment>Indicates a resource containing and defining the subject
                  resource</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                                  19990303#Resource"/>
  <rdfs:domain rdf:resource="http://www.w3.org/TR/1999/PR-rdf-schema-
                                  19990303#Resource"/>
</rdf:Description>

<rdf:Description ID="ConstraintResource">
  <rdfs:label xml:lang="en">ConstraintResource</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>Resources used to express RDF Schema
                  constraints.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="ConstraintProperty">
  <rdfs:label xml:lang="en">ConstraintProperty</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:subClassOf rdf:resource="#ConstraintResource"/>
  <rdfs:comment>Properties used to express RDF Schema
                  properties.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="domain">
  <rdf:type resource="#ConstraintProperty"/>
  <rdfs:label xml:lang="en">domain</rdfs:label>
  <rdfs:comment>This is how we associate a class with properties
                  that its instances can have.</rdfs:comment>
</rdf:Description>

<rdf:Description ID="range">
  <rdf:type resource="#ConstraintProperty"/>
  <rdfs:label xml:lang="en">range</rdfs:label>
  <rdfs:comment>Properties that can be used in a schema to provide
                  constraints.</rdfs:comment>
  <rdfs:range rdf:resource="#Class"/>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

<rdf:Description ID="Property">
  <rdfs:label xml:lang="en">Property</rdfs:label>
  <rdfs:comment>The concept of a property.</rdfs:comment>
```

```
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdf:type resource="#Class"/>
</rdf:Description>

<rdf:Description ID="Literal">
  <rdfs:label xml:lang="en">Literal</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:comment>This represents the set of atomic values, eg. textual
                 strings.</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement">
  <rdfs:label xml:lang="en">Statement</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>This represents the set of reified
                 statements.</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#subject">
  <rdfs:label xml:lang="en">subject</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
                              ns#Statement"/>
  <rdfs:range rdf:resource="#Resource"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate">
  <rdfs:label xml:lang="en">predicate</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
                              ns#Statement"/>
  <rdfs:range rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
                              ns#Property"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#object">
  <rdfs:label xml:lang="en">object</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
  <rdfs:domain rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
                              ns#Statement"/>
</rdf:Description>

<rdf:Description ID="Container">
  <rdf:type resource=""/>
  <rdfs:label xml:lang="en">Container</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Resource"/>
  <rdfs:comment>This represents the set Containers.</rdfs:comment>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
  <rdfs:label xml:lang="en">Bag</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Sequence">
  <rdfs:label xml:lang="en">Sequence</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Alt">
  <rdfs:label xml:lang="en">Alt</rdfs:label>
  <rdf:type resource="#Class"/>
```

```
  <rdfs:subClassOf rdf:resource="#Container"/>
</rdf:Description>

<rdf:Description ID="ContainerMembershipProperty">
  <rdfs:label xml:lang="en">ContainerMembershipProperty</rdfs:label>
  <rdf:type resource="#Class"/>
  <rdfs:subClassOf rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-
                                              ns#Property"/>
</rdf:Description>

<rdf:Description about="http://www.w3.org/1999/02/22-rdf-syntax-ns#value">
  <rdfs:label xml:lang="en">object</rdfs:label>
  <rdf:type resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Description>

</rdf:RDF>
```

(W3C 1999c, 24-27)

APPENDIX D: EXAMPLES OF NETWORK ELEMENT DEFINITION

In the following is presented the BarFNE class in both XML and RDF/XML for the reason that it is presented in figure 4.7 but is not defined by the OMeS specification. In the following are the definitions of the BarFNE class first in XML and then in RDF/XML.

Class BarFNE defined in XML:

```
<element name='tchFails' required='false' type='indicator'/>

<element name='callSuccessfull' required='false' type='indicator'/>
<!-- Base class for BarFNE -->
<archetype name='BarFNE' model='refinable' order='all'>
  <refines name='FooPMTarget'/>
  <!-- Number of traffic channel fails -->
  <element ref='tchFails' minOccur='0' maxOccur='1'/>
  <!-- Number of succesfull calls -->
  <element ref='callSuccessfull' minOccur='0' maxOccur='1'/>
</archetype>

<element name='BarFNE'>
  <archetype>
    <refines name='BarFNE'/>
  <archetype/>
</element>
```

(based on Nokia Networks 1999, 8)

Class BarFNE defined in RDF/XML:

```
<Class rdf :ID="BarFNE"
    comment="Base class for BarFNE"/>
    <subClassOf rdf:resource="&fo;FooPMTarget"/>
  </Class>

  <Property rdf:ID="tchFails"
    comment="Number of traffic channel fails">
    pri:minCardinality="0"
    pri:maxCardinality="1"
    pri:minimum="0"
    <domain rdf:resource=&fo;FooPMSetup"/>
    <range rdf:resource="&pr;Integer"/>
  </Property>

  <Property rdf:ID="callSuccessfull"
    comment="Number of successfull calls">
    pri:minCardinality="0"
    pri:maxCardinality="1"
    pri:minimum="0"
    <domain rdf:resource=&fo;FooPMSetup"/>
    <range rdf:resource="&pr;Integer"/>
  </Property>

<Class>
```

APPENDIX E: EXAMPLES OF MEASUREMENT FILE

Example of OMeS measurement file in XML:

```
<?xml version='1.0'?>

<FooOMeS xmlns='http://www.nokia.com/xmlschema/MF-FNE1.xsd'>

  <FooPMSetup  MediatorID='PLMN-operator/FNE_123'
               startTime='19990211T190000+0200'
               duration='60'
               reliability='0'
               stopTime='19990211T200155+0200'>
    <BarFNE MOID='/DN:FUN-23'>
      <tchFails>16</tchFails>
      <callSuccessfull>314</callSuccessfull>
    </BarFNE>
  </FooPMSetup>


  <FooPMSetup  MediatorID='PLMN-operator/FNE_123'
               startTime='19990311T110000+0200'
               duration='60'
               reliability='0'
               realStopTime='19990311T120148+0200'>
    <BarFNE MOID='/DN:FUN-46'>
      <tchFails>12</tchFails>
      <callSuccessfull>248</callSuccessfull>
    </BarFNE>
  </FooPMSetup>

</FooOMeS>
```

(based on Nokia Networks 1999, 18)

Example of OMeS measurement file in RDF/XML:

```
<?xml version='1.0'?>

<rdf :RDF  xmlns :rdf= http://www.w3.org/1999/02/22-rdf-syntax-ns#
           xmlns=http://www.w3.org/TR/PR-rdf-schema-19990303#
           xmlns:ome=http://www.nokia.com/RDF/OmeDM.rdf#
           xmlns:foo=http://www.nokia.com/RDF/foo-OmeDM.rdf#
           xmlns:fne=http://www.nokia.com/RDF/NFNE.rdf#>


    <fne:BarFNE ome:MOID="/DN:FUN-23"
                fne:tchFails="16"
                fne:callSuccesfull="314">
           <ome:hasSetup>
               <foo:FooPMSetup
                   foo:MediatorID="PLMN-operator/FNE_123"
                   foo:startTime="19990211T190000+0200"
                   foo:duration="60"
                   foo:reliability="0"
                   foo:stopTime="19990211T200155+0200"/>
           </ome:hasSetup>
    </fne:BarFNE>

    <fne:BarFNE ome:MOID="/DN:FUN-46"
                fne:tchFails="12"
```

```
                fne:callSuccesfull="248">
            <ome:hasSetup>
                <foo:FooPMSetup
                    foo:MediatorID="PLMN-operator/FNE_123"
                    foo:startTime="19990311T110000+0200"
                    foo:duration="60"
                    foo:reliability="0"
                    foo:stopTime="19990311T120148+0200"/>
            </ome:hasSetup>
        </fne:BarFNE>

    </foo:FooPMTarget>

</rdf:RDF>
```

From these examples, we see that the network element was foo network element FNE that has mediator id "PLMN-operator/FNE_123". Management object id was in the former example "/DN:FUN-23" and in the latter "/DN:FUN-46".

The time in the first measuring interval was November 2, 1999 at 19.02 to 20.01 (with a duration of 60 minutes). The result values were: 16 traffic channel fails (i.e. failed calls) and 314 successfull calls. The time in the latter measuring interval was November 3, 1999 at 11:02 to 12:01, with the same 60 minutes duration. The result values were: 12 traffic channel fails and 248 successfull calls. Reliability value, i.e. the reliability of the measurement data was in both cases true.