Transforming relational databases into XML documents. Case study

MANUELLA KADAR, IOAN ILEANĂ

1 Decembrie 1918, University of Alba Iulia, Romania

ABSTRACT

This paper presents a particular case of transformation applied to a relational database created with Visual FoxPro 6.0 and connected to a MySQL server. We have used XML-DBMS middleware written by Ronald Bourret for transferring data between XML documents and relational databases. It maps the XML document to the database according to an object-relational mapping in which element types are generally viewed as classes and attributes and PCDATA as properties of those classes. An XML-based mapping language allows the user to specify customize this mapping. XML-DBMS is available as a set of Java packages. XML-DBMS, along with its source code, is freely available for use in both commercial and non-commercial settings. It is not copyrighted. System requirements for the achievement of such transformation are: XML-DBMS, JDK (Java Development Kit) 1.1.x or 1.2.x, MySQL relational database, JDBC driver for the database, an XML parser written in Java, a DOM level 1 implementation written in Java, SAX (Simple API for XML) version 1.

Here we present some sollutions applied to an archaeometallurgical database that collects artefacts discovered on the teritorry of Romania and have full archaeological and analytical description. Features of the database are: detailed, accurate, up-to-date documentation of the metallic objects, easy access, processing and filtering of data that facilitate the archaeological research in the interpretation of human cultures and development of technologies. The material is stored in an objective way, enabling all the interested researchers to get acquainted with it and to get different interpretations. The information stored is useful to scholars aiming to study the production and exchange systems in Eneolithic and Bronze Age Europe, to identify geological sources for raw materials available in Eneolithic and Bronze Age, to achieve typological classifications of artefacts correlated with classifications based on chemical composition and microstructure analyses, to understand specific features of metalworking techniques in this region and to compare them with other early metallurgical centres from prehistoric Europe. Our efforts are focused on the online publishing of data and to make it available to the scholars worldwide interested in the field of prehistoric metal production. Some issues we have faced: wide range of data which needs integrating, otherwise standards for current archaeological work such as GIS information, photographic information, text, and databases, archaeological complexity expressed by multiple sites, multiple kinds of artefacts, multiple terminology.

1. GENERAL METHODOLOGY

In this paper we discuss aspects as concerns the transformation into XML of a relational database created as research tool in an archaeometallurgical project. The repository is consisted of artifacts and metallurgical by-products belonging to the Eneolithic and Bronze Age within the Carpathian Arch. Some of the archaeological questions to be answered by searching the entered data are reconstruction of the production and exchange systems in the Eneolithic and Bronze Age Europe, identification of the geological sources for raw materials available in Eneolithic and Bronze Age, typological classifications of artifacts correlated with classifications based on chemical composition and microstructure analyses, specific features of metalworking techniques in this region and comparison with other early metallurgical centers from Central and South-Eastern Europe.

Interdisciplinary methodological approach has been adopted with the application of the latest analytical techniques by integrating various methods of assessment and means of dissemination. Artifacts have been considered in their archaeological and social context, so that they could be used for the social and spatial characterization of the technological systems which have given these objects.

2. FEATURES OF THE RELATIONAL DATABASE CREATED IN VISUALFOXPRO

A common and powerful method for organizing data for computerization is the relational data model. Relational databases have a very well-known and proven underlying mathematical theory, a simple one (the set theory) that makes possible automatic query optimization, schema generation from high-level models and many other features that are now vital for mission-critical Information Systems development and operations. Key areas include optimization of the index configuration, data placement, and storage allocation (Kadar e.a. 2003).

Most applications in archaeology use traditional Entity-Relation modeling and physical database design to create and maintain operational databases. These databases are typically very specialized and are designed to support very specific application requirements. Recent trend is to built data warehouses. These new mega-databases enable end users to access information based on data that was previously unavailable to them in a single place. For the data professional, the newest challenge is to design an optimized relational database that satisfies a much different set of requirements (Kadar 2002). Some requirements fulfilled in this database are related to the detailed, accurate, up-to-date documentation of the metallic objects, easy access, processing and filtering of data that facilitate the archaeological research in the interpretation of human cultures and development of technologies. The material is stored in an objective way, enabling all the interested researchers to get acquainted with it and to get different interpretations.

Interpretative observations resulted from the subjective view of the individual researcher such as cultural, functional or similar attributions have been entered into the fields labeled "Notes" or "Observations" in order to provide maximum level of objectivity.

2.1. DATABASE STRUCTURE

The database was structured as follows: – a master General Data Form (Figure 1), with child forms such as Archaeological Data Form (Figure 2), Physical Description Form (Figure 3), Microstructure Data Form (Figure 4), Chemical Composition Form (Figure 5), Bibliographical Data Form (Figure 6).

General Data Form comprises the scanning and measurable data with special entities for personal observations and interpretations.

Archaeological Data Form was designed as a sub-form that contains the site description, location of the find, the circumstances of the discovery, dating, cultural attribution and fields for observations and bibliography.

Physical Description Form is a sub-form that contains functional attributes, technological and ornamental description, various treatments to which the object might have been submitted, after its recovery, evaluations to what extent those may have affected its present state of preservation, state of preservation and the signs of wear.

Microstructure Data Form is a sub-form for the identification of artifacts` microstructures.

Comparisons with previous analyses obtained from the work of other research teams can be achieved.

Chemical Composition Form is a sub-form that contains different methods of determination applied to each object and results. Comparisons between several methods applied during the last centuries are available and assessments of values can be obtained.

Bibliography Form is a connection to a collection of references and bibliography has been designed to offer existing information on the specific issues.

2.2. FURTHER DEVELOPMENT OF THE VISUALFOXPRO DATABASE

As further improvement of the database we thought of including a three-dimensional reconstruction of the site and the features inside it by using entered data on elevations. Inter-link with other independent databases and archives in a graphic environment that permits access to various levels of research and increases the possibilities for consultation and management of the cultural heritage is also intended to be achieved.

3. TRANSFORMING THE RELATIONAL DATABASE INTO XML DOCUMENTS

When one deals with transformation of data from archaeological databases into XML documents faces issues such as:

- The wide range of data which needs integrating, that are standard for curent archaeological work, such as GIS information, photographic information, text, and databases;.
- Archaeological complexity:
- Multiple sites
- Multiple kinds of artifacts
- Multiple terminology

XML-DBMS is a middleware written by Ronald Bourret (http://www.rpbourret/xml/index.htm) for transferring data between XML documents and relational databases. It maps the XML document to the database according to an object-relational mapping in which element types are generally viewed as classes and attributes and PCDATA as properties of those classes (Bourret, 2004). An XML-based mapping language allows the user to customize such mapping. XML-DBMS is available both as a set of Java packages and as a PERL module [courtesy of Nick Semenov (nsemenov@yahoo.com)]. XML-DBMS, along with its source code, is freely available for use in both commercial and non-commercial settings. It is not copyrighted.

3.1. SYSTEM REQUIREMENTS FOR THE ACHIEVEMENT OF THIS APPLICATION THERE HAVE BEEN USED:

XML-DBMS
JDK (JavaDevelopment Kit) 1.1.x or 1.2.x
Relational database: VisualFoxPro connected to MySQL
A JDBC driver for the database
An XML parser written in Java
A DOM level 1 implementation written in Java
SAX (Simple API for XML) version 1

It is necessary to map the XML document schema (DTD, XML schema) to the database schema, then the data transfer software is built on this mapping. The software may use and XML query language (Xpath, XQuery) or simply transfer data according to the mapping (the XML equivalent of SELECT*FROM Table)

3.2. MAPPING DOCUMENT SCHEMAS TO DATABASE SCHEMAS

Two mappings are commonly used to map an XML document schema to the database schema:
table-based mapping and object-relational mapping.

3.2.1. TABLE-BASED MAPPING

The table-based mapping is used by the middleware product that transfers data between the XML document and the relational database. It models XML documents as a single table or set of tables. That is, the structure of an XML document must be as follows, where the <database> element and additional <table> elements do not exist in the single-table case:

```
<database>
        <table>
            <row>
                <column1>...</column1>
            <column2>...</column2>
            ...
      </row>
          <row>
            ...
      </row>
        ...
  </table>
        <table>
  ...
        </table>
...
</database>
```

The structure of the document must exactly match the structure expected by mapping, in such case it is used XSLT. Before transferring data to the database the document is first transformed to the structure expected by the mapping, the data is then transferred. Similarly after transferring data from the database, the resulting document is transformed to the structure needed by the application.
XML-DBMS views an XML document as a tree of objects and then uses an object-relational mapping to map these objects to a relational database.
In this view, element types generally correspond to classes and attributes and PCDATA correspond to properties. Child element types are generally viewed as pointed-to classes; that is, an interclass relationship exists between the classes corresponding to parent and child element types (Bourret 2004).

3.2.2. OBJECT-RELATIONAL MAPPING

In this model, element types with attributes, element content, or mixed content (*complex element types*) are generally modeled as classes. Element types with PCDATA-only content (*simple element types*), attributes, and PCDATA are modeled as scalar properties.
The model is then mapped to relational databases using traditional object-relational mapping techniques or SQL object views. Classes are mapped to tables, scalar properties are mapped to columns, and object-valued properties are mapped to primary key / foreign key pairs (Bourret 2004).
We wrote here an example of xml file describing the general data record of a hoard.

```
<Artifacts>
   <HoardRegistration ArNumber=”12”>
     <Museum MuseumNumber=”10”>
       <MuseumName>Bruckenthal Museum</MuseumName>
       <Street>1 Piata Sfatului</Street>
       <City>Sibiu</City>
```

```
      <County>Sibiu</State>
      <PostCode>510009</PostCode>
    </Museum>
    <RegistrationDate>040945</RegistrationDate>
    <Line LineNumber="1">
      <Part PartNumber="1">
        <Description>
          <P><B>Axe adze:</B><BR />
          Cooper</P>
        </Description>
        <Epoque>Eneolithic</Epoque>
      </Part>
      <Culture>Bodrogkeresztur</Culture>
    </Line>
    <Line LineNumber="2">
      <Part PartNumber="2">
        <Description>
          <P><B>Arm band<B><BR />
         bronze</P>
        </Description>
        <Epoque>Early Bronze Age</Epoque>
      </Part>
      <Culture>Baden</Culture>
    </Line>
  </HoardRegistration>
</Artifacts>
```

This could be viewed as a tree of objects from five classes: Artifacts, HoardRegistration, Museum, Line, and Part, as shown in the following diagram:

```
          Artifacts
              |
      HoardRegistration
          /   |   \
     Museum Line Line
              |    |
            Part  Part
```

Tables for the Artifacts Sample

The following tables are needed to store the artifacts.xml document, as it is mapped with artifacts.map

	HoardsRegistration			Lines	
	Number	VARCHAR(10)		LineNumber	VARCHAR(10)
	ArNumber	VARCHAR(10)		Number	INTEGER
	RegistrationDate	DATE		Part	VARCHAR(10)
	Epoque	VARCHAR(20)		Culture	VARCHAR(10)

	Museums			Parts	
	Number	VARCHAR(10)		Number	VARCHAR(10)
	Name	VARCHAR(40)		Description	VARCHAR(255)
	Street/td>	VARCHAR(50)			
	City	VARCHAR(50)			
	County	VARCHAR(2)			
	PostalCode	VARCHAR(10)			

4. CONCLUSIONS

One advantage of XML is the potential for very varied output with considerable easiness. We would like to have the catalogue hosted online and to make it available in digital form through other means, as well. XML possess greater flexibility than 'traditional' database packages (Philips, 2001) therefore we have decided to apply object relational mapping and to transfer data from the relational database into the xml documents by using the middle ware product mentioned before.

REFERENCES

BAZIAN, M., *et al.* (2001) – *Totul despre VisualFoxPro™ 6,* Ed. Teora, Bucureşti.

BOURRET, R. (2004) – XML *and Databases*, at http://www.rpbourret/xml/index.htm.

KADAR, M. (2002) – *Data modeling and relational database design in archaeology*, în *ActaUA*, 3, p. 73-80.

KADAR, M.; BUCUR,V.; CEUCA, E. (2003) – *Relational Database Management System for the Early Metallurgy of Copper and Bronze in Transylvania,* in *Proceedings of the 30th Conference of Computer Applications in Archaeology CAA2002*, (ed.) Martin Doerr and Apostolos Sarris, Atena, p. 335-339.

PHILIPS, L. A. (2001) – *XML,* Teora ed., Bucharest.

FIGURES

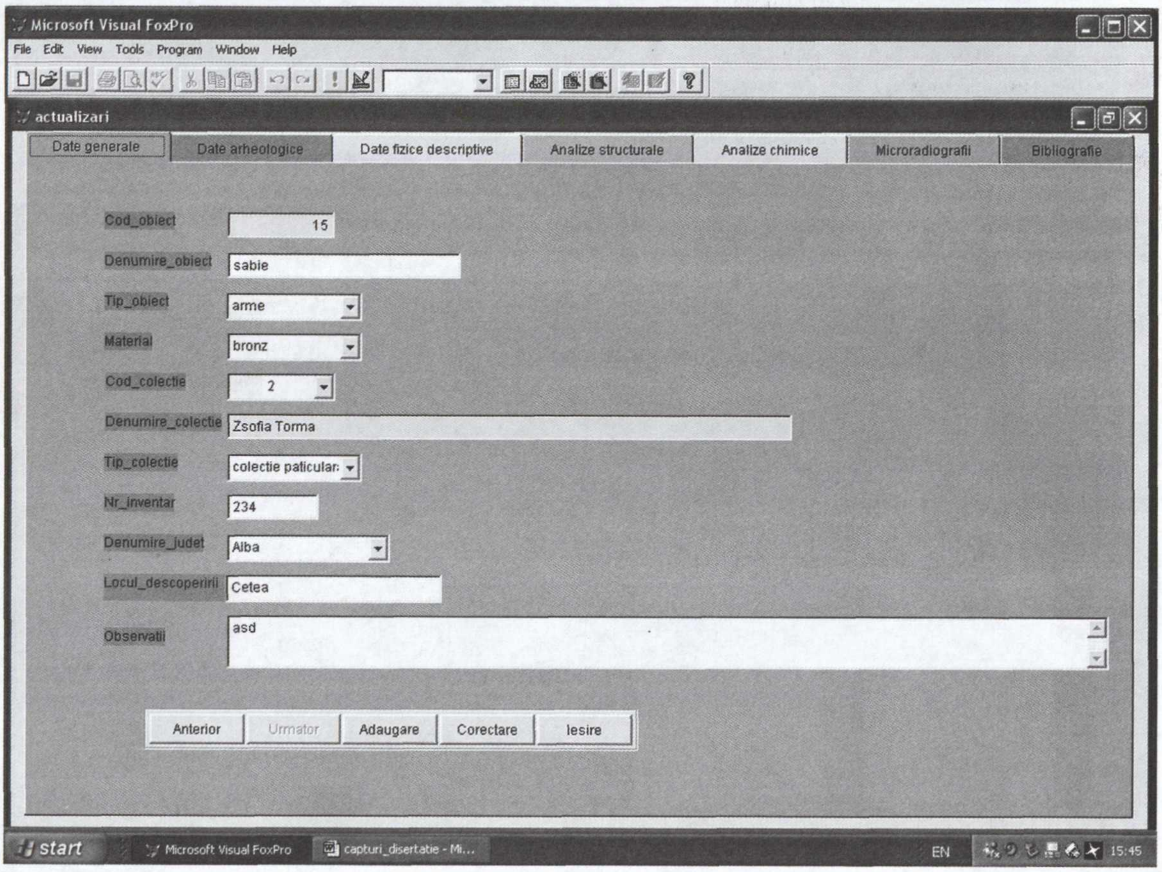

Fig. 1 – General Data Form.

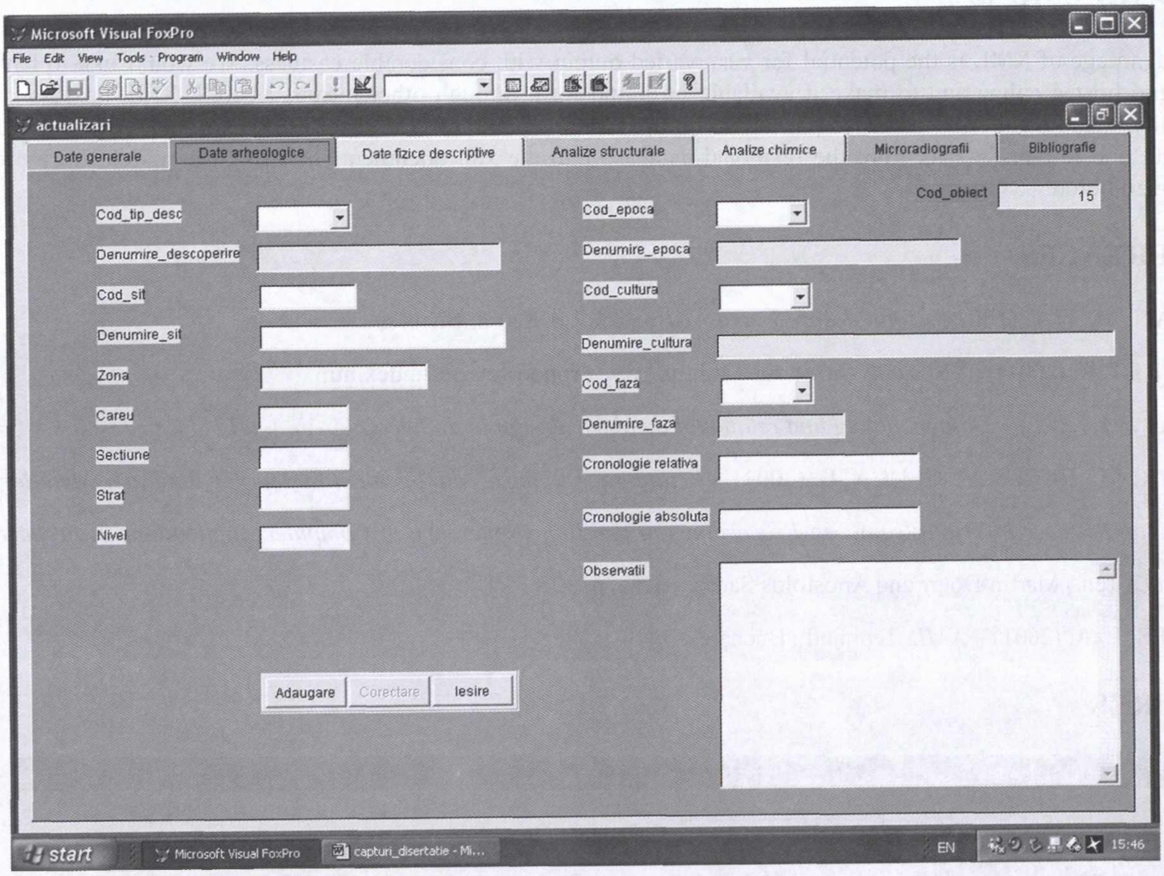

Fig. 2 – Archaeological Data Form.

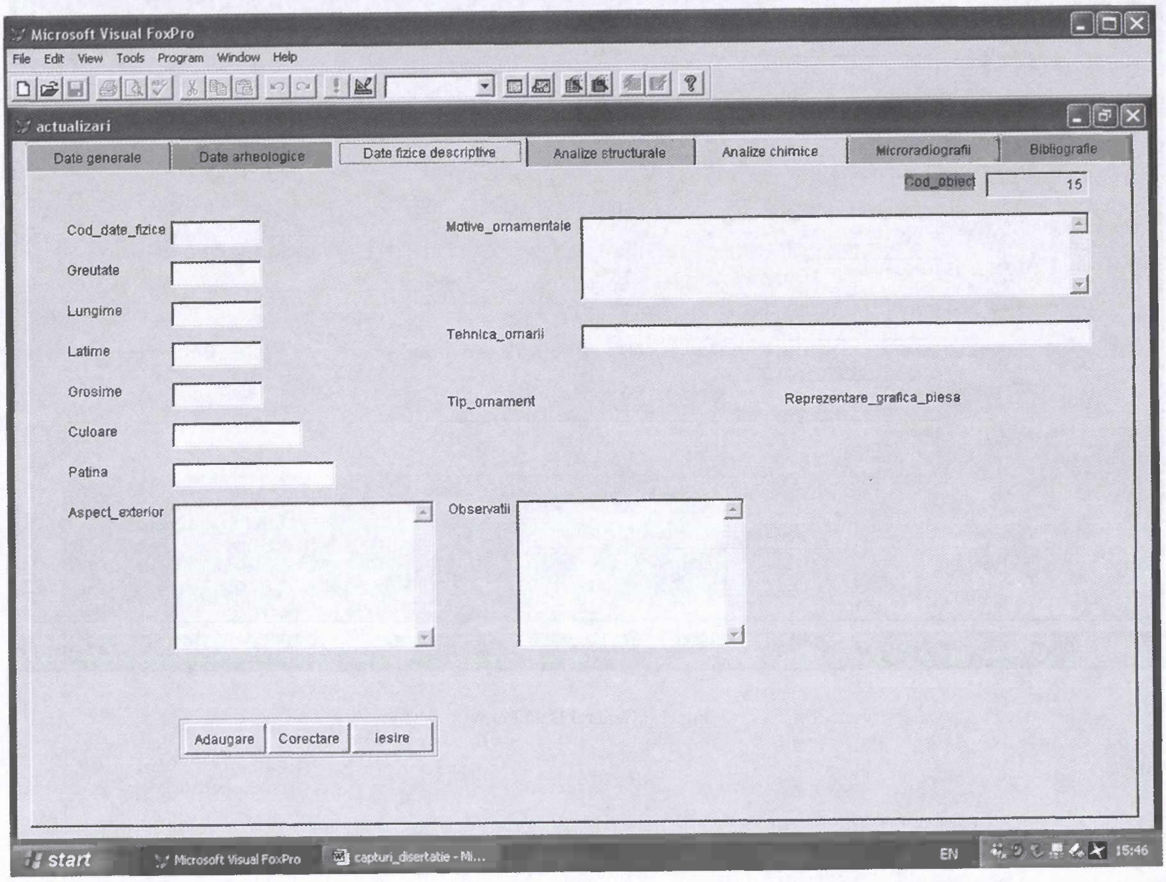

Fig. 3 – Physical Description Form.

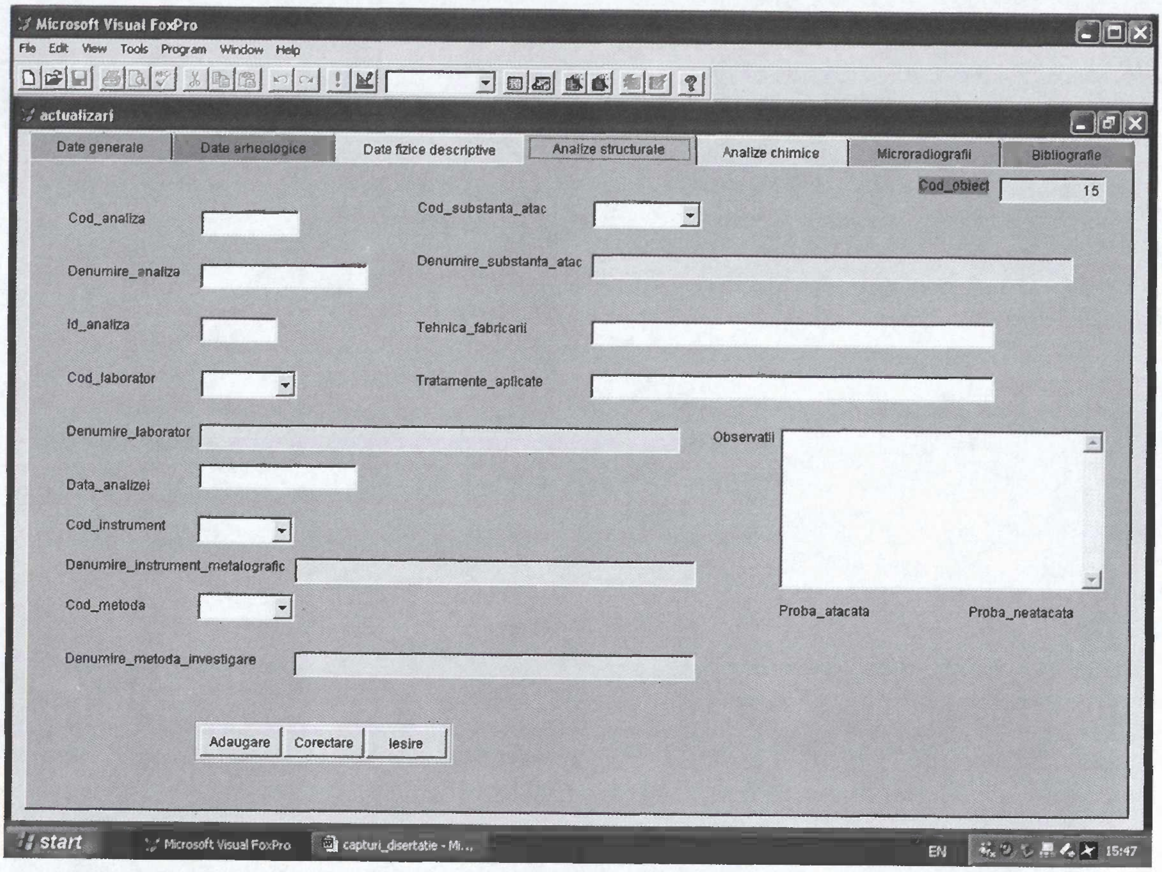

Fig. 4 – Microstructure Data Form.

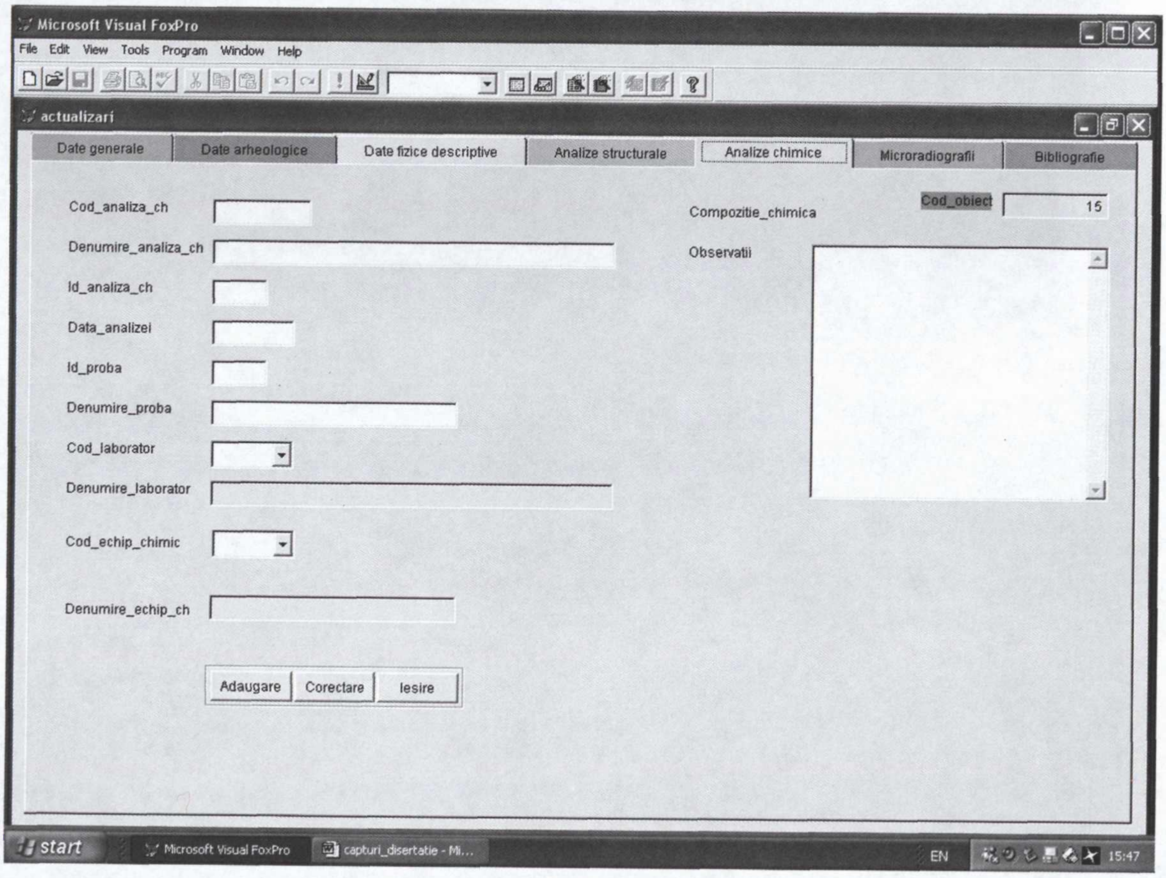

Fig. 5 – Chemical Composition Form.