TSML: A XML-based Format for Exchange of Training Samples for Pattern Recognition in Remote Sensing Images

Marinalva Dias Soares [1]
Luciano Vieira Dutra [1]
Nandamudi L. Vijaykumar [1]

Abstract: The diversity of remote sensing and geospatial data formats makes exchange of data a difficult task, in particular, for applications in knowledge discovery, pattern recognition, analysis and integration of data. One of the activities applied in remote sensing data is the classification (supervised or non-supervised). In the supervised classification, focus of this paper, such as the choice of classifier, the training samples have a major influence in the classification's results. Thus, the extraction of samples and the association of the same to a class is an important task. Moreover, these samples could be shared among classifiers from different applications for analysis and comparisons of results. In this context, this paper focuses on training samples in supervised classification for pattern recognition applications. Therefore, the objective of this paper is to present a model (for both vector and raster formats) to specify training samples based on XML (eXtensible Markup Language). The adoption of XML enables flexibility in reusing as well as extending new facilities so that exchange of data for classification among different Pattern Recognition software tools in image of remote sensing is made possible.

1 Introduction

Advances in remote sensing technologies have increased the volume of Earth observation data. The Earth science datasets acquired through remote sensing instruments can be transformed into important scientific discoveries through knowledge discovery techniques [1]. Pattern recognition tools, by employing classification (supervised or unsupervised), are used to find classes (or patterns) of interest which are transformed into useful information to several sectors such as industry and government that can be used in decision making. However, Earth observing data are stored in a wide variety of formats in heterogeneous sources. Converters can be used to transform data into other formats, but the problems, usually faced by the converters, are information loss or alteration.

These formats can range, for example, from simple free formats such as ASCII to complex formats such as GRid In Binary (GRIB), Hierarchical Data Format (HDF) or HDF

[1]Instituto Nacional de Pesquisas Espaciais, INPE, Av. dos Astronautas, 1.758
Jd. Granja - CEP: 12227-010, So Jos dos Campos - SP
{marinalva, dutra}@dpi.inpe.br, vijay@lac.inpe.br

for NASAs Earth Observing System (HDF-EOS). According to [8], there are historical and practical reasons for these different standard data formats. Some agencies developed these formats for their own use and needs and others formats were developed by different communities within Earth Science for specific needs such as compactness and portability. However, as the Earth Science community became more connected, and the scientific problems turned more complex, data sharing encouraged cross collaborations among different agencies and scientists. Earth Science scientists, in order to use data, now have to understand these different data formats. In addition, applications, to analyze the data, used by the scientists must be modified every time a new data format is encountered, or the data must be translated into some other familiar format, which limits the utility of the application.

Many consequences can occur due to this diversity of formats: interoperability among software tools that deal with these data (like Geographic Information System, Pattern Recognition and Image Classification, for example) is more difficult; scientists feel difficult in dealing with each of the data formats, because they may not have expertise in understanding all the different formats [2]. Dealing with heterogeneous formats can become a major bottleneck in the analysis of data. Since most scientists may not be programmers, much of their time and effort are required to understand and decode data from unfamiliar formats. Scientists have to either write a data decoder or reader module for a new data format or translate the data into a format that their analysis tool can handle.

Image processing is extensively applied to extract important features to perform analysis. The extraction of information from satellite images is, more commonly, made using classification techniques. Classification is a process of recognizing categories of objects and labeling entities (normally pixels) to be classified [9]. The classification process may therefore be considered as a form of pattern recognition, in which each pixel represents one point on the Earth's surface.

Classifications may be supervised or unsupervised. In supervised category, the process involves the selection of representative training samples (or training pixels) identified as members of a predefined class whose classification scheme is based on a decision rule (e.g. minimum distance, maximum likelihood) to assign spectral image data to land cover classes [7, 9]. In unsupervised category, a priori pattern is not given to the classifier; instead, it establishes the classes itself based on the statistical regularities of the patterns. According to [7], in general, supervised classification is applied due to higher accuracy level achieved and more robust methods exist, compared to the unsupervised approach.

Procedures for supervised classification of images need to define samples. The selection of samples should be careful, because they must represent the classes, or patterns that one wants to find in the image and have an influence on the classification accuracy. The use of representative training data can help the classifier to produce more accurate and reliable results. Often users need to use classifiers of different systems (like ENVI [16], Spring [17], Idrisi [18], MultiSpec [19], Definiens [20], etc) in search for better classification results for analysis. For this, the same training samples could be used. But this task is not trivial because each system has its own definition to structure and store samples. There are many standards for images such as JPEG, TIFF and others, but most of them are not self-describing and there is no standard that enables the interchange of image structure (image segmentation, regions of interest, classes, statistics, etc.) and training samples. Thus, to achieve interoperability in terms of data (especially in remote sensing) to perform classification of the same data using classifiers of different systems, many steps are necessary to transform the data from one format to another.

Even when systems are flexible enough to export samples in a certain format to another, there is still an issue of embedding some system specific information leading to an extra difficulty to be handled. Normally, the file structure to be exported changes a lot. The use of ASCII format has been seen as an interesting option for enabling the easy transfer of data among different systems, but usually these files have a specific syntax that makes difficult the development of generic readers. XML has a more flexible structure and syntax to facilitate the development of generic readers (or parsers).

Thus, the scope of this work is to guarantee interoperability of traning samples among several applications dealing with supervised classification. In order to achieve such interoperability, a flexible data model, based on XML, was developed. As the model is based on XML, this format can be extended to accomodate characteristics of a specific system and facilitates the development of pasers by systems that may use it. XML structure and syntax are well accepted in most of the scientific community.

Section 2 describes some related researches, section 3 presents the model TSML, section 4 describes the case study performed to validity TSML and section 5 presents the conclusions.

2 Using XML to Facilitate the Access to the Earth Observation Data: Some Related Researches

Markup languages were developed to provide a common medium of data description and display for such systems [6]. XML or eXtensible Markup Language [22] was an attempt in this direction. XML is a text based meta-language. Being a meta-language, other languages can be developed by extending XML for use in specific areas of application. XML supports a richer set of features, such as user-defined tags that allow both data and descriptive information about data to be represented within a single document. The capabilities of XML make it a common data format for data exchange among different applications.

Various efforts based on XML and metadata has emerged as a simple, flexible, and powerful way for computers and applications to communicate and to exchange information. The next sections describe some these efforts. XML became an Internet Standard of W3C as an important new technology of the universal format for structured documents and data on the Web. Most communities of various information systems tend to adopt XML for new data formats for interoperability on the Internet as well as their Intranet. Now, GIS community too adopts new XML based formats for spatial data exchange among applications. Some previous researches, based on markup language that aims interoperability, are: Geographic Markup Language (GML) [25], Remote Sensing Markup Language (RSML) [27] and the XML-based Brazilian format GeoBR [28, 29] and Earth Science Markup Language (ESML) [8].

GML was created by the OGC in 2000 [25] and is based on XML. Its main goal is to store and to transfer geographic information, including spatial and non-spatial properties of features geographical, among applications through the Web. GML stores the data but does not indicate how the data is to be displayed. GML presents the benefits of being an open standard, based on text, uses XML grammar, reduces the costly conversion among different format databases and facilitates the sharing and interchange of data on the web. GML provides a common model for writing schemas, which allows that software that can read any GML schema and interprets that schema to determine which elements are features and which are properties. But in GML the user can write her or his own GML schema in her or his domain area. Being an open standard, GML has been widely used as a means for storage of spatial data from different GIS. But some users find difficult to use GML because they have the impression that it is complex, extensively specified with several objects and has more than 1000 tags.

Aiming to contribute to the area of interoperability of geographic data and to resolve the needs of representing spatial data in the Brazilian context, INPE has developed GeoBR [29], a geographical data model published officially in 2002 with the main objective to be an open standard format for exchanging spatial data. GML, while in version 2.1, did not meet the requirements to become the standard format for exchange of spatial data in the Brazilian case [29]. Therefore, GeoBR was developed and the differences from GML are: a) GeoBR uses a single, generic conceptual model to represent the different types of spatial data. The user does not need to define her or his own data schema, unlike GML; b) GML requires that each institution sets its data schema, which results in additional investment for data conversion; c) GeoBR supports the semantic conversion (ontology) by terms dictionary, while GML does not by itself solve the semantic integration of heterogeneous schemas among different geographic information systems.

RSML was proposed as an attempt to fill the gaps in formats more common in remote sensing. Unlike GML and GeoBR, RSML presents a model of remote sensing metadata used by data producers and a model for raster remote sensing operators/algorithms. RSML is concerned with metadata format and can operate with existing raw, proprietary and graphics image formats. RSML will provide a uniform syntax for producer, application scientist and end user. RSML is rooted on an object-oriented model of the remote sensing metadata requirements and can be extended to accommodate new metadata types using specialization. RSML does not address vector data neither class hierarchy, but the RSML specification for a metadata model for remote sensing data in application environment was the closest approach of the model being described here.

ESML is not a data exchange format by itself. It is different from other works described for neither being a data format nor a data model, but a technology that allows applications to share information without the need to develop a data converter, even if they use different formats from each other. This is possible because ESML provides a standard method for describing the structure of a dataset in any of several common scientific data formats. With this, software developers can build data format independent of their scientific applications utilizing the ESML library. ESML is extensible and new data format can be added in ESML library.

There are three primary components in ESML that enable data and application interoperability:

1. ESML description files contain metadata with content and structural information for the corresponding data file format.

2. ESML schema defines the XML grammar for writing ESML description files

3. ESML library is used by analysis tools to parse the relevant ESML description file for structural information about a data file, and to read the data from the file. The ESML library is written in C++ for Windows and LINUX operating systems and it has interfaces of the ESML library for Python and Interactive Data Language (IDL).

The current schema supports descriptions for unstructured data formats such as ASCII, Binary, GRid In Binary (Grib), network Common Data Format (netCDF), Hierarchical Data Format - Earth Observing System (HDF-EOF) and WSR88D Level II. A separate ESML element is defined for each individual data format. Descriptions for additional data formats can be added to the schema. ESML schema is publicly available via the ESML web site http://esml.itsc.uah.edu. Using ESML and its API (Application Programming Interface), applications can understand and use a data file regardless of its format as long as the format has been fully described using ESML.

Although the researches listed above have been developed to provide interoperability among different systems that make use of such data, none of them has a common model for structuring and storing training samples to be reused by classifiers of different systems.

3 TSML - A Solution to Exchange of Training Samples

The difficulties in exchanging training samples reported in Introduction lead to the development of TSML. TSML (Training Samples Markup Language) is a flexible and extensible structure to store training samples for classification algorithms.

TSML has been designed to be well structured. This fact enables partitioning a training set in such a way that each pixel can be separately accessed directly so that subsets of training samples can be created. As the model is based on XML and it is a fact that several programming languages support dealing with XML, it is a matter of writing a parser to convert the XML file to a specific format needed by a given applications.

The conceptual model of TSML, which includes the main data object that can be part of a process of image classification, is shown in Figure 1.

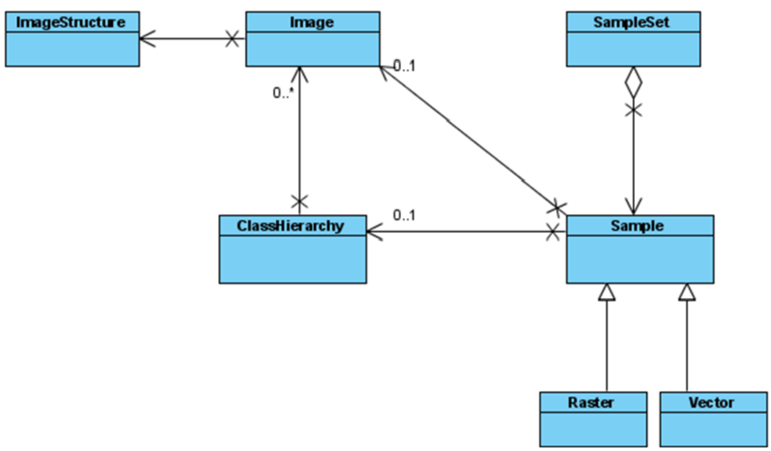

Figure 1. Conceptual Model of Samples

The model shown in Figure 1 shows that the training sample can be both in raster and vector formats, and may have a hierarchy of related classes (classes are the patterns that the samples represent the image, or a priori knowledge about the area covered by the image, for example, forest, agriculture, pasture, water, etc). Once the hierarchy of related classes is developed by a specialist, it can be applied to another image as long as this image has similar categories (forest, agriculture, pasture, water, etc.). The main advantage of applying this hierarchy to another image is that there is no need to develop a different hierarchy; however, it can be extended with more categories.

The XML Schema corresponding to the model is shown in Figures 2, 3, 4, 5, 6 and 7. The XML Schema shown in Figure 2 contains tags describing all data fields required for supervised classification. The XML diagram consists of tags for each information from sample separately. There is a sample set where each sample is composed by:

- Identifier, name and type of sample (training, test or validation), can be in vector or raster formats and also can be geocoded (Figure 2)
- Each sample has an associated class or class hierarchy (className, superClass, subClass, color) (Figure 3). The color is important for identifying each class after the classification (thematic map)
- The sample in raster format has an identifier of each pixel, gray scale for each pixel in each channel, coordinates x and y of each pixel, statistical values of sample like: min, max, number of pixels, mean and other elements not defined can be included (Figure 4)
- The sample in vector format contains geometries and coordinates x and y: point, polygon, polyline, rectangle and ellipse (Figure 5)
- If the sample is geocoded, it has information such as latitude and longitude (Figure 6)
- The image from which samples are extracted has the following information: path, name, number of lines, number of columns, header, number of channels and can contain any other elements (Figure 7).

The XML Schema of the model describes the elements and roles of composition of one sample and it differs in the following aspects from other ASCII-based formats, such as ENVI and Spring. For example:

- As TSML is based on XML, it is structured for easy storage and reading of data related to training samples and image structure.
- All data related to the samples, for both raster and vector, are contained in a single format, TSML. This prevents the user to work with multiple files according to the type of information of the sample he or she wants to use.
- Each element may be an independent entity, so that each system exports or imports only the data necessary to perform the classification. For example, if the samples are geocoded, the elements corresponding to the X and Y coordinates are not necessary to locate the sample in the image.
- The model can accommodate samples in raster or vector, or both.

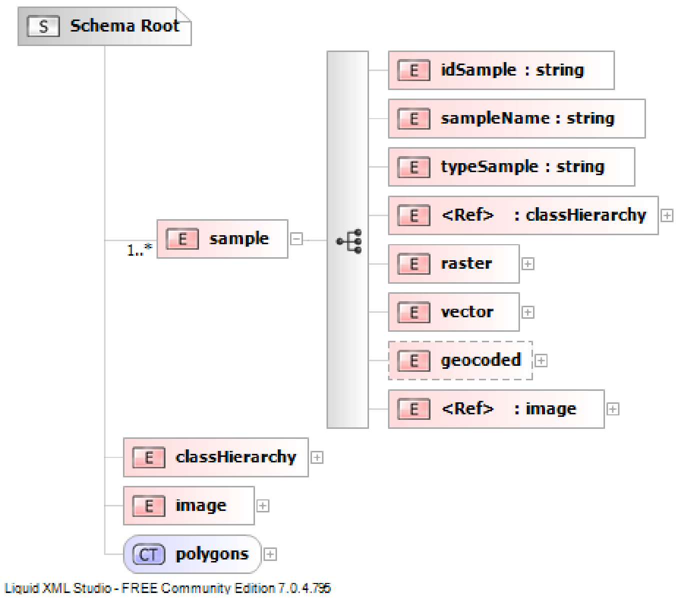

Figure 2. Diagram of the XML Schema for TSML

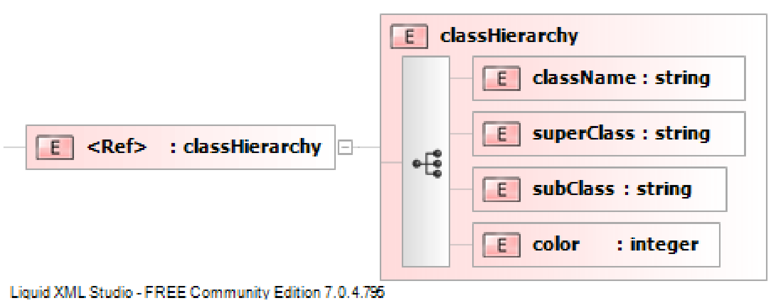

Figure 3. Details of class hierarchy

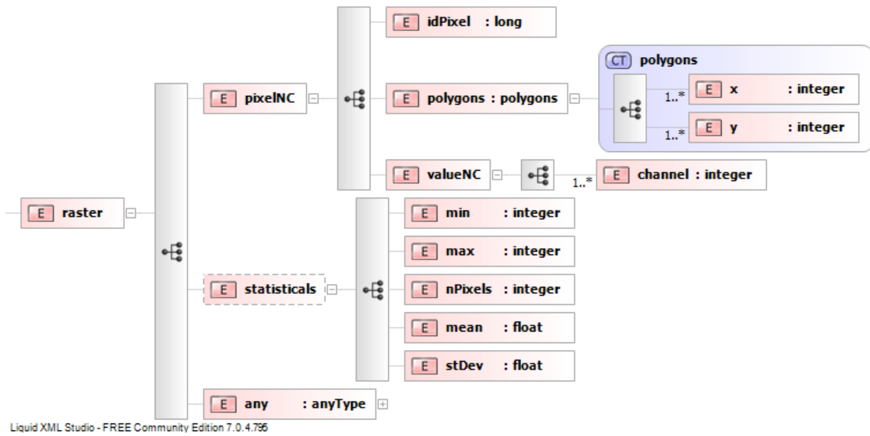

Figure 4. Details of raster format

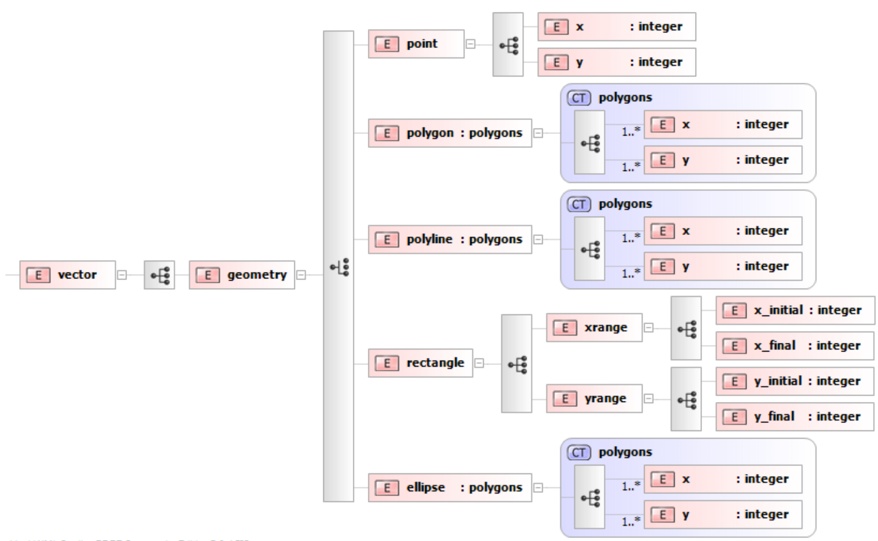

Figure 5. Details of vector format

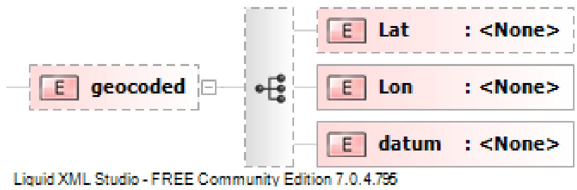

Figure 6. Geocoded sample

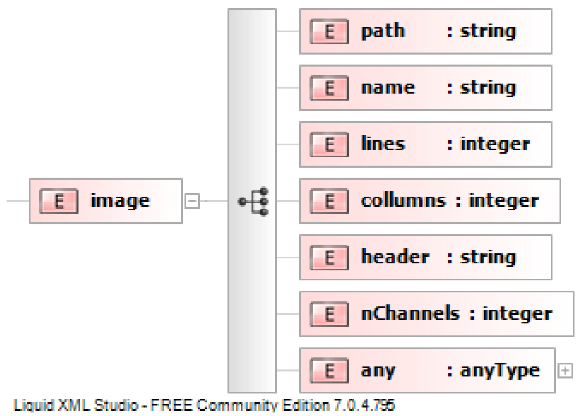

Figure 7. Image information

- The model can accommodate data from the training as the average vector, covariance matrix etc.

Without the use of interoperability, one possible scenario is the one shown in Figure 8. Application 1 imports and exports data in GIF, JPEG and TIFF formats. Applications 2, 3 and 4 have their own formats (DXF, JPEG, ShapeFile; GIF, SAV, ASCII; TIFF, SPR, HDF respectively). It is a fact that all the Applications (2, 3 and 4) have a format in common with Application 1. Therefore, if one wants to exchange data between Applications 1 and 2, it is necessary that the data exported from Application 1 is in JPEG. On the other hand, if a communication between Applications 2 and 3 is necessary, then a conversion is in order. In this case, if Application 3 doesnt have a converter, then the communication has to use Application 1. Application 1 receives the data from Application 2 to be converted (by Application 1) into GIF and exported to Application 3. There is another option of depending on converters in each application. This is a decision of the designer.

Thus, the interoperability occurs by transporting data into a format for a system that reads that format and exports it to the required format. The problem is that the user needs access to those systems, which generally are commercial sofwares.

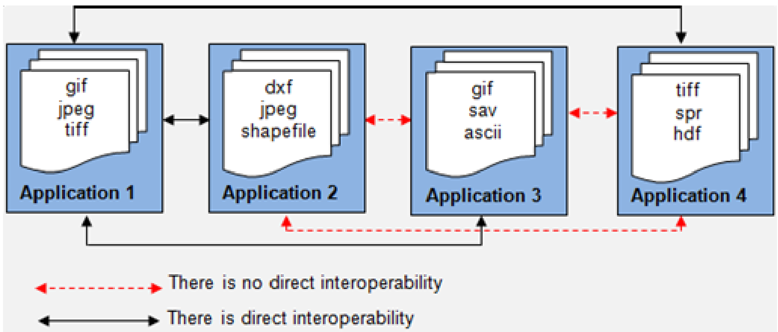

Figure 8. Scenario of data interchange among remote sensing system

Figure 9 shows the interoperability among different applications through TSML. The application that wants to import samples in TSML format should just build a parser that reads XML. Since most programming languages has operators to deal with XML files, the development of a parser for such systems is not complex.

In the architecture illustrated in Figure 9, there is a possibility for support to the semantic equivalence through a dictionary of ontologies (or terms) that establishes common concepts. One possible implementation for this could be made through a function, whose input data are the elements (terms) of TSML file and terms of the application. This dictionary could be updated by applications that can also export it with the TSML file to other applications. However, the semantic equivalence was not implemented as this is not the scope of this paper.

In systems that access the data directly into a relational database, for example, TSML also can be converted into a relational schema to storage in a database. There are several approaches to store XML data in relational databases [31]. The most straightforward approach would be to store an entire XML document in the database. Another approach is the mapping of XML document structure to a relational schema. The XML data is stored in relational columns preserving hierarchical relationships between XML data. For this, an XML document can be mapped to multiple tables. However, any of the approaches suggested are trivial and this is not the scope of this paper.

There are tools that are not designed directly for image processing and classification, but their algorithms can be used for this purpose as long as the data input is properly prepared

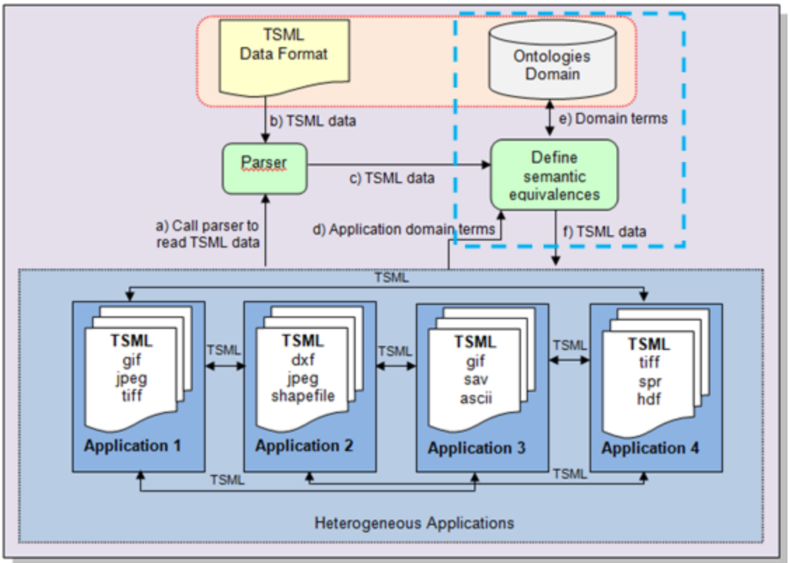

Figure 9. Interoperability among different applications through TSML

according to their formats. As one example, we can cite the Weka [30], which has several data mining algorithms and data classification such as Bayes, Tree, Neural Network and others. However, in order to use Weka, the data must be structured into a format that can be properly accessed. As an experiment, a simple application has been developed to use Weka that will be explained in section 4.

The design of TSML caters for organizing the samples into categories. For example, one can organize the samples in categories such as training, tests, validation, unknown, etc. ENVI, for example, does not make difference in its samples. It is impossible to label this category whether it serves as a training sample, testing sample, etc. This task is entirely up to the user if she or he wants to organize the samples into different categories. On the other hand, TSML with the facility to categorize the samples enables the applications to tailor scripts to deal with the specific classes. It is important to note that the exchange of samples among different systems can be done by exporting TSML to the format that the other systems read or importing TSML directly from other system through a parser. The case study described in the section 4 uses both the types of exchange.

4 Case Study

In order to validate the exchange of samples in TSML, a test with two distinct systems, SACI (Sistema de Anlise e Classificao de Imagens - Image Analysis and Classification System) and ENVI, was performed. SACI is coded in IDL [16] and consists of both deterministic and statistical classifiers. ENVI is an important commercial software used for image analysis and classification in several countries. The area under study for validating TSML was from Tapajs National Forest from Amazon region and the image selected was from Landsat-TM using bands 3, 4 and 5. The training samples were extracted in SACI and stored in the only available "SAV" format, a binary format of IDL. In order to show the applicability of TSML, a module was developed within SACI to store these samples in TSML format and now TSML is the default format of samples within SACI. The Maximum Likelihood algorithm was used to train the samples and to classify the image under study.

In the next step, these samples were imported by ENVI. However, it is essential to note that a modification in ENVI had to be implemented so that it could import TSML, which is the first form of the exchange cited in the section 3. After importing, Maximum Likelihood algorithm was used to train the samples and to classify the image.

Although SACI and ENVI are two distinct systems, the algorithm to train the samples and to classify the image was the same. It is expected to obtain results with a high degree of similarity. The results of classification from both the systems as shown visually in Figure 10 and numerically with the confusion matrix in Tables 1 and 2 were in fact very similar.

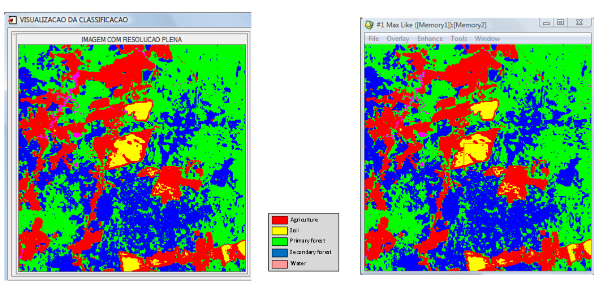

Figure 10. Classification in SACI (left) and ENVI (right) systems with Maximum Likelihood algorithm

The results obtained using the Maximum Likelihood algorithm implemented in SACI and ENVI present a small confusion only for classes soil, secondary forest and water. In SACI, only 3 pixels of the soil class were classified as water, 3 pixels of the secondary forest class were classified as soil and 1 pixel of the water class was classified as soil. This confusion differs from that in ENVI only in soil class, with 2 pixels classified as secondary forest and 8

Table 1. Confusion matrix of the classification in SACI with Maximum Likelihood.

	Agriculture	Soil	Primary forest	Secondary forest	Water
Agriculture	**252**	0	0	0	0
Soil	0	**501**	0	3	1
Primary forest	0	0	**270**	0	0
Secondary foret	0	0	0	**231**	0
Water	0	3	0	0	**59**

Table 2. Confusion matrix of the classification in ENVI with Maximum Likelihood.

	Agriculture	Soil	Primary forest	Secondary forest	Water
Agriculture	**252**	0	0	0	0
Soil	0	**494**	0	3	1
Primary forest	0	0	**270**	0	0
Secondary foret	0	2	0	**231**	0
Water	0	8	0	0	**59**

pixels as water. The kappa coefficient in SACI was 0.99 and in ENVI was 0.98. The kappa coefficient is a measure of agreement, i.e., how close the observed values are with those expected. The values obtained in SACI and ENVI are compatible and acceptable in both systems and are excellent.

The same samples were used for classification using a neural network implemented in ENVI. The results are similar and the confusion matrix is showed in Table 3. One can observe that values are very similar to the results showed in Table 1 and 2. The same samples could be used in a neural network implemented in Matlab tool, for example, and others. Therefore, the TSML format is promising in the exchange of training data enabling the use of different algorithms in different systems.

Table 3. Confusion matrix of the classification in ENVI with neural network.

	Agriculture	Soil	Primary forest	Secondary forest	Water
Agriculture	**252**	0	0	0	0
Soil	0	**494**	0	3	1
Primary forest	0	0	**270**	0	0
Secondary forest	0	2	0	**231**	0
Water	0	8	0	0	**59**

Another test, which aimed to analyze the separability of the classes in the 1, 2 and 3 channels, was performed with the same samples set in Weka tool. Unlike the exchanges between SACI and ENVI, the exchange between SACI and Weka was performed by exporting of the samples directly from SACI for the Weka's ARFF format, which is the second form of exchange cited in the section 3. To verify the separability of the classes, J48 algorithm (not available neither in SACI nor in ENVI) was applied on samples. Figure 11 shows the generated tree.

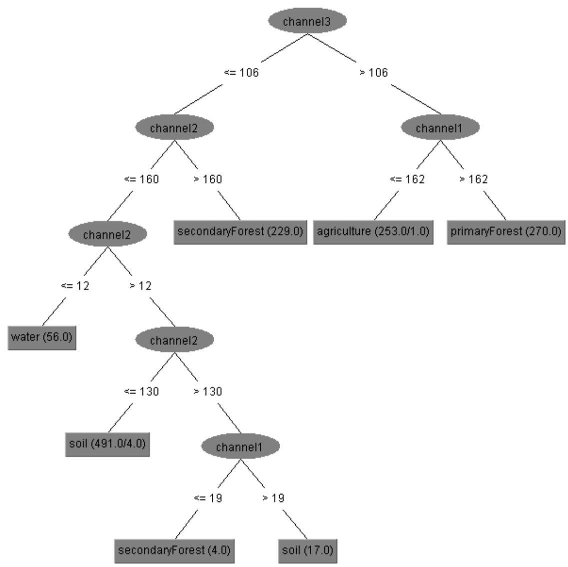

Figure 11. Tree generated in Weka

One can observe that classes agriculture and primary forest have a good separability in channels 1 and 3. However, channel 2 presents difficulty in separating water, soil and secondary forest classes.

5 Conclusions

This paper has shown a model of training samples structured in such a way that is flexible and extensible. Being based on XML, this model allows the exchange of samples with minimal effort to import the data without their loss or alteration. Tests of the importing samples in ENVI were performed to show the feasibility of the model through the direct import of data into XML. Thus, it was possible to compare the classification performed in SACI and ENVI with the same algorithm and also to use the neural network from ENVI for comparison of different algorithms. The results of the classification were similar to those of the original using SACI system.

TSML is a promising format because the data is represented using the XML document structure. The elements provide useful metadata and structure the data in such a way that they can be transferred and manipulated by different applications. Furthermore, these data can be changed to meet the needs of any specific application.

TSML promotes the reuse of the samples among heterogeneous systems and reduces the effort required in developing a converter of unknown or complex formats.

The samples were exported directly from SACI to Weka for analysis of separability of the classes. This test was possible due to the ease of conversion of TSML format to ARFF format. The results confirm the advantages of the model and reinforce its importance for exchange among different classification systems and activities that deal with combination of results of classification from different classifiers. More tests should be performed, but the closed source commercial software hinder this process. Such tests could be performed in ENVI because it allows modules to be added using IDL. So it was possible to develop the parser to read the TSML in ENVI.

References

[1] CARAGEA, D.; HONAVAR, V. Knowledge discovery from disparate earth data sources. In: Second NASA Data Mining Workchop: Issues and Applications in Earth Sciences. Poster Session. Pasadena, CA: NASA, 2006.

[2] RAMACHANDRAN, R.; CRISTOPHER, S. A.; MOVVA, S.; LI, X.; CONOVER, H. T.; KEISER, K. R.; GRAVES, S. J.; MCNIDER, R. T. Earth science markup language: a solution to address data format heterogeneity problems in atmospheric sciences. American Meteorology Society, p. 791-794, june 2005.

[3] THEODORIDIS, S.; KOUTROUMBAS, K. Pattern Recognition. Second edition. Elsevier, 2003.

[4] WEBB, A. R. Statistical Pattern Recognition. 2. ed. England: John Wiley Sons, Ltda, 2002.

[5] JAIN, R. P. D. A. K.; MAO, J. Statistical pattern recognition: A review. IEEE Transactions on Pattern Analisys and Machine Intelligence, IEEE, v. 22, n. 1, p. 4-37, 2000.

[6] SRIPADA, L. N.; LU, C.-T.; WU, W. Evaluating gml support for spatial databases. In: Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC04). IEEE Computer Society, 2004.

[7] KAVSOGLU, T. Increasing the accuracy of neural network classification using refined training data. Environmental Modelling and Software 24 (2009), pp. 850-858. www.elsevier.com/locate/envsoft

[8] RAMACHANDRAN, R.; GRAVES, S.; CONOVER, H.; MOE, K. Earth science markup language (esml): a solution for scientific data-application interoperability problem. Computer and Geociences, v. 30, p. 117-124, 2004.

[9] MATHER, P. M. Computing Processing of Remotely-Sensed Images. John Wiley. 2004.

[10] LU, D., WENG, Q. A survey of image classification methods and techniques for improving classification performance. International Journal of Remote Sensing. Vol. 28, No. 5, 10 March 2007, 823870.

[11] AL-ANI, A.; DERICHE, M. A New Technique for Combining Multiple Classifiers using The Dempster-Shafer Theory of Evidence. Journal of Artificial Intelligence Research. Vol. 17, 2002, PP. 333-361.

[12] CHEN, K.; WANG, L.; CHI, H. Methods of Combining Multiple Classifiers with Different Features and Their Applications to Text-Independent Speaker Identification. International Journal of Pattern Recognition and Artificial Intelligence, 11(3), 1997, pp. 417-445.

[13] DARA, R. A.; KAMEL, M. Effect of Sharing Training Patterns on the Performance of Classifier Ensemble. 2004 IEEE International Conference on Systems, Man and Cybernetics.

[14] HATAMI, N.; EBRAHIMPOUR, R. Combining Multiple Classifiers: Diversify with Boosting and Combining by Stacking. International Journal of Computer Science and Network Security, VOL.7 No.1, January 2007.

[15] NAJAFI, M.; JAMZAD, M. A Solution of Combining Several Classifiers for Face Recognition. Proceedings of the World Congress on Engineering and Computer Science 2007, WCECS 2007, October 24-26, 2007, San Francisco, USA.

[16] ENVI. ITT Visual Information Solutions. http://www.ittvis.com/

[17] CAMARA, G.; SOUZA, R. C. M.; FREITAS, U. M.; GARRIDO, J. Spring: Integrating remote sensing and gis by object-oriented data modelling. Computer and Graphics, v. 20, p. 395-403, 1996.

[18] IDRISI. Idrisi Homepage. http://www.idrisi.com

[19] BIEHL, L.; LANDGREBE, D. MultiSpec - A Tool for Multispectral-Hyperspectral Image Data Analysis. 13th Pecora Symposium, Sioux Falls, SD, August 20-22, 1996. http://cobweb.ecn.purdue.edu/~biehl/MultiSpec/

[20] DEFINIENS.Definiens: Inderstanding Images. http://www.definiens.com

[21] GARCIA, V.; SNCHEZ, J.S.; MOLLINEDA, R.A.; ALEJO, R.; SOTOCA, J.M. II Congreso Espaol de Informtica,Zaragoza, Spain, 2007, 283-291.

[22] W3C. XML - eXtensible Markup Language. http://www.w3.org/XML/

[23] FGDC Homepage. http://www.ofgdc.gov/

[24] OGC Homepage. http://www.opengeospatial.org/

[25] Geography Markup Language, http://www.opengeospatial.org/standards/gml

[26] ISO/TC 211 Geographic information/Geomatics. http://www.isotc211.org

[27] MOHANTY, K. K. Rsml: A proposed xml based format and processing language for remote sensing data. Indian Cartographer, p. 330-335, 2002.

[28] JUNIOR, P. L.; CAMARA, G.; PAIVA, J. A.; MONTEIRO, A. M. V. Intercambio de dados geograficos: Modelos, formatos e conversores. In: Anais do I WorCap - Workshop de Computao Aplicada. Sao Jose dos Campos, Brasil: INPE, p. 36-38, 2001.

[29] JUNIOR, P. de O. L. GEOBR: Intercambio Sintatico e Semantico de Dados Espaciais. INPE, 2002. Dissertacao de Mestrado. http://mtc-m16.sid.inpe.br/col/sid.inpe.br/jeferson/2004/09.20.09.49/doc/publicacao.pdf

[30] WITTEN, I. H.; FRANK, E. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, June 2005. ISBN 0-12-088407-0.

[31] GRAVES, M. Projeto de Banco de Dados com XML. Pearson Education, 2003. ISBN 85-346-1471-7.