Architecture of an Interactive Classification System

Ilze Birzniece
Riga Technical University, Department of Systems Theory and Design
Riga, Latvia
ilze.birzniece@rtu.lv

***Abstract* – The paper describes the design of interactive inductive learning-based classification system. The architecture of machine learning systems can be viewed from two perspectives, namely, (1) the stages of system design and (2) model of system's functioning and components. Both of these design issues of different existing classification systems are discussed in the related work. A general architecture for the interactive classification system is proposed. Domain-dependent parts of the system are specified in the more detailed architecture of the interactive multi-label classification system for study course comparison. Interactive inductive learning-based classification system in uncertain conditions could ask a human for decision, and it is has been proven that applying this approach can reduce the number of misclassified instances, especially, when the initial classifier performs poor.**

Keywords–classification; inductive learning; machine learning; software architecture; supervised learning.

I. INTRODUCTION

Machine learning (ML) is the ability of a computer program to improve its own performance, based on the past experience [1]. Classification is one of ML tasks where the program learns to classify new instances from a human or environment provided training set. Classification problems arise in a number of areas, like credit scoring, pattern recognition, medical diagnostics, document classification, etc.

Motivation for creating an interactive classification system comes from several sides. One of them is inappropriateness of the automated classification methods for all domains where ML techniques could be applied to. Application domains are getting more complex in terms of data amount, representation forms, relationships within data, etc. In the real world information is often organized in vague or complicated forms like plain text, semi-structured text, graphs, etc. The transformation from original data to classifier-acceptable data structures is needed, and in this process some information can get lost or mapped inaccurately. This leads to creation of an incomplete classifier that does not generalize well the problem domain and probably will not be able to make predictions for all new unseen instances when the classifier is applied. Consequently, ML approaches face new challenges in solving tasks which could benefit from automated solutions but do not conform to typical ML application areas. Furthermore, people who are well aware of the complexity of the domain usually do not believe in a fully automatic approach and are ready to invest some efforts towards a more suitable solution [2].

Other facilitator for developing an interactive classification system is the practical need in the area of curricula comparison. This task is very time-consuming for humans and is also not trivial for application of ML methods directly because of the mixture of domain features.

Therefore, the mechanism for human involvement in handling instances that cannot be classified using only the classifier is proposed [3]. This mechanism (1) deals with instances which the classifier was not able to classify, by asking a human to decide a classification, and (2) improves the classifier's knowledge base with the rules derived from this experience. There is no single agreement on using this term in the literature, therefore, in this work, an "interactive classification system" is denoted as a system which involves human in handling instances that the classifier is not able to classify.

In our previous research on development of the interactive classification system, different existing approaches of interactivity in the classification process have been examined [3], and ways of incorporating human classified instances into the classifier revealed [4]. Proposal to apply the interactive classification approach to the university study course comparison problem has been given in [5], defining it as a ML task in [6] and adding a formal background in [7]. In [8] a common sight over curricula comparison as a problem of both information extraction and classification is given. This paper follows the suit and proposes general architecture for interactive classification systems, as well as specifies a more detailed architecture of the interactive multi-label classification system in a domain of study course comparison. The background for development of the interactive classification system is based on different existing architectures of "non-interactive" classification systems due to the lack of detailed interactive system descriptions. The intended interactive system is to be built using the best practices from former approaches and reusing ideas where appropriate.

The paper is organized as follows. Section II surveys the related work on different existing classification system architectures, starting with the general system's design, which is common for interactive and non-interactive systems, and following with system's functioning, which is separated for both types of systems. The architecture of the proposed interactive classification system is introduced in Section III. Section IV defines the particular classification problem and gives a short description of design decisions towards the interactive multi-label classification system for university study course comparison. Conclusions and the intended future work are given in Section V.

 ISBN: 978-1-61208-232-5

II. ARCHITECTURE AND DESIGN OF CLASSIFICATION SYSTEMS

Representation of architecture of ML systems can be taken from two viewpoints, namely, (1) the stages of system design and (2) model of system's functioning (components). This section amalgamates design stages and models of system's functioning from a wide variety of authors, represented in a joint format by the author of this paper. Summarization of different existing architectures, especially from the two mentioned viewpoints, to the best of author's knowledge, has not been done before. Existing approaches are being analyzed and compared regarding the common elements in them. Typical communalities found are denoted in schemas with the same representation (bold block lines and interrupted block lines) and summarized at the end of subsection *A*.

In this section, firstly, proposed approaches for the system's development life cycle will be described, and, secondly, system's functional models will be explained. System's development stages do not vary much regarding the amount of interactivity built into the system, but system's functioning is different in these cases, therefore, topic is discussed separately for "non-interactive" and interactive classification systems.

To clarify the terms used in this paper, the difference between the classifier and the classification system has to be explained. In the context of the paper, the classifier means the exact model or rule set according to which a new unseen instance can be classified, whereas the classification system is an extended functional structure which allows to pre or post-process data and applies the classifier. Designing of the classification system includes designing the classifier. The latter is produced by a ML method, in this case supervised learning algorithms which induce the classification model in the form of a tree or If-Then rules. Thus, the classification system is a classifier and its peripherals which ensure the classification process.

There are different types of ML applications, therefore, in literature the corresponding systems are named variously. Some are called pattern recognition systems [9], others are called classification systems [10], inductive learning systems [11], inductive learning technique applications [11] or just learning systems [12]. However, they share the same fundamental elements in the path that is followed to design the application [11]. These systems can also be a part of intelligent systems, since the definition of an intelligent system is "system that learns during its existence"[13]. Thus, the characteristics of intelligent systems are also applicable to classification systems. In this paper, the term "classification system" will be used, unless the authors of the reviewed literature had insisted on defining it otherwise.

A. Designing classification systems

In the design theory, there are several types of design problems. In general, design tasks can be divided into three classes [14], [15], which can be characterized as follows.

1. Routine

In the routine design, knowledge sources and problem-solving strategies are generally known in advance and a priori plan of the solution exists.

2. Innovative

In the innovative design, problem-solving strategies are generally known, but the problem lacks a set of constraints. It can be called as an original combination of existing components.

3. Creative

In the creative design (also called the original design), neither problem-solving strategy nor knowledge sources are known which leads to a major invention or an entirely new product.

In the space of possible designs, the routine design involves implementation of a known type; the innovative design involves generation of new subtypes; the creative design involves generation of entirely new types [15]. Some authors, e.g., [16], suggest usage of the fourth class, namely, redesign.

By the definition of design problems, design of the "standard" classification system is more like a routine design task with choosing the right components and tuning the parameters.

Cherkasskey [17] claims that good understanding of the whole classification procedure is important for any successful application. He adapts approach from Dowdy and Wearden [18] and presents the general experimental procedure for development of the classification system (see Figure 1).

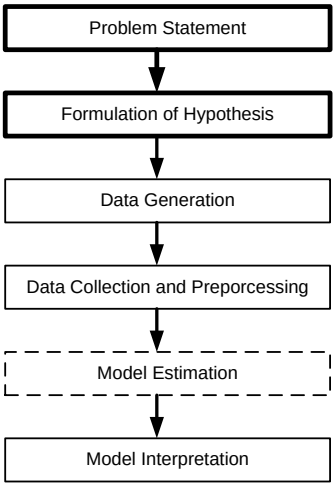

Fig. 1. Design stages adopted from Dowdy and Wearden [18]

- Statement of the problem

Domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. It is important not to focus on the learning methods used instead of a clear problem statement.

- Hypothesis formulation

The hypothesis in this step specifies an unknown dependency, which is to be estimated from experimental data. At this step, a modeler usually specifies a set of input and output variables for the unknown dependency. There may be several hypotheses formulated for a single problem.

- Data generation/experiment design

 ISBN: 978-1-61208-232-5

This step is concerned with how data is generated – under the control of a modeler or not. Further, it is important to make sure that the past (training) data used for model estimation and the future data used for prediction, come from the same (unknown) sampling distribution. If this is not the case, then, in most cases, predictive models estimated from the training data alone cannot be used for prediction with the future data.

- Data collection and preprocessing

This step has to do with both data collection and the subsequent preprocessing of data. Data preprocessing includes at least two common tasks: outlier detection and removal and data encoding and feature selection.

- Model estimation

The main goal is to construct models for accurate prediction of future outputs from the known input values.

- Interpretation of the model and drawing conclusions

In many cases predictive models need to be used for human decision making. Hence, such models have to be interpretable in order to be useful because humans are not likely to base their decisions on complex ‘‘black- box’’ models. Note that the goals of accurate prediction and interpretation are rather different because interpretable models would be simple but accurate predictive models might be rather complex.

Different design components of a learning system are given by Mitchell in his notable book “Machine Learning” [12]. Figure 2 demonstrates the involved steps.

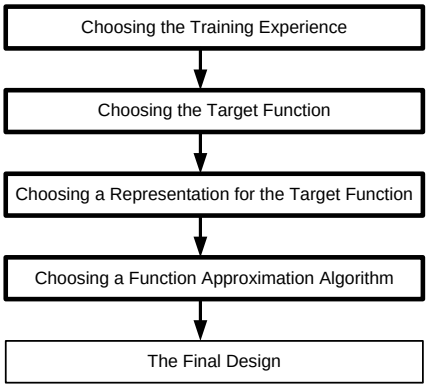

Fig. 2. Design stages adopted from Mitchell [12]

- Choosing the Training Experience

The first design choice is training experience from which the system will learn. It is very responsible decision because training experience significantly impacts success or failure of learning system. Learning experience could be used directly or indirectly, with teacher-provided or self-generated learning examples and can represent real examples distribution more or less precisely.

- Choosing the Target Function

It is also, sometimes, hard to define the choice. If the target function is too difficult to learn perfectly, some approximation can be applied instead.

- Choosing a Representation for the Target Function

The choice of representation involves a crucial tradeoff between expressiveness and simplicity.

- Choosing a Function Approximation Algorithm

In order to learn a target function, a set of training examples is required. Examples are derived from training experience and the learning algorithm is specified for choosing the weights to best fit the set of training examples.

- The Final Design

The final design phase leads to the system’s model, and the learning system is naturally described by four distinct modules that represent the main components in many learning systems. It will be described in the next subsection.

Design process is also sufficiently discussed in the context of pattern recognition systems. These systems share similar development stages; variations are in the focus and features of particular patterns. Figure 3 represents a development model of the classification system which is adapted from both [9] and [10].

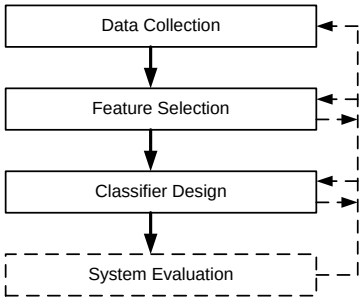

Fig. 3. Design stages of a pattern recognition system

As is apparent from the feedback, stages are interrelated and can be used to return and redesign earlier stages.

- Data Collection

This stage constitutes a large part of the entire time and effort for designing a classification system.

- Feature Selection

If necessary, this includes feature generation and extraction. This is similar to “Choosing the Training Experience” in Mitchell’s [12] model, and also is defined as a critical design step. Prior knowledge about application domain plays a major role in choosing features. One desires features which are simply to extract, invariant to irrelevant transformations, insensitive to noise, and useful for distinguishing different classes.

- Classifier Design

This stage also stands for Choosing Model and Classifier Training. A lot of design questions should be considered here, including the class of algorithms to apply, choice of particular method to use, specific parameters, etc. It might involve serious analysis or experimentation to decide upon these questions.

- System Evaluation

Evaluation of results is important both to measure the performance of the system and to identify the need for improvements in its components.

Vardenius and Someren, in their survey [11] about the application of inductive learning techniques (ILT), argue that one “must take a broader view than strict application of an ILT

 ISBN: 978-1-61208-232-5

to a dataset". They claim that an ILT application is a project the result of which is (1) either a system that can support the user in solving his problem, or (2) a body of knowledge that enables the user to solve the problem himself. The project approach can be described in the form of process models.

An approach for developing a classification system which also covers all relevant levels of the project life cycle is proposed by UK Department of Trade and Industry [19] (see Figure 4). The system's design includes some stages relating project management that were not encountered in the models mentioned earlier in this section. However, the formal aspects of the process are not very well developed in this design model switching the focus on monitoring and control possibilities of application development.

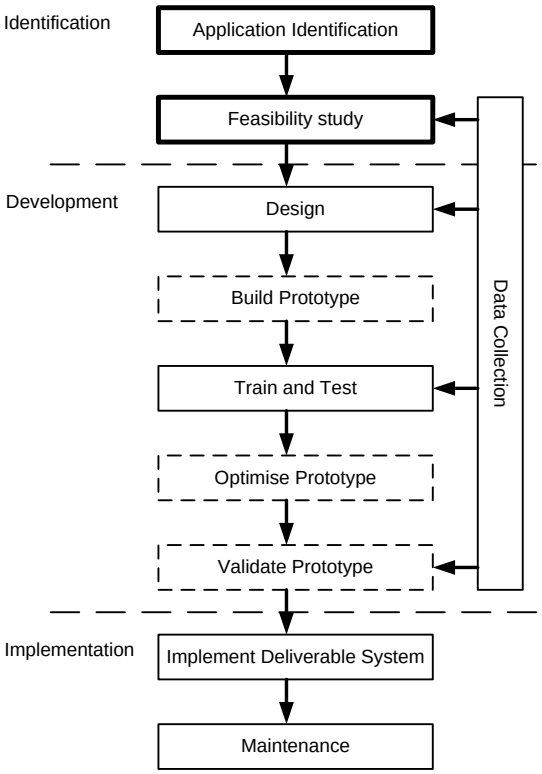

Fig. 4. Design stages adopted from [19]

As stated in the beginning of the section, classification systems fall under the intelligent system category. Bielawski and Lewand in their book [20] propose five step procedure for designing intellectual systems (see Figure 5).

The most important are said to be the first two steps. It also corresponds to previously discussed opinions from other authors.

Vardenius and Someren conclude that there is no uniform view on inductive learning system development. However, they find that the process of classification system application consists of three levels and a control element.

- Application level

In this level, a real world problem is to be analyzed, including identification of resources (such as data, human experts), decomposing the problem, constructing a conceptual model, defining the scope of solution. ML approaches can be used to solve the whole problem or just a part of it.

- Analysis level

This level includes data acquisition, attribute selection, pre-processing, etc. Very important part of this stage is selection of one or more appropriate learning techniques.

- Technique level

Additional choices about selected learning algorithms could be considered, e.g., different parameters of method application.

- Control element – project management

During execution of all levels, decisions and constraints should be taken. This element ensures implementation of the learning system in line with actual user needs.

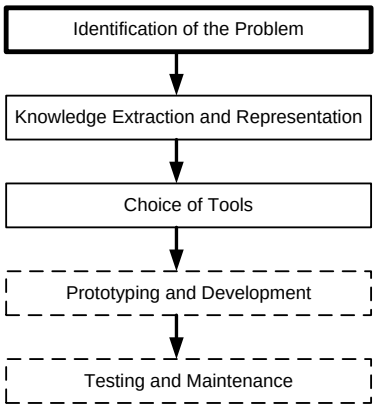

Fig. 5. Intellectual system design steps [20]

There are several conclusions that can be made after reviewing classification system development approaches. Most of the models described in this section give a great weight to initial stage which is called either a problem statement and formulation of hypothesis [18], identification of the problem [20], application identification and feasibility study [19] or incorporates a set of stages from choosing the training experience to choosing a function approximation algorithm [12] and denoted with bold block lines. Another common thing is that the design process of the classification system in some form should contain analysis for choosing the best solution for a particular task. Creating a classification system for a new application is rarely the case of one-way direct software implementation; therefore, the search for appropriate classification system elements (algorithms, methods, parameters, etc.) is done either in analytical way or carrying out experiments (shown with interrupted block lines or feedback arrows in schemas), or even implementing a prototype (like in the models of [19], [20]).

In general, the design process of an interactive classification system does not differ from a design of a non-interactive system, therefore, in this context no need for a new approach arises.

B. Functioning of classification systems

If design explains how to create a classification system, what actions to take and which questions to consider, the

 ISBN: 978-1-61208-232-5

architecture of the system describes how the final system operates and from which parts it consists of. In this aspect arises the need for diversification of non-interactive and interactive system architectures.

1) Non-interactive classification systems

A simple but accurate schema of classifier's functioning is given by Han and Kember in [21]. Similar models are presented also by other authors. In Figure 6, the data classification process can be separated in two stages. In the learning part, training data is analyzed by a classification algorithm. The learned model or the classifier can be represented in different forms, e.g., classification rules. In the classification part test data is used to estimate the accuracy of the classifier. If the accuracy is considered acceptable, the rules can be applied to the classification of new data.

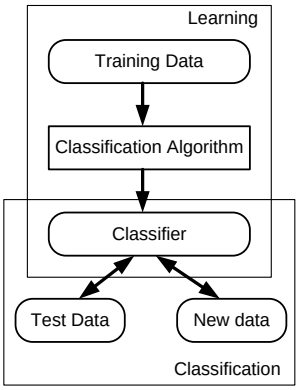

Fig. 6. Classifier building an applying model

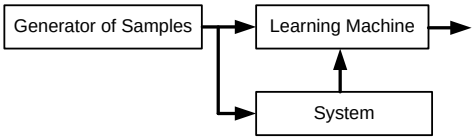

Fig. 7. Learning scenario components by Cherkassey [17]

However, this model does not qualify for a classification system; it is only the schema for building a classifier and is a part of a classification system.

Cherkassey [17] presents a general learning scenario which involves three components: Generator of random input vectors, System that returns an output for a given input vector, and the Learning Machine that estimates an unknown (input, output) mapping of the System from the observed samples (see Figure 7). The given formulation is very general and describes many practical learning problems found in engineering and statistics, including classification.

Another learning task is given by Mitchell [12]. He describes a learning cycle of the system which improves its own performance through repetition (see Figure 8).

- Performance System

This is the module that solves the given performance task by using the learned target function(s). It takes an instance of a new problem as input and produces a trace of its solution as output.

- Critic

Takes the history as input and produces as output a set of training examples of the target function.

- Generalizer

Receives the training examples as input and produces an output hypothesis as its estimate of the target function. It generalizes from the specific training examples.

- Experiment Generator

Its role is to pick up new problems that will maximize the learning rate of the overall system. It takes the current hypothesis (currently learned function) as input and outputs a new problem for the Performance System to explore.

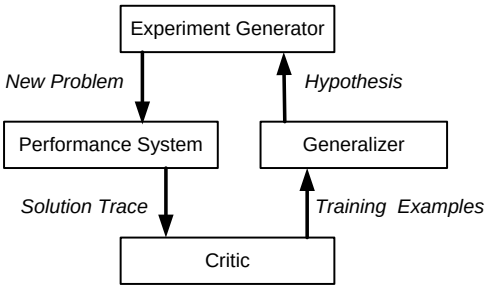

Fig. 8. The main components in learning systems by Mitchell [12]

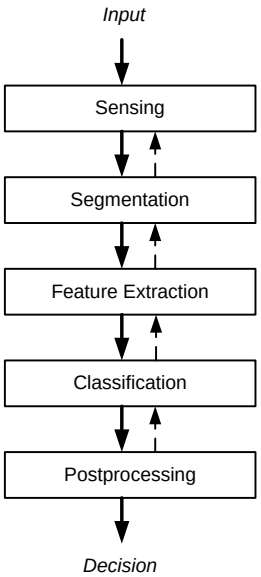

Fig. 9. The components of a typical pattern recognition system [9]

Figure 9 shows a diagram of the components of a typical pattern recognition system [9]. A sensor converts system inputs into signal data. The segmentor isolates sensed objects from the background or other objects. A feature extractor deals with object properties that are useful for classification. The classifier uses extracted features to assign an object a class. The post-processor takes into account other considerations to make a decision of further actions. Although the description stresses a one-way data flow, the feedback from higher levels back to lower levels is also possible.

 ISBN: 978-1-61208-232-5

2) Interactive classification systems

Regarding architecture descriptions of interactive classification systems they are few, scattered and of different types. Most well known interactive approaches to classification are based on active learning [22], data visualization (e.g., [23]) and ripple down rule [24].

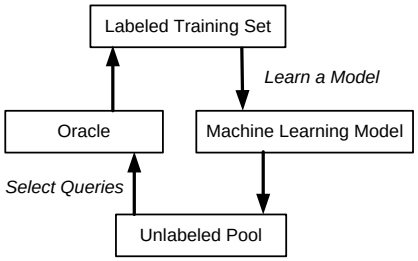

Fig. 10. The pool-based active learning cycle [25]

Active learning is a subfield of machine and is based on hypothesis that if the learning algorithm is allowed to choose the data from which it learns. it will perform better with less training [25]. Figure 10 illustrates the most common type of active learning – the pool-based active learning. A classification system may begin with a small number of instances in the labeled training set, request labels from the oracle (usually a human) for one or more selected instances from the pool, and learn from the query results. There are several scenarios in which active learners may pose queries, and there are also several different query strategies to decide which instances are the most informative. The architecture of system's functioning is given in [23].

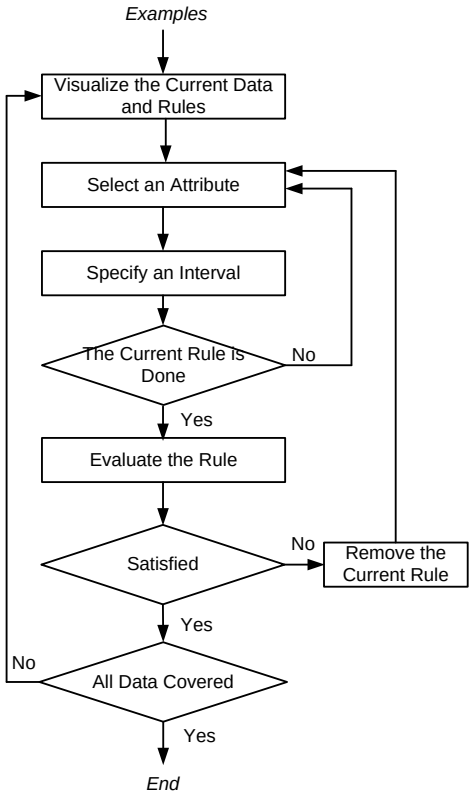

Fig. 11. Rule construction in CVizT system [23]

CViZt system's part for building rules is depicted in Figure 11. Building of the classifier is to interactively and iteratively construct classification rules one by one. The aim of visualizing the current data and rules is to give a look into distribution of the dataset and ease perceiving correlations between attributes. Rule construction consists of selecting attributes and their respective interval of values which is done by a human. A potential rule is automatically evaluated. If the rule accuracy is greater than the pre-specified threshold, then it is accepted and appended to the classifier. The process is repeated until all examples are covered by rules. This approach relates to rule construction by human which is in a sense similar to ripple down rules. The latter is one of the approaches to directly acquire and encode knowledge from human experts.

Several research papers of the last twenty years refer to the concept "interactive inductive learning" or explore the idea of human interaction in the concept learning process. Systems and approaches proposed in these papers are from distinct fields and suggest different types of human interaction. The following types of human interaction are described in [26-32].

1. Systems where the human feedback is asked to evaluate only the given result (decision or prediction).
2. Systems that learn concept classification based on the classification by human.
3. At first, human is giving his/her knowledge to the system and affirming the rules that are induced by the system afterwards.
4. The human evaluates and selects the rules induced by the system in the classifier forming stage.
5. Learning systems where the human is the learner and the computer should be able to interact in a user-friendly way.

The full survey of the above-mentioned related works can be found in [3]. Interaction with a human in these systems takes place in different phases of learning. However, no explicit architectures are provided there. Although these approaches are interactive, they do not conform to the problem being addressed in this paper – creating the classifier automatically and involving a human to deal with unclassified instances.

To sum up the related work, the architecture of classification systems is described in literature quite widely. Differences between the given architectures are determined mainly by focus, scope or the intended application area. However, there are no major contradictions between them. Descriptions of classification system architectures usually are either in terms of general classifier building guidelines or summary of very abstract components. We assume that more detailed architectures of classification systems are domain specific and hard to reuse for other purposes (e.g., CVizT system), therefore, not widespread across scientific literature. Every new case requires a problem domain analysis with respective design decisions. Therefore, also the interactive classification system has to be designed on demand, taking into account the specificity of the need for computer-human interaction in the final architecture.

 ISBN: 978-1-61208-232-5

III. INTERACTIVE CLASSIFICATION SYSTEM

This section explains the proposed interactive classification system's architecture from different viewpoints. Stages of the system's design will be described in the next section when a particular classification problem will be set, which will serve as a background for making domain-specific design decisions.

A. Main components of the interactive classification system

Figure 12 shows the tasks which should be carried out within a classification system for domains with complex data types and the need for appropriate pre-processing and structuring.

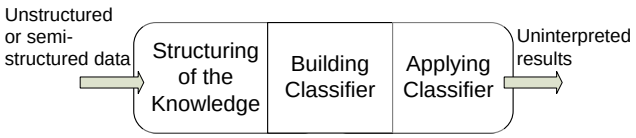

Fig. 12. The main tasks in classification process for unstructured or semi-structured input data

Different parts of this process have various domain dependencies with respect to implementation and reuse.

- Structuring of the Knowledge

Looking from the system's development viewpoint, this is a domain dependent task. The necessary techniques and methods for processing of data are hard to establish without the knowledge of data representation forms in a particular problem domain. Data can be held in structures which require specific extraction and preparation of attributes.

- Building Classifier

This is a relatively domain independent stage and can be defined prior to application for a particular problem area. Principles of the classifier forming are well studied and used. However, the choice of a particular learning approach, method and parameters is tightly connected with actual data, since there are no domain independent reasons to favor one classification method over others [3].

- Applying Classifier

Technical aspects of classification may be considered domain independent, but the choice of how to represent the results is affected by the initial data structure and further processing needs (both for systems and humans).

The output of classification gives uninterpreted results which could be passed to some framework for domain specific interpretations or further processing.

One can conclude that the specification of initial data processing can be given only in connection with a particular application domain, while classifier building and most of decision about classifier applying can be made in advance. Domain less dependent tasks can be defined in earlier architecture development stages than the dependent ones.

B. General architecture of the system

The need for interactivity requires this functionality to be represented in system's architecture. The aim of developing interactive approach is not to improve one certain learning algorithm. Instead it is necessary to develop an extension for those algorithms which lack mechanism for dealing with unclassified instances or where this mechanism can be replaced. This approach affects the way how the classifier is applied to new instances, not the way it learns and makes the predictive model.

The amount of necessary changes in the classification part (in comparison to "standard" non-interactive approach) depends on particular learning scheme and its implementation. If the information about unclassified instances is achievable after the attempt to classify them, uncovered instance handling can be added as an external supplement without modifying the initial classification process. Otherwise, the new instance classification procedure should be extended with the possibility to trace unclassified instances. Figure 13 shows how the interactivity is implemented into the general model of the classification process.

Blocks with solid line are "standard" elements of the classification system, e.g., similar to the one in Figure 6 (in Section II.*B)*. Blocks and arrows with interrupted lines are introduced to ensure interactivity with a human expert in order to assign a class value for unclassified instances. This includes the further mentioned functions.

1. Capturing unclassified instance(s) which were not covered by any rule in the classifier applying stage.
2. Forwarding these instances and additional information to the human.
3. Receiving and processing the human decision.
4. Using the human-provided knowledge to update the learning examples.

The fourth step –updating of the classifier – is an issue that is discussed in more details in the recent work [4].

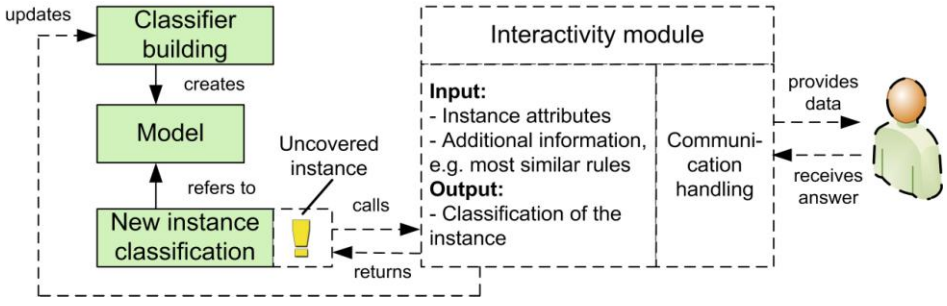

Fig. 13. Inclusion of interactivity in the general classification model

 ISBN: 978-1-61208-232-5

C. *Modules of the system*

For the proposed interactive classification system a modular architecture is chosen. Such architecture is chosen because the modules are relatively independent from each other and can be changed and replaced without affecting other parts of the system which would be not the case if an integrated architecture was applied. Since there are some domain dependent parts in the intended architecture, it is more suitable to use modules where some of them can be static while others change for each application area.

Each module has its own purpose and tasks. One or more modules are involved in performing specific functions. Table I describes each module in details, explaining its functionality and connectivity with other modules.

In practice modules communicate not only directly; examples to learn from and induced rules are stored in separate data bases, and particular module functions are activated by a human. Figure 14 shows physical data flows and initiations of processes in the system, including the user who is actually a part of the interactive classification system.

Figure 14 shows actions typically performed in the interactive classification system, avoiding details of inner processes within modules. The user passes the learning data to the Data processing module through the user interface requesting data preparation for further processing. Prepared data is saved in the Examples storage and the response about the achieved results is presented to the user. When the user initiates creation of the classifier, the Classifier building module uses data from the Example base and infers model to be stored in the Rule base, also representing the results to the user. To assign classification to a new instance or a set of instances, user invokes the Classifier applying module. If the classification can be made by rules in the Rule base, the user receives classification results as a response. If there is an

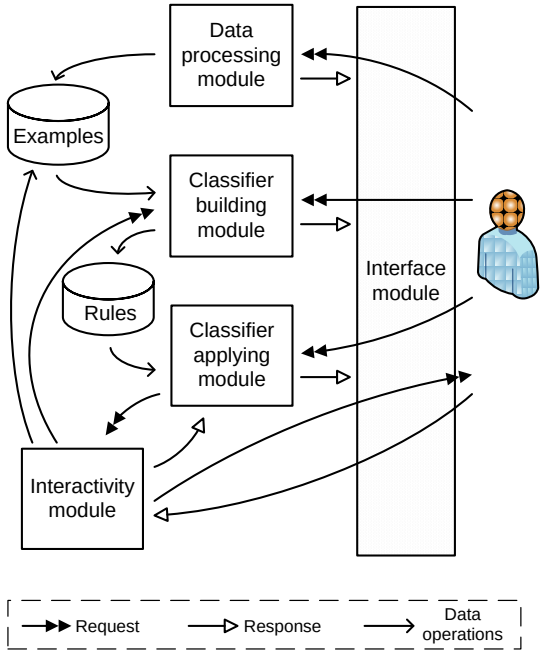

Fig. 14. Functioning of the interactive classification system

instance or instances which cannot be classified, the Classifier applying module sends a request to the Interactivity module to handle the situation. The Interactivity module asks for an expert classification of the instance through interface; this is the situation when a request for a response is being sent from the system to the user, not vice versa. After receiving the user feedback, the Interactivity module gives a response to the Classifier applying module which shows the classification results to the user as previously. The Interactivity module also updates the Example base with a new training example that was built from the unclassified instance and the user-given classification to it. Afterwards the Interactivity module sends a request to the Classifier building module to start a new learning cycle and update the Rule base. However, this is not the only scenario possible for the system's usage.

IV. Architecture of the Interactive Classification System for Study Course Comparison

The need for a specific type of a classification system arises from a problem domain. As described in the related work (Section II), the statement of the problem, analysis of the domain specific factors and application identification are basis of system's development. All further design decisions should be based on actual necessities, of course, taking into account technical capabilities. The area for which the interactive classification system is to be specified is the curriculum management.

Almost every model from section II.*A* could be applied to describe the design of the interactive classification system. However, design steps will be defined with respect to the procedure of Bielawski and Lewand [20]. This description framework is preferred most due to its simplicity and general concordance with the system to be designed. It is an appropriate framework to explain the decisions made during the design stages. In table II, the main design steps are analyzed.

TABLE I

Modules of the Interactive Classification System

Data processing module
Provides exchange of data representation formats. - Ensures the user with the possibility to input learning data in different layouts and helps the user with data structuring. - Ensures the user with the possibility to view learning data and classification rules in different representation formats. - Ensures data transformation for inner processes within and between modules. Direct connection with other modules: - Interface module
Classifier building module
Produces a classifier or a model for the given learning data set. The classifier in internal structures is represented as an application-specific model. If-Then rules can be extracted from this format (if the representation form of the learning algorithm itself produces rules). This module is based on already implemented learning schemes. Direct connection with other modules: - Interface module
Classifier applying module
Applies the given classifier to the provided instances, finds classification and

calculates statistics. This module is based on the already implemented learning schemes which are extended with the ability to intercept instances that are not covered by any rule from the classifier. In this case the interactivity module is called. Direct connection with other modules: - Interface module - Interactivity module
Interactivity module
Ensures communication handling with a human. Closely tied to the classifier applying module. - Represents an unclassified instance and additional information to the human expert as well as receives the answer. Additional information about the instance is, e.g., most similar rules. - Initiates classifier updates after receiving a human's response. - Ensures handling human requests for classifier representation in form of rules. Direct connection with other modules: - Interface module - Classifier building module
Interface module
Ensures human-friendly communication between the system and its user. - Represents data. - Transmits predefined human requests and inputs to other modules of the system. Direct connection with other modules: - All system's modules

TABLE II

DESIGN STEPS OF THE INTERACTIVE CLASSIFICATION SYSTEM

1. Identification of the problem
Globalization and student mobility have led to the need for curricula and study course comparison. This comparison is necessary in order to make sure that learning curricula in a foreign institution still matches the requirements of its home curriculum. Another important area where curricula are to be compared is curriculum development. However, comparison of curricula is based on the compatibility analysis between individual courses. A course comparison is a very time consuming process if performed only manually. The main domain characteristics which play a significant role in design decisions are the following [6]. **Understanding decision making steps is important for a human**. This condition defines the use of the decision tree or rule generating algorithms among all ML methods because of their explanatory power. **Small initial learning base.** This condition causes suspicion of inducing an incomplete classifier. Therefore, an interactive classifier would be useful. **Many classes with similar probability to appear.** As a curriculum usually consists of ten to fifty different study courses and there is no ground for preferring one course over the others, a default rule for assigning a class to unclassified instances is not a proper approach. **Multi-label class membership.** In the case of course classification a certain course can be similar to several other courses; therefore, an assignment of more than one class is possible.
2. Knowledge extraction and representation
To compare different study courses, one needs to define course features that can be used for comparison. The study course is an issue that does not naturally possess well-defined attributes relevant for the comparison of course contents. Attributes used to describe study courses in the classification system should not only be representative but also available. It is not always that education providers and trainers give a detailed description of course contents [33]. However, learning outcomes usually are well described; therefore, they can be used as a means for study course compatibility analysis. Besides learning outcomes other accessible attributes can be involved in classification, namely, study level, number of credit points for the course, etc. The comparison of learning outcomes has to be unified since the verbal description of learning outcomes may vary for different educational institutions. For mediation of learning outcomes European e-Competence Framework could be used since it is European-wide framework for ICT competences.
3. Choice of tools
In this case tools mean not only software tools but also the very learning algorithms used to induce the classifier. A domain with natural multi-label class memberships, like course comparison, requires appropriate learning methods. To save time and efforts for implementing basic learning algorithms, already prepared tools and libraries could be used. It the case with multi-label classification needs there are not too many tools to choose from. *Mulan* library for multi-label classification is chosen because (1) it is based on *Weka* tool which implements many classification algorithms for experimenting, and (2) it is extendable (that is important for dealing with unclassified instances, adding human-friendly interface, and introducing interactivity in the classification system). Both *Mulan* library and *Weka* software are written in Java which consequently leads to implementing the whole classification system in Java.
4. Prototyping and development
Finding the best classification algorithm, tuning parameters, verifying chosen features can be done most powerful by experimenting. Implementing prototype helps to pre-evaluate system's performance and decide about architectural details.
5. Testing and maintenance
The testing stage is meant to evaluate different parameters of the classifier and other parts of the system, e.g., ease of use for a human. During execution of the system it should be capable of classifying new instances as well as communicating with a human and updating the classifier with the knowledge achieved from interaction with the human expert.

The prototype of the interactive classification system has been developed and applied in the domain of study course comparison as well as on Medical data set *(from Computational Medicine Center's 2007 Medical Natural Language Processing Challenge)*. More detailed results of the experiments with proposed interactive classification system are published in [34]. It is proved that applying the interactive approach can reduce the number of misclassified instances, especially, when the initial classifier performs poor. However, subsequent research includes comparing the achieved results with other approaches used in course comparison and applying the interactive classification for other domains.

V. CONCLUSIONS AND FUTURE WORK

This paper proposed the architecture of an interactive inductive learning-based classification system that in uncertain conditions could ask a human for decision and improve the knowledge base with the rule derived from this human-made decision. The architecture of the system is specified for application in a particular problem domain which, in this case, is the university study course comparison. The design steps of an interactive classification system lead to the particular design and implementation decisions, e.g., modular architecture, use of the *Mulan* library for implementing multi-label classification algorithms, etc.

Research on classification systems caused creation of several taxonomies. Firstly, terms "classifier" and "classification system" were distinguished, and secondly, classification system's design stages were separated from the system's structure and functioning.

Contributions of the paper are the following:

- Different existing classification systems' architectures are summarized from the viewpoint of design and functioning, complemented with analysis of the common elements in them.

 ISBN: 978-1-61208-232-5

- General scheme with main stages of the classification process for domains with unstructured or semi-structured input data is given, providing also separation of domain dependent and independent parts in a system's architecture.

- The general architecture of the interactive classification system is provided highlighting the aspects where interactivity makes difference from the "standard" classification approaches.

- Modules of the interactive classification system, their main properties and interrelations are defined. The modules are: Data processing module, Classifier building module, Classifier applying module, Interactivity module, and Interface module.

- For the particular case – study course comparison – one architecture of system's design is applied to describe decisions made in the development process.

Future works include further refinement of the modules, developing, prototyping, and experimenting with the system.

ACKNOWLEDGEMENTS

This work has been supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University".

REFERENCES

[1] K. J. Cios and L. A. Kurgan, "Hybrid Inductive Machine Learning: An Overview of CLIP Algorithms.," in *New Learning Paradigms in Soft Computing*. vol. 84, ed Heidelberg, Germany: Physica-Verlag GmbH, 2002.

[2] R. Coletta, *et al.*, "WebSmatch: a platform for data and metadata integration," DataRing Project meeting June 20 2011 Montpellier

[3] I. Birzniece, "From Inductive Learning towards Interactive Inductive Learning," *Scientific Journal of Riga Technical University. Computer Sciences. - Applied Computer Systems* vol. 41, pp. 106-112, 2010.

[4] I. Birzniece, "Interactive Inductive Learning System: The Proposal," in *Proceedings of the Ninth International Baltic Conference Baltic DB&IS 2010*, Latvia, Riga, 2010, pp. 245 -260.

[5] I. Birzniece, "Interactive Inductive Learning Based Study Course Comparison," in *Rethinking Education in the Knowledge Society*, Switzerland, Ascona, 2011, pp. 339 – 347.

[6] I. Birzniece, "Interactive Inductive Learning System," in *Selected papers from the DB&IS 2010*, Latvia, Riga, 2010, pp. 380-393.

[7] I. Birzniece and M. Kirikova, "Interactive Inductive Learning: Application in Domain of Education," *Scientific Journal of Riga Technical University. Computer Sciences. - Applied Computer Systems*, vol. 47, pp. 57-64, 2011.

[8] I. Birzniece and P. Rudzajs, "Machine Learning Based Study Course Comparison," in *Proceedings of the IADIS International Conference on Intelligent Systems and Agents 2011 (ISA 2011)*, 2011, pp. 107-111.

[9] R. O. Duda, *et al.*, *Pattern Classification*, 2nd ed.: Wiley - Interscience, 2001.

[10] S. Theodoris and K. Koutrumbas, *Pattern Recognition*, 3rd ed.: Elsevier, 2006.

[11] F. Verdenius and M. W. v. Someren. (1997) Applications of inductive learning techniques: a survey in the Netherlands. *AI Communications*. 3 - 20.

[12] T. Mitchell, *Machine Learning*: McGraw Hill, 1997.

[13] W. Fritz. (2010, September 12). *Intelligent Systems and Their Societies*. Available: http://intelligent-systems.com.ar/intsyst/glossary.htm

[14] D. C. Brown, "Intelligent Computer-Aided Design," in *Encyclopedia of Computer Science and Technology*, J.G.Williams and K.Sochats, Eds., ed, 1998

[15] J. S. Gero, "Design Prototypes: A Knowledge Representation Schema for Design," *AI Magazine* vol. 11, pp. 26 - 36, 1990.

[16] A. Bahrami, "Routine design with information content and fuzzy quality function deployment," *Journal of Intelligent Manufacturing*, vol. 5, pp. 203 - 210, 1994.

[17] V. Cherkassky and F. Mulier, *Learning from Data: Conceptrs, Theory, and Methods*, 2nd ed.: John Wiley & Sons, 2007.

[18] S. M. Dowdy and S. Wearden, *Statistics for research*, 2nd ed. ed. New York: Wiley 1991.

[19] DTI, *Neural Computing*, "Department of Trade and Industry" ed.: Learning Solutions, 1994.

[20] L. Bielawski and R. Lewand, *Intellignet Systems Design: Integrating Expert Systems, Hypermedia, and Database Technologies*: John Wiley & Sons, 1991.

[21] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, 2nd ed.: Elsevier, 2005.

[22] B. Settles, "Curious Machines: Active Learning with Structured Instances," University of Wisconsin-Madison, 2008.

[23] J. Han and N. Cercone, "Interactive Construction of Classification Rules," presented at the Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, 2002.

[24] P. Compton and R. Jansen, "A philosophical basis for knowledge acquisition," *Knowledge Acquisition*, vol. 2, pp. 241-258, 1990.

[25] B. Settles, "Active Learning Literature Survey," University of Wisconsin–Madison2010.

[26] M. Okabe and S. Yamada, "Interactive Web Page Retrieval," in *Active Mining: New Directions of Data Mining*, ed Amsterdam OS Press, 2002, pp. 31-40.

[27] R. C. Tanumara, *et al.*, "Learning Human-Like Color Categorization through Interaction," *International Journal of Computational Intelligence*, vol. 4, pp. 338-345, 2007.

[28] W. Buntine and D. Stirling, "Interactive Induction," in *Machine Intelligence: Towards An Automated Logic of Human Thought*. vol. 12, J. E. Hayes, *et al.*, Eds., ed New York Clarendon Press, 1991.

[29] M. Hadjimichael and A. Wasilevska, "Interactive Inductive Learning," *International Journal of Man-Machine Studies*, pp. 147-167, 1993.

[30] M. L. Wong and K. S. Laung, *Data Mining Using Grammar-Based Genetic Programming and Applications*. USA: Kluwer Academic Publishers, 2000.

[31] X. Li, *et al.*, "Learning in an Ambient Intelligent World: Enabling Technologies and Practices," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, pp. 910-924, 2009.

[32] T. P. Minka and R. W. Picard, "Interactive Learning with a „Society of Models"," in *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition*, 1996, pp. 447-452.

[33] I. Birzniece and M. Kirikova, "Interactive Inductive Learning Service for Indirect Analysis of Study Subject Compatibility," in *BeneLearn* Belgium, Leuven, 2010, pp. 1-6.

[34] I. Birzniece, "Machine Learning Approach for Study Course Comparison," in *International Conference on Machine Learning and Data Mining (MLDM 2012)*, Berlin, Germany, 2012, pp. 1-13.

 ISBN: 978-1-61208-232-5