A Model-based Approach to Information Retrieval Systems Development

João Ferreira[1], Alberto Silva[2], and José Delgado[3]

[1] *ISEL*, [2] *INESC-ID*, [2,3]*IST*,

[1]jferreira@deetc.isel.ipl.pt

[2]alberto.silva@acm.org

[3]Jose.Delgado@tagus.ist.utl.pt

ABSTRACT

We propose a novel model-based approach (MDA) for the design and creation of Information Retrieval (IR) systems. This is based on a specific language that provides common notation and concepts and a collaborative modular environment for the design of IR systems. The language is a UML profile, involving several stereotypes for the IR area. From this profile we derive standard libraries of modules that can be used in the development of IR systems. Through appropriate templates, we transform models into software code according different programming language and different IR platforms.

KEY WORDS

MDA, UML, IR, Modeling, Profile, Template.

1. Introduction

Information Retrieval (IR) has been developed during the last four decades based on algorithms and methods. The development of IR systems is a complex process, usually performed by groups of IR researchers or commercial companies. Usually the creation of IR systems is not a collaborative effort among groups in spite of the existence of commons modules among the IR systems. These efforts highlight a lack of availability of specific (Personalized) IR systems, because it is a complex task. At the beginning of this decade, some modular IR platforms have been done [1,2,3] but these modules are still too large to allow flexibility. On the other hand, if we have smaller modules, the new appearing problem would be how to assemble them together, like a LEGO construction without instructions. To avoid these problems, some IR systems [1,2] give some flexibility through the availability of several options in a predefined API. This scenario leads us to a new approach regarding IR systems' construction based on models, that integrates the best practices and fundaments around the Model Driven Architecture (MDA) paradigm and specification of requirements, such as modularization, separation of concerns, reutilization, use-case driven, model-to-model and model-to-code transformations [4,5,6]. These aspects are implemented through the methodology proposed in this paper, providing a roadmap where designers can follow as well as model-to-model transformation templates in order to accelerate their system development tasks. The main steps to build IR systems are illustrated in Figure 1: (1) IRML, a language based on UML extension mechanics focused on the the IR area; (2) IR-Models, derived from IRML, that provide standard libraries for the IR-System; (3) a conceptual IR system, constructed through the models stored in a database; (4) conceptual systems are transformed in XML; (5) predefined templates created by an inverse engineering process are used to create code modules from the models chosen in XML format; (6) a specific platform will integrate the different modules to generate the IR-system.

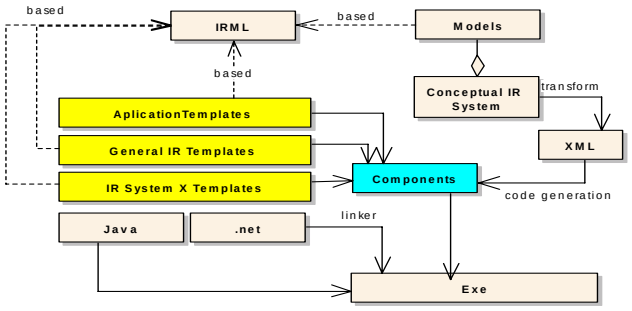

Fig. 1. Top-vision of the model-based approach for IR systems development.

The benefits of this process are relevant, namely: (1) to facilitate the IR-Systems building process (increasing customization possibilities); (2) this standard, if well accepted, proposes a collaborative environment within the IR Community, originating and supporting faster development; (3) automatic construction of IR-systems from models (higher level view) facilitates the process, easily allowing changes.

This paper is organized into four sections: (1) introduction; (2) methodology, which is divided into two sections: IRML, language proposed based on UML and abstract models; (3) case study, MyTv program guide and personalized television; and (4) conclusions.

This approach has been applied in software engineering areas [7,8,9,10], and the most related works were on personalization of websites using modeling methods [11,12,13,14], due to the diversity of personalization

policies over the development cycle of websites. We intend to go further steps ahead integrating on this process the creation of IR systems. This approach makes sense due to the use of common modules\parts in different IR systems and also due the diversity of retrieval approaches. There is a big number of different IR systems [15,16,17,18] always constructed from zero. In spite all the diversity of systems, they relay on the same principals, use statistic proprieties of documents, creation of documents' representatives with smaller dimensions, interfaces that users allow to perform queries, expand, feedback, matching and optimization methods.

2. Methodology

We propose a new approach or methodology for the creation of IR systems inspired on a set of best practices or principles: it is based on high-level models or specifications; it is component-based architecture centric; it is based on generative programming techniques. This approach follows in essence the MDA philosophy with some specific characteristics. We propose a repository that keeps related information, such as models, applications, software architectures, generated artifacts and even information concerning the software process itself (e.g., generation steps, tests and integration milestones). Figure 2 overviews the methodology, in particular the main actors and corresponding tasks. Generically, this methodology receives system requirements (e.g., functional, non-functional and development requirements) as its main input, and produces a set of artifacts (e.g., source code, configuration scripts or data scripts) as its main output.

The tasks performed by the software architect are critical to the process of creation of an IR system. The architect is responsible by the following tasks: (1) to define a suitable and easy-to-use UML profile (IRML); (2) based on IRML, to define abstract models; (3) to define and to select an IR infra-structure to support the IR system; (4) to develop templates: (i) to model transformation features, such as "Model2Model Transformation Templates", (ii) to produce new models, (iii) to develop model-to-code template features, such as "Model2Code Transformation Templates", and (iv) to produce software and documentation artifacts from models, using generative programming techniques.

Starting from system requirements (created for instance from meetings, interviews, JAD sessions among designers, clients, end-users and other stakeholders) [19]. The Requisites Engineer collects objectives and identifies the motivation to create the system. The Designer is responsible for the design of the IR system through available models, producing an integrated set of models (the "Design System" task). Still, the designers can apply model transformations automatically according the "Model2Model Transformation Templates" developed previously by architects. This task can be useful in certain situations in order to simplify or to accelerate the design task. The correctness and quality of the models produced are essential to obtain good results in the subsequent tasks. After the design intervention, programmers apply model-to-code transformations, which means to apply generative code techniques to models, based on templates provided by the architects. Because it is not possible to capture and to design all the system requirements, programmer intervention is still required The Developer creates systems via the MDA approach proposed (see Figure 3). Conceptual models are transformed into XMI (T1) and simplified in XML (T2). From these XML models, and with appropriate templates created in an inverse engineering process, we generate the modules of the IR system, which are linked the Eclipse Platform, a well-known and stable platform with widespread support in the Java community. Although commonly known to software developer community as an open-source IDE (Integrated Development Environment) for Java software development, Eclipse should be best described as "an open universal framework for building developer tools" [20]. Consequently, programmers are involved to produce specific components, typically helper source code, such as facades, adapters, controllers and business logic.

Finally, the intervention of testers and integrators is necessary to prepare and to perform different tests in order to guarantee the system quality. These activities are suggested in the Figure 2 by the "Prepares, Performs Tests and Integrates the IR System" tasks.

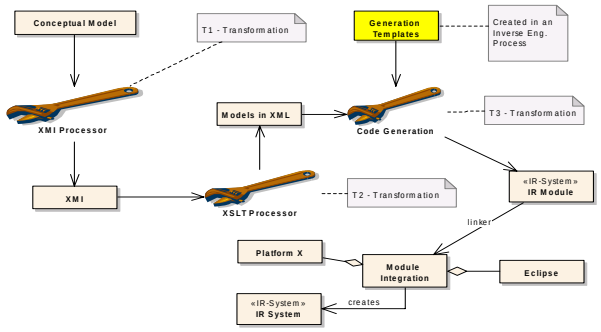

Fig. 3. Code generation from models with appropriate templates. Description of these transformation and examples of templates can be found at [28].

2.1 IRML

UML 2 presents a new definition based on MDA following the following main principles: modularity, layer division and extensibility. This new version goes in line with the main objective of an IR-Language. This subject can be explored at [21].

Based on UML we propose a meta-model for IR (IRML), based on new a stereotype specific for IR [22]. IRML is a set of coherent UML extensions that allow a high-level, visual modeling way to design interactive systems. To simplify the design of IR-Systems we propose three views. These views act like the different views in an architectural project of a house and simplify the process

by dividing the problem in smaller pieces. The views are chosen between a compromise of simplicity (more views) and complexity (less views). From the construction of several systems (more than 100) we observed that these three views simplify the process of construction of IR-System and at the same time are enough to describe the problem.

The **IR-UseCaseView** defines IR-Actors and their actions on the system. In this view the external relations of the system and the main objectives are defined. The main actors are the following: (1) **IR-User**, responsible for defining the information needs (query or user profile), receives the system results (list of document ordered by relevance criteria) and gives feedback to the results received; (2) **IR-Producer**, creates documents to be stored; (3) **IR-Authority**, creates and maintains the knowledge space, creates test collections and defines topics with relevant documents from associated test collections. Communities of users that are automatically identified by the IR-System are validated through this entity; (4) **IR-Investigator**, uses the testing platform (IR-System) to validate IR-Algorithms or IR-Processes created. Chooses the test environment, and collection as well as the topics and evaluates the results.

IR-InformationView defines the system data input and output. In this view, the data flow is shown using class diagrams: (1) **IR-Document**, information produced by the author and to be recovered by the user. The main associated problems are: available in different formats, Human language is subjective, depending on the context, some authors give wrong information on purpose in order to be present on the top positions of retrieval results from search engines; (2) **IR-Collection**, group of documents stored. The largest collection is the Web; (3) **IR-Index**, the output of the IR-IndexProcess, and the input of IR-MatchingProcess; (4) **IR-InformationNeeds** is an input of IR-MatchingProcess, composed of a list of terms (with assigned weights) and divided in two main classes: **IR-UserProfile**, stable user information needs, and **IR-Query**, which represents momentaneous information needs; (5) **IR-KnowledgeSpace**, represents a specifically organized (by an authority) information space, divided in Classification Systems (generic or area specific), User Community and Thesaurus; (6) **IR-System**, integrated process resources that transform input data in a list of documents organized by relevance. The system can be divided in two main systems: Search and Information Filtering .

IR-ProcessView, in which attributes and sequence of actions to transform input into output are defined according to the proposed objectives. The main processes are: (1) **IR-Process** is a concept that involves data transformation by a pre-defined sequence of activities and is divided in four main processes. Additional processes such as translation are identified as IR-Processes; (2) **IR-IndexProcess**, transforms collection documents into a small representative with the following the process: convert to a pre-defined format; identify fields: words, phrases, headers, URL, etc; remove stop words; stemming; word count; calculate predefined statistical measures; store. In this list of processes, we can introduce a dictionary and a translation process; (3) **IR-MatchingProcess**, compares the document's representatives with user's information needs according to an IR-Algorithm. For a detailed analysis, see [23]; (4) **IR-OptimizationProcess**, whose main objective is to improve the output of the system and is divided in two main areas: work on input data, mainly feedback process (automatic, manual or a mixture of both); work on output data, combination (also called fusion) of results. The main approaches are: Similarity Merge (combination of relevant measures) or Weighted Rank Sum (WRS), that uses rank-based scores (e.g. 1/rank) instead of document scores. Several forms of WRS can be explored. A detailed discussion can be found on [23]; (5) **IR-EstimationProcess**, responsible for the test collection, classification algorithms parameters' estimation, language models or other IR-algorithms.

2.2 IR Abstract Models

IR-DataView Models

Queries can be performed based on three main formats: (1) QueryAdhoc, in which the user chooses free terms; (2) QueryClassifiedSystem, that lets the user choose categories and terms from a Classified System; (3) Topics (TREC) use previously built queries for evaluation purposes.

The UserProfile has more attributes (email, password and periodicity could have a negative profile), that can be: (1) UserProfile+free: the user chooses the terms of his stable information needs; (2) UserProfile+Community: the user chooses one community that he belongs to and the central community profile is used as user profile (each user can choose more than one community); (3) UserProfile+Colaborative: the user rates the subjects and the system identifies users with similar ratings providing a collaborative environment; (4) UserProfile+Community: the user chooses categories and terms from a Classified System; (5) UserProfile-, negative profile: the user defines subjects which he is definitely not interested in.

Two types of IR-Index are proposed: (1) General, storing raw statistical data without any manipulation (it is a flexible but slow index that can be used by any IR-MatchingProcess); (2) Specific IR-Index, where data is previously manipulated according to the chosen IR-MacthingProcess.

As IR-Knowledge Space, we have: (1) Classified Systems which can be area specific (like ACM, MCS) or generic (like CDU, Yahoo), with abstract classes; (2) the Community is automatic created but the released for the classified space is performed by IR-Authority. The system identifies possible communities by clustering the users' profile and an IR-Authority checks if it makes sense. Community is identified by a central user profile; (3) Thesaurus, whose main use is as a dictionary to avoid

spelling errors and help the process of phrase identification.

IR-Result is the output of IR-Matching process and is a list of documents organized by a relevance measure. The main attributes are: filepath, summary and relevance measure.

IR-Process View: We will now look on IR-Process for the two main services: Retrieval (Search) and filtering of Information.

IR-Retrieval-System

The main processes of a IR-Retrieval system are shown on Figure 4: (1) IR-IndexProcess creates and stores the representative of documents collection. The process is flexible, which means that you can add or remove modules. We can add (to this process) translation and speller checking through standard available dictionaries; (2) Estimation process is available to Language Models (IR-MatchingProcess) based on a test collection. (3) IR-MatchingProcess compares the documents' representatives with the user profile, and can use feedback techniques; (4) Combination, combines results through measures or rank order formulas to try to improve results.

Filtering System's main process are: (1) translator; (2) Event, which is responsible to trigger the notification process based on periodicity; (3) IR-IndexProcess; (4) Feedback, important for user profile optimization; (5) IR-MatchingProcess based on cosine measure and correlation to find communities.

3 Case Study

We have been developing several examples to show the advantages to develop IR systems by using the proposed methodology [24]: (1) MyEnterpriseNews, a system to find relevant news about a subject; (2) MynewsPaper, a personalized news paper; (3) MyDocument, a system that organizes documents in a department or company using a local hierarchical knowledge space. In all these systems have common modules and we, use the same user interfaces modules. In this paper we concentrate on a filtering system of TV programs (MyTV) applied to two worlds [25], Text and Image, using common modules. We show conceptual models of these systems and we intend to be able to show a functional working system soon, developed completely with this approach. To simplify the initial approach we started developing templates to big IR modules (e.g. OpenFts [26,27]) and well know IR systems [1,2], latter we intend to improve by creating templates for specific IR tasks and consequently increase modularity.

MyTV: Program Guide

The main objective is to create a personalized notification system about TV programs (in this case, broadcasted by the Portuguese cable operator TvCabo <www.tvcabo.pt>) to registered users. The User profile is based on a list of programs.

Use case view: **User (*IR-User*)** defines a profile by: (1) choosing free terms; (2) browsing a classified space; (3) choosing user communities. The user can introduce a negative profile, reflecting topics, which he does not have interest in. He receives from the system the relevant programs in a predefined periodicity by email and can give feedback on notifications received and evaluate programs seen in a scale 1 to 5. **TvCabo (*IR-Authority*)**, is responsible for the validation of user communities identified automatically by the system and also for the classified space.

Information View, Figure 5**:** list of programs, classification system, user profile, identified communities and results (recommended programs).

Process View's main process is described on Figure 7, using a standard filtering process with an additional translator process implemented with ***Wordtrans*** **<wordtrans.sourceforge.net>,** translating information in 6 languages.

MyTV: Personalized Television

The quantity of available channels is increasing, which raises the problem of users missing interesting programs. Users can minimize this problem by zapping on channels available, or reading the programming guides in advance, but this imposes effort on the user. To avoid this, we propose a system that is able to send and alert (based on a user profile) on a small window regarding interesting programs broadcasted during the period that the television is switched on. These systems have the challenge of identifying video programs automatically and to match them against the user profile. This is a new system that we will create by using parts of an existing system, demonstrating the potential of this new approach.

Use case view is the same, but TvCabo (IR-Authority) has one additional task that is the creation of a test collection.

Information view. We have made a change on the document collection: instead of a text index, we now have low-level properties of video, see Figure 6. Low level properties of video are identified by the algorithms: (1) shot boundary detection, (2) GofGopColor, descriptor; (3) EdgeHistogram, descriptor; (4) MotionActivity, descriptor. From these algorithms, results follow low level characteristics using MPEG-7 in a video segment: (1) Color, GofGopColor, calculates color histogram in a space HSV; (2) EdgeHistogram calculates edges for 16 zones; (3) MotionActivity, from 1 to 5 measures of movement intensity; (4) density of cuts. Sound characteristics will be introduced in future work.

TV Communities (IR-Community) are groups of users with common interests identified by a high-level profile.

Process view: in relation to the previous system, we removed the index process and added three new processes: ConvertorHighLowLevel using an estimation process based on test examples, ConverterLowHighLevel and process to identify low level characteristics of video (described on Figure 7). These converters from high to low level (and vice versa) are run from a test collection of programs by an estimation process. Tuning these processes is a complex task. For example, a football game has green as predominant color, global movements with

direction change, movement in low speed (repetitions). Other programs usually have other characteristics.

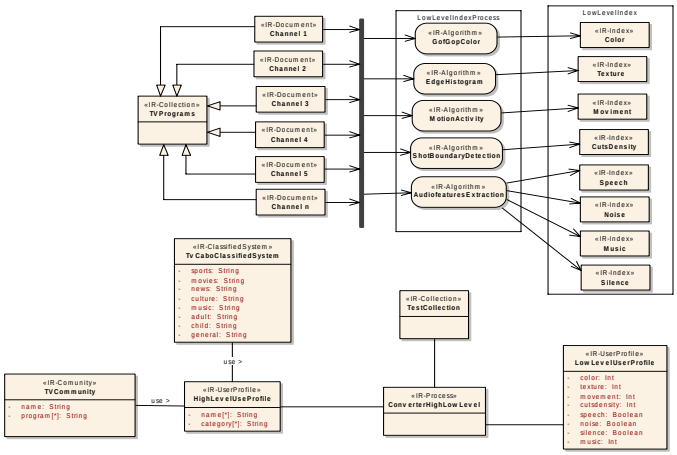

Figure. 6. Information View of MyTV personalized television

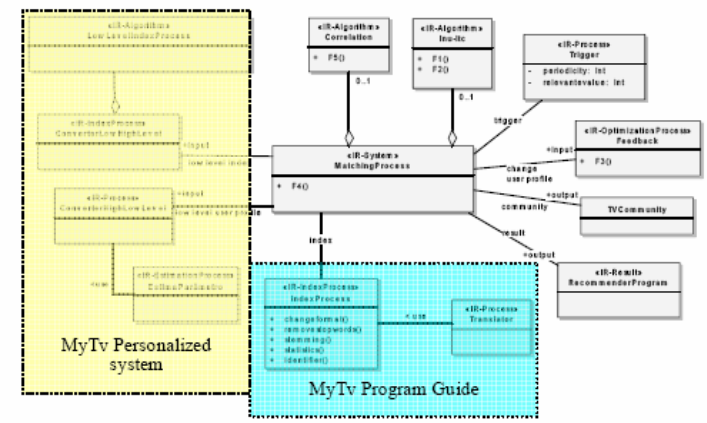

Figure 7. Process View of MyTV program guide and personalized television. Processes without a dashed box belong to both systems. The process view of MyTv program guide is the common processes plus the specific ones (index process and translator process). For a complete description of used algorithms, see [22].

4 Conclusions

We propose a new methodology to develop IR systems using a novel model-based approach as well as generative programming techniques. The vision underlying the methodology is not new by itself. It is effectively a revisitation of existing expectations that in the past did not have so much success: the idea that the building of information systems would be performed almost automatically starting from high-level and platform independent specifications. Meaning that, in the end, classic tasks like programming would be performed almost automatically.

Applying the proposed methodology to the development of IR bring us the following benefits: (1) increase the widespread of personalized IR systems, because it would be easier to develop these kind of systems if there are available a certain number of models and involved templates; (2) reuse of models and templates at high-level in order to realize this potential; (3) promote the collaboration among different research groups; and (4) better productivity, which means less time and low cost spent in development.

An interesting point of this approach is the theoretical possibility to produce code in different programming languages, according a set of software architectures and frameworks (in this context, according different IR platforms), as it is suggested by the MDA philosophy [25]. That issue will be evaluated and researched in future work.

References

[1] Lemur <www.lemurproject.org/>

[2] Terrier <ir.dcs.gla.ac.uk/terrier/>

[3] Okapi < http://www.soi.city.ac.uk/~andym/OKAPI-PACK/>

[4] Model Driven Architecture. http://www.omg.org/mda/

[5] OMG. “White Paper on the Profile mechanism”, Version 1.0, OMG Document ad/99-04-07. OMG UML Working Group.

[6] Kotonya, G., Sommerville, I., Requirements Engineering Processes and Techniques, NYork. Jonh Wiley & Sons,98.

[7] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns – Elements of Reusable Object-Oriented Software. Addison Wesley, 1994.

[8] C. Hofmeister, R. Nord, D. Soni. Applied Software Architecture. Addison Wesley, 1999.

[9] The Software Patterns Series. *Addison Wesley*, 1996-2002.

[10] M. Juric, et al. J2EE Design Patterns Applied. *Wrox Press*.

[11] Koch, N., Kraus, A. and Hennicker, R. (2001). The Authoring Process of the UML-based Web.Engineering Approach, *Proceedings of the 1st International Workshop on Web-Oriented Software Technology*.

[12] De Troyer, O. and Leune, C (1998). WSDM: A User-Centered Design Method for Web Sites. *In Computer Networks and ISDN systems Volume 30, Proceedings of the 7th International WWW Conference*, pages 85-94, *Elsev*ier.

[13] Ceri, S., Fraternali, P., Bongio, A., Brambilla, M., Comai, S. and Matera, M.(2002). Designing Data-Intensive Web Applications. *Morgan Kaufmann Publishers Inc.*.

[14] Frasincar, F., Houben, G.-J. and Vdovjak R. (2002). Specification framework for engineering adaptive web applications, *The Eleventh International World Wide Web Conference, Web Engineering Track*.

[15] http://bit.csc.lsu.edu/~kraft/retrieval.html

[16] http://www.dcs.gla.ac.uk/idom/ir_resources/ir_sys/

[17] http://www2.sims.berkeley.edu/resources/collab/

[18] http://www.glue.umd.edu/~dlrg/filter/software.html

[19] Kotonya, G., Sommerville, I., Requirements Engineering Processes and Techniques, *N Y. Jonh Wiley & Sons*, 1998.

[20] Booch, G. (1999); Jacobson, I.; & Rumbaugh, J. The Unified Modeling Language. Reference Manual. *Reading, MA: Addison-Wesley*, 1999.

[21] Ferreira, João; Silva, Alberto; Delgado, José (2006). IRML - Information Retrieval Modeling Language. *Proceedings of Modelling, Simulation and Optimization (MSO 2006), 6th IASTED International Conference*, 11-13 September 2006, Gaborone, Botswana.

[22] Ferreira, João; Silva, Alberto; Delgado, José (2005). A modular platform applicable to all statistical retrieval models, *Proceedings of the ITA05*, 7- 9 of September 2005, in Wrexham, Wales.

[23] Ferreira J. A Methodology to Design Information Retrieval Systems. Ph.D. Thesis, Instituto Superior Tecnico, Lisbon, 2006 (in Portuguese).

[24] Ferreira, João; Jesus, Rui; Abrantes, Arnaldo. MyTv: Sistema Personalizado de Televisão, *JETC 2005*, 17-18 November 2005, Lisbon (in Portuguese).

[25] OMG. "White Paper on the Profile mechanism", Version 1.0, OMG Document ad/99-04-07.

[26] Ferreira, João; Silva, Alberto; Delgado, José (2004). Infraestrutura modular de teste para pesquisa de informação, *IADIS Conferencia Ibero-Americana WWW/Internet 2004* - October 7 - 8, 2004.

[27] Openfts: <openfts.sourceforge.net>

[28] Alberto Rodrigues da Silva, Gonçalo Lemos, Tiago Matias, Marco Costa, The XIS Generative Programming Techniques, *Proceedings of the 27th COMPSAC, IEEE Computer Society* in USA, Dallas, Nov03.

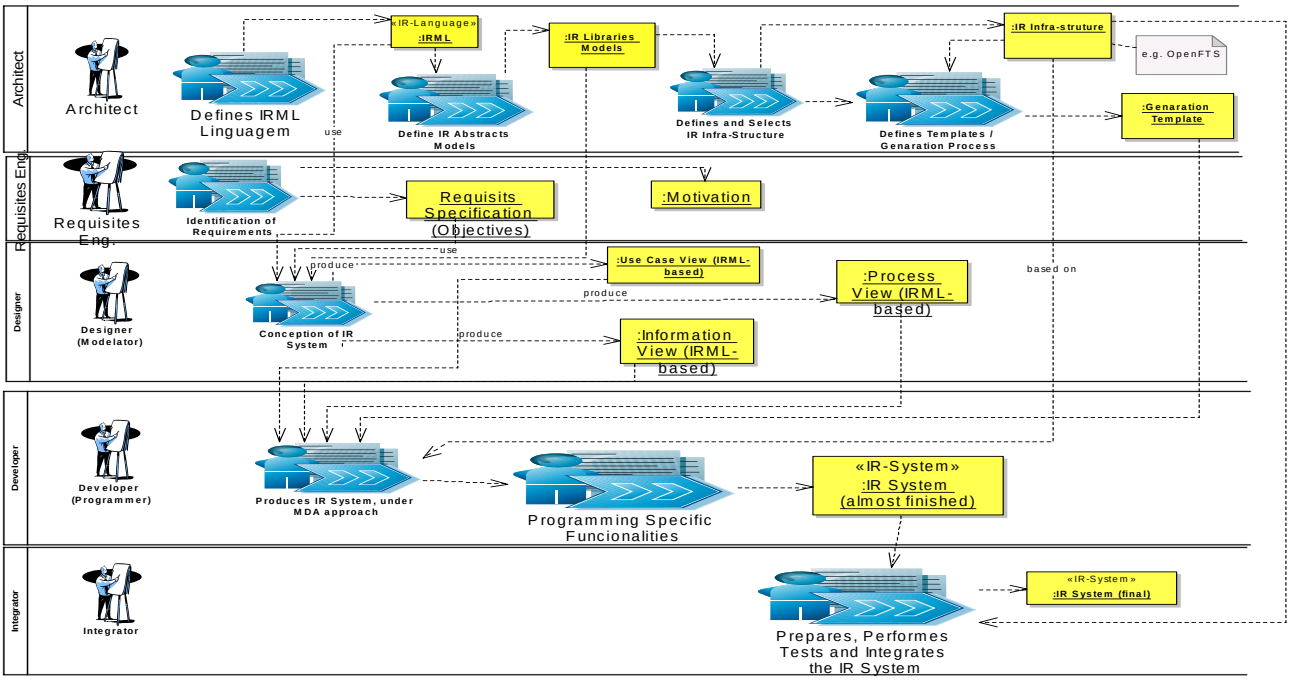

Figure. 2. Methodology for the creation of IR systems

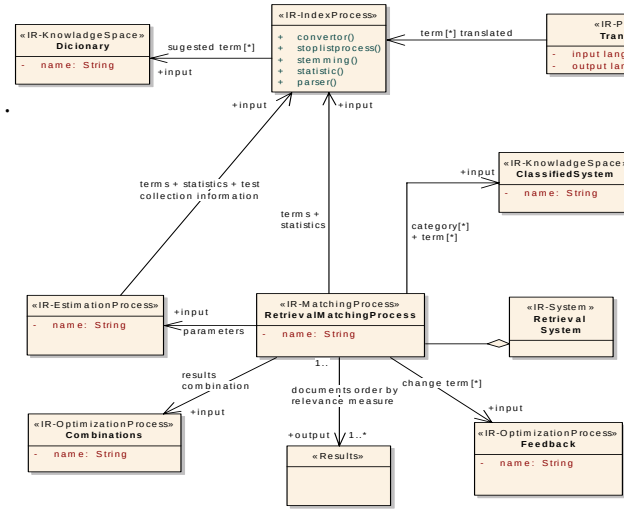

Figure.4. Process View of a generic retrieval system

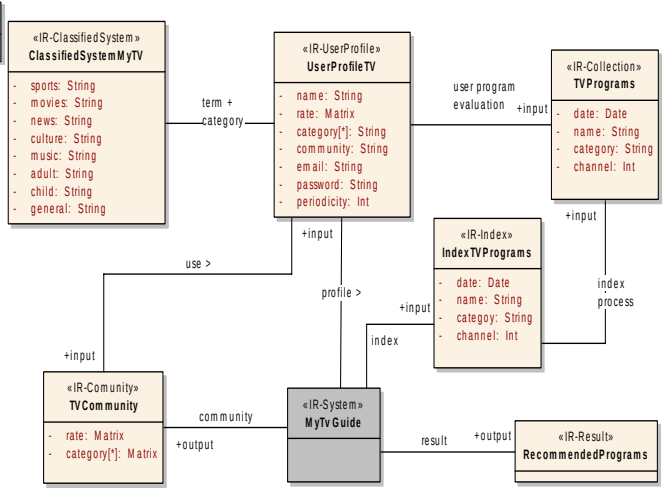

Figure 5. Information View of MyTV Programs Guide