

Technology Selection for Identification Applications

Behnam Jamali, Ed Sharp, Alan Thorne, Peter Cole

Auto-ID Labs White Paper BIZAPP 043



Behnam Jamali
Associate Director
Auto-ID Labs
Adelaide



Peter H. Cole
Research Director
Auto-ID Lab
Adelaide



Alan Thorne
Associate Director
Auto-ID Labs
Cambridge

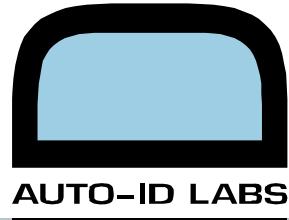
Ed Sharp
Research Fellow
Auto-ID Labs Cambridge

Contact:

Auto-ID Labs Adelaide
The University of Adelaide
Department of Electrical and Electronics

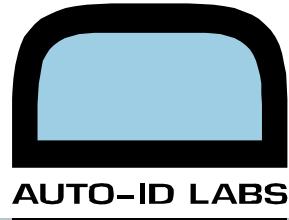
Phone: +61 8 8303 4119
Fax: +61 8 8303 4360

E-Mail: bjamali@ieee.org
Internet: www.autoidlabs.org



Index

Abstract	3
1. Introduction.....	3
2. ID Technology Selection.....	4
2.1. The technology selection process	4
2.2. The selection process	5
3. The Tool	6
3.1. Functionalities	6
3.2. The Interfaces	6
3.3. Methods of operation.....	9
4. Implementation.....	11
4.1. Description of criteria.....	11
5. Technical Details	16
5.1. Software tools	16
5.2. Ruby on Rails.....	16
6. Conclusions.....	21
References	22



Abstract

This paper documents the conversion of a theoretical model for selecting the right identification technology for a given application into a working software application (tool) that assists the user through the selection process.

1. Introduction

From Fortune 100 enterprises to small businesses, in today's business world, item management, identification and tracking have become critical drivers of business processes and decisions that impact organizational growth, risk, and profitability.

It is imperative that businesses more closely align their auto-id infrastructure with their application needs. An ID Application Matching solution (from hereon referred to as the tool) from Auto-ID Labs answer this business challenge, enabling organizations to strategically align business applications and auto-id infrastructure to reduce cost and boost performance, availability, and functionality.

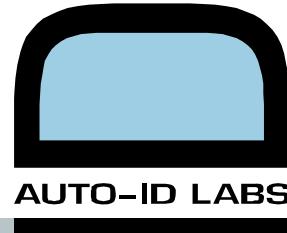
This tool was originally developed as part of a sponsored project by Aero-ID project at Cambridge University (UK). It was designed to assist users in selecting an identification technology appropriate to their needs.

The ID Application Matching solution (the tool) is built upon a common web-based framework of comprehensive data, content, and application services, providing the appropriate infrastructure and management for diverse application requirements. This paper outlines the development of the ID Application Matching.

In this work we consider the characteristics of different ID technologies, examine the complexities of selecting them against user defined criteria and outline the development of a software application to assist in the selection process.

From the users' perspective the tool is split into the following four steps:

1. Answering questions about the application and the environments.
2. Visually refining the values of several criteria to reduce the number of potential technologies to a manageable number.
3. Using the report generated by the tool to investigate the remaining technologies.
4. Entering the results of those investigations back into the tool in order to be able to repeat the selection process with the new values, if a technology has not by that stage been decided upon.



2. ID Technology Selection

This section presents the theoretical basis upon which the tool was developed.

2.1. The technology selection process

Research done at the Auto-ID labs has shown that there are six main forms of automatic identifications in use today. All of these technology elements offer methods for storing some form of identifier in a way that can be read by a machine at a later time. These technology elements are:

1. Barcodes
2. Contact Memory Buttons
3. Magnetic Strips
4. Optical Strips
5. Radio Frequency Identifications (RFID)
6. Smartcards

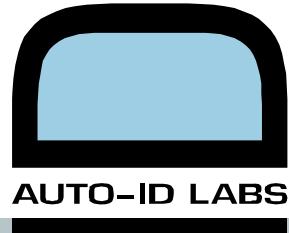
The different ways of capturing characteristics of these technologies and their limitations are studied in work done by Edouard Thisse at the Cambridge Auto-ID Lab.

It has been chosen to classify the performance indicators of different technologies at three levels, as set out below:

Level 1 – Conceptual: This level encompasses all the characteristics of a technology that will always be true for that technology. For example, an RFID system does not require line of sight.

Level 2 – Theoretical: This level encompasses variable performance parameters of the technology, for example read range, read speed, etc.

Level 3 – Practical: This level encompasses further details of the specifications of the items in the market, and contains details of practical performance parameters that might be measured for particular products. For example actual read range measurement of certain type of RFID tag.



For a given application, one would start in step 1 by identifying all the available technologies. These can then be eliminated, in step 2 at the conceptual, theoretical and practical levels, based on failure to meet the application's criteria, set by the user, for acceptable performance parameters.

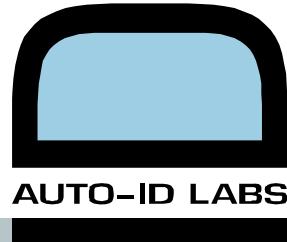
Until this point, the company would be largely relying on manufacturer's specifications. Once the number of technologies has been narrowed down sufficiently, the company can afford to test the remaining technologies and build a business case if the technology is found to be suitable.

2.2. The selection process

In general terms, one can adopt any of the two approaches for selecting an ID technology. The first option is select an ID technology and conduct research on that particular technology and develop knowledge based as how that technology can be used and what its advantages and limitations are. Armed with this knowledge, one can search for potential applications for the technology.

A second approach can be adopted when one has a particular application in mind and needs to find a technology that can meet the constraints of that application. It is this approach that the ID Technology Application Matching tool will be of assistance. As soon as a particular application is decided upon, the user would be able to input constraints of that application and the tool will start to filter out technologies at the conceptual, theoretical and practical levels.

The tool is designed to filter technologies by using set of criteria against which all the available technologies will be judged, so that each technology could be compared against the others. The selection process starts with a complete list of all the technologies for which the tool has information for. As constraints for each criterion are added, the technologies that no longer match the constraints are eliminated from the list and become unavailable.



3. The Tool

This section introduces the functionality of the tool, the user interface and the way a user would interact with the tool.

3.1. Functionalities

As outlined in the previous section, the main function of the tool is to filter out technologies based on the criteria input by the users. A secondary function of the tool is to allow the users to input results of their own experiments on particular technologies, enhancing the tool's default values and relationships when they use the tool subsequently.

Further on the tool can be used as a database to store information about parts, standards, technology vendors and solution providers. Manufacturers can input information about their product into the tool which then would be available to the users during the ID selection process.

Users may also employ the tool as a logbook to keep track of the experiments they have performed on particular technologies and share that information with other parties.

3.2. The Interfaces

The tool contains two interfaces, the questions interface and the criteria interface.

3.2.1. Questions interface

The aim of this interface is to gather as much information as possible about the users' application and environment in which the application is to be used in order to set some initial conditions for filtering out inappropriate technologies.

The Tool takes a series of questions stored in its database and asks the user to answer them. Based on the answer provided to a question, there might be some follow-up questions.

The questions asked by this interface are limited to following answer types:

- Yes/No (check box)
- Some numerical value (input box)



- Single selection from a list of several options (drop-down box)
- Multiple selection from a list of several options (list box)

When there are no more follow-up questions to be asked, the Tool will move on to the criteria filters interface as described in the next section.

3.2.2. Criteria Filters interface

The main feature of this interface, as shown in Figure 1, is two-dimensional display used to represent the technologies' performance on two independent axes. Each axis can be set to display one of the following 14 performance parameters, and the criteria which have been set by the user and against which the performance parameters will be judged and technologies filtered according whether or not they satisfy the criteria.

Data Storage (bits)	Life Expectancy (months)	Operational Temperature (°C)
Read Range (m)	Constant Traverse Speed(m/s)	Extreme Humidity (RH)
Simultaneous Read (number)	Character Set (bits/char)	Extreme Temperature (°C)
Read Time (s)	Authentication (Crypto level)	Physical Area (mm) ²
Read Area (mm) ²	Operational Humidity (RH)	

Each criterion is described in more detail in Section 4.

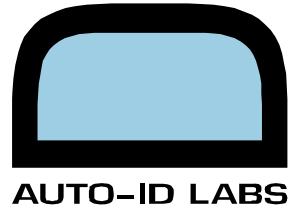
In Figure 1, every technology is represented by an ellipse. Its width and height depends on the range of technological parameters that are found for that technology and the performance parameter being displayed.

The value of the filter for each criterion can be changed visually, by dragging the scrollbars at the left and top of the two dimensional display. The red blanking indicates the technologies that are filtered out by the criterion. There are three types of threshold for the filter:

- More than: the technology's value must be more than a certain value otherwise it is filtered out. For example life expectancy must be more than 12 months.
- Less than: the technology's value must be less than a certain value otherwise it is filtered out. E.g. read time must be less than 0.5 second.
- Between: the technology's value must be within a range of acceptable values. E.g. operational temperature must be between 0° C and 70° C.

The green ellipses are technologies that, provided are not covered by the red blanking, meet the criteria imposed in respect of the two criteria imposed on the display.

In Figure 1, for example, RFID technology meets the criteria for data storage and humidity.



In Figure 1, all the criteria imposed so far are aggregated on the right hand side window. The appearance of true on all lines of that window indicates that RFID has satisfied all the criteria so far. This display would potentially change as other criteria are imposed.

The “Level of detail” scrollbar alters how the data is aggregated. At the lowest level of detail, as illustrated in Figure 1, one ellipse represents each of the six technologies, thus there are six ellipses. At the greatest level of detail, individual technology examples are displayed.

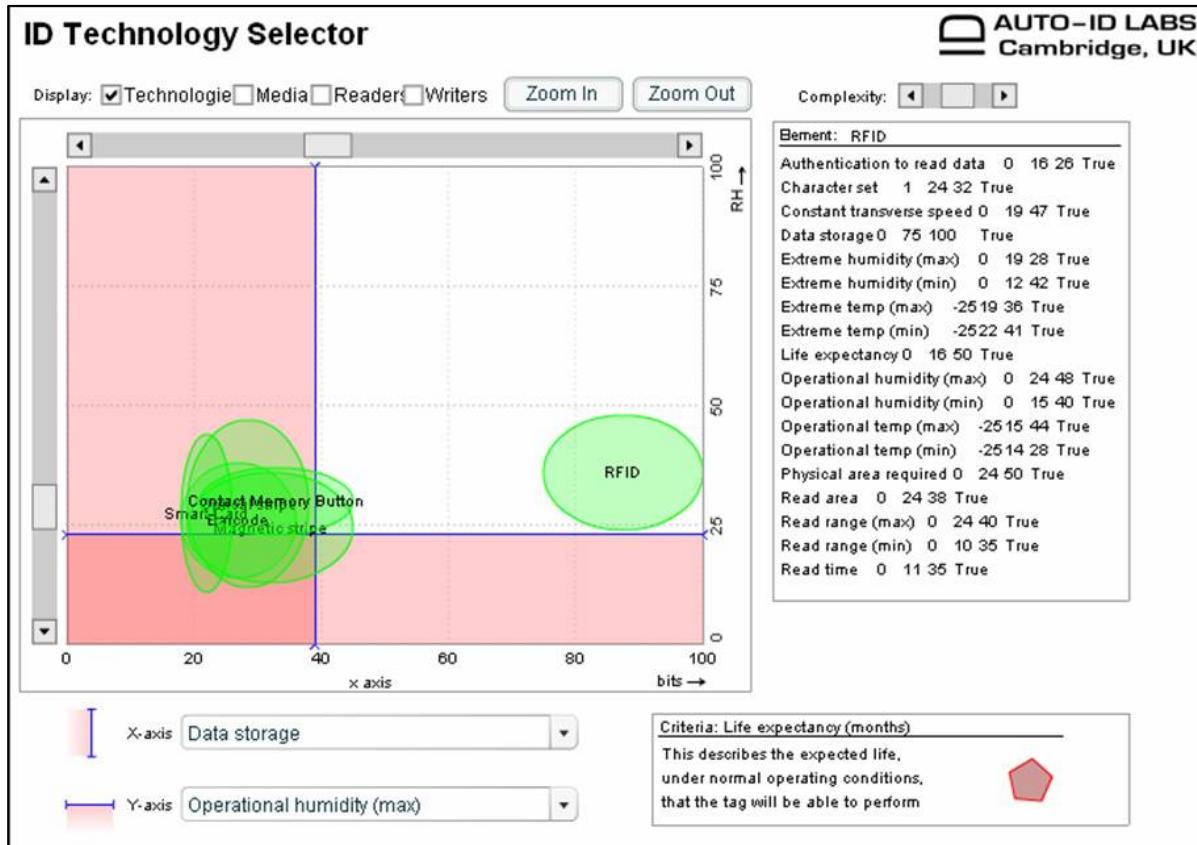


Figure 1 Screen shot of the criteria filters interface



3.3. Methods of operation

The tool can be used by three groups of users, the technology vendors/manufacturers, the tool administrators and the end users. The following sections describe how the tool can be used by these different categories of users.

3.3.1. Technology manufacturers

Manufacturers of media, readers, writers and sub-components (e.g. antennae) will be able to input and maintain their products in the application's database. This information will drive the filtering of technologies based on the criteria for practical level, and possibly the theoretical level.

It will be in the interest of the manufacturer to keep adding their latest products into the database in order that an end-user might choose them. When information is entered by a manufacturer, it will not go directly into the live system, but will be temporarily held to allow a moderator to check the information.

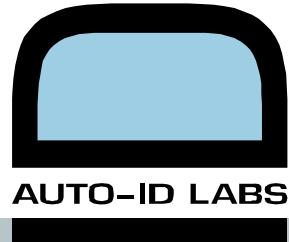
3.3.2. Administrators

3.3.2.1. Updating information

The administrators have to keep levels 1 and 2, the conceptual and theoretical levels up-to-date. This will involve keeping up-to-date with development in all the technology fields and making new entries into the database whenever necessary. Similar to the manufacturers, entries will be temporarily held for a moderator's inspection before being added to the live system.

3.3.2.2. Moderating information

The administrators are also the moderators of the data input by the manufacturers. The moderation process is described in later sections.



3.3.2.3. General administration

The administrators will be responsible for adding companies into the tool. As will be elaborated in later sections, there will be many different types of companies stored in the tool, for each of which there will be information to maintain.

Companies may also have users that have the ability to log on to the tool. These users have information stored such as their usernames and passwords. The administrators will be responsible for maintaining this information, too.

3.3.3. End-users

This is done in four phases as described below:

3.3.3.1. Phase 1 – Application questions

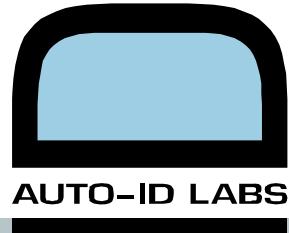
In this phase the user is required to answer questions about the application using the interface described in the interface section above. The answers to the questions provide initial conditions for some of the criteria filters that help to filter out unsuitable technologies.

3.3.3.2. Phase 2 – Refining the criteria filters

After answering the questions in phase one, the tool calculates initial conditions for the 14 filter criteria. The tool then will switch to the criteria filter interface. Some technologies will immediately appear as red ellipses. The initial conditions set in the phase one have filtered them out already.

This phase involves tweaking the values of the filters on each of the 14 criteria until the user is left with smaller set of technologies. As the number of technologies is reduced, the level of detail displayed can be increased. The increase in detail displayed allows the user to be more precise in the filtering.

3.3.3.3. Phase 3 – Reporting on technologies



Once the user is satisfied with the conditions set by the filters, the tool can produce a report. For each technology, the specification is given and links to datasheets are provided. The report might also include warnings about the factors that might need further investigation. For example if the user has answered in phase 1 that the product concerned has a lot of metal, then in the report, next of an RFID technology there may be a warning that metals have an adverse effect on read range and therefore further experimentation is needed.

3.3.3.4. Phase 4 – Entering experimental data

Once the user has selected a particular technology, an experiment with that technology can be performed. The results of the experiment, say, read range versus reader's output power of an RFID setup; those experimental values can be entered into the system as an augmentation of manufacturers' data. In later operations the user has a choice of using the original data or the revised data. The user can now continue the iterations in phase 2, but using the new value for read range and therefore allowing the filtering to be more accurate.

For applications where the manufacturers' specifications have to be verified before the part can be used, there will be the facility for the moderator to validate or override manufacturers' specifications in a similar manner to that of overriding a given value with an experimental value.

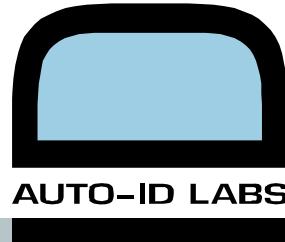
4. Implementation

This section explains each of the 14 criteria used by the tool in filtering.

4.1. Description of criteria

The following sections describe each of the criteria on which the user can filter technologies. The titles are mostly self-explanatory, but each criterion contains an explanation of how it is calculated for the different technologies. The filter threshold type is also noted for each criterion.

4.1.1. Data storage (bits)



Filter threshold type: "more than"

The values displayed on the graphs will be the maximum values for that media configuration.

Barcode: It takes the best method of encoding (given standards constraints) and calculates the amount of data that could be written using the greatest number of dpi to print on the area of media available.

CMB, Magnetic stripes, Optical Stripes, RFID and SC: They take the sum of all the user-writeable areas of data to be the total data storage.

Direct Part Marking (DPM): It takes the best DPM writer and the best encoding and calculates how much data could be stored in the area that is available.

4.1.2. Read range (mm)

Filter threshold type: "between"

As a criterion, read range is one of the criteria most susceptible to the subtleties of the environment in which the ID solution is being used. The threshold is set to "between", allowing the user to specify both a minimum and a maximum read range. The maximum range is, perhaps, for privacy reasons.

Barcode: The outside limit is given by the focal range of the best reader (given standards constraints). However, depending on the resolution of the barcode, the reader may not be able to read that given barcode over the whole of its focal range.

CMB, Magnetic stripes, Optical stripes and SC: They all have a read range of 0mm.

4.1.3. Simultaneous reads (number)

Filter threshold type: "more than"

This criterion describes the number of media that can be read simultaneously with one "read" of the reader. There are some unexpected results, such as a CMB reader that has two CMB interfaces and can read both at once.

Barcode, CMB, Magnetic stripes, Optical stripes, RFID and SC: are all defined by the greatest number of simultaneous reads for the best reader (given the standards constraints).

4.1.4. Read time (seconds)

Filter threshold type: "less than"



This is the time taken to perform a "read". Both simultaneous reads and read time must be considered together to calculate the overall number of media that can be read per second. We will illustrate how the same overall read rate could be achieved for very different situations and therefore why read rate was not chosen as the criterion.

On a fast moving conveyor belt each item might be passing through the reader's read area very rapidly. The reader only needs to be able to read one media at a time, but it has a very limited time in which to perform the "read".

At the other extreme, a warehouse rack might contain a thousands items. If they are not moving, then the read time is not limited and so it would be acceptable to read one hundred media simultaneously, in a time one hundred times as long as the conveyor belt reader.

The overall read rate is the same, but the solution for one application will not work for the other application.

Barcode: The best reader (given standards constraints) will dictate the average read time.

Magnetic stripes: The best reader (given standards constraints) will dictate the fastest speed at which the media moves through the reader. By dividing the stripe length by this speed, the read time is calculated.

CMB, Optical stripes, RFID and SC: The best readers (given standards constraints) give a data read rate. The user criteria for the amount of data storage required can be divided by the read rate to give a read time.

4.1.5. Read area (mm^2)

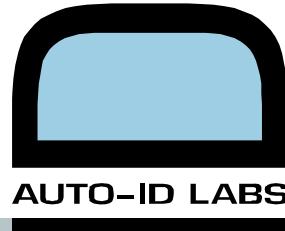
Filter threshold type: "more than"

The read area is the area, in a plane perpendicular to the axis of the reader, in which an ID will be acquired from a media. Where the user sets the threshold, it will be dictated by the amount of control the user has over where the ID media is placed. If the media can be guaranteed to be in the same place on every product and the product can be guaranteed to be in the same orientation and position when it is read, then the required read area would be 0 mm^2 . The greater the uncertainty or lack of maintaining order, the greater will be the required read area.

CMB, Magnetic stripes, Optical stripes and SC: All these must align exactly with their readers and therefore have 0 mm^2 read areas.

Barcode and RFID: The best reader (given standards constraints) will dictate the read area through its scanning angle or antenna.

4.1.6. Life expectancy (months)



Filter threshold type: "more than"

The life expectancy figure should be the nominal life over which the media will perform whatever functions are expected of the media. Like read range, this criterion will be strongly influenced by the environment in which the media is used.

Barcode, CMB, Magnetic stripes, Optical stripes, RFIF and SC: For now, the figure is just a nominal life for the media. Eventually it should be the worst life for all the sub-components of the media e.g. the worst life of the RF chip, the antenna, the chip-antenna connection, the physical media etc.

4.1.7. Constant transverse speed (mm/s)

Filter threshold type: "between"

The constant transverse speed limit is defined as the fastest speed, perpendicular to the reader, at which the reader is still capable of reading the media.

Barcode, CMB, Magnetic stripes, Optical stripes, RFID and SC: For now, the tool makes a very rough approximation to the maximum constant transverse speed. By taking the width of the read area (section 5.1.5) and dividing it by the read time (section 5.1.4) an upper limit for the constant transverse speed is calculated.

4.1.8. Character set (bits/character)

Filter threshold type: "more than"

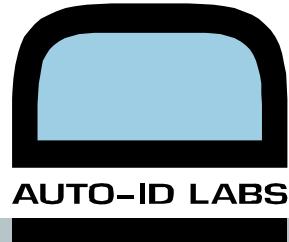
For several technologies, the readers output bytes, in which case the character set is said to be "byte" or 8 bits/character. For readers that emulate other readers, for example, a barcode reader that connects to the computer and emulates being a keyboard, the range of characters they can produce is limited. There are specifications that define which characters can be stored. As a generalization, they can be broken down into numeric, single-case alphanumeric, multi-case alphanumeric, ASCII and full Unicode.

Barcode and Magnetic stripes: They are defined by the encoding standard.

CMB, Optical stripes, RFID and SC: They can read byte streams or 8 bits/character.

4.1.9. Authentication to read data (cryptographic level)

Filter threshold type: "more than"



We have defined authentication to read data as the level of cryptographic protection required to access the user-stored data. This does not include how the user chooses to encrypt the information before they write it to the media. To illustrate this, consider a barcode — all the information is displayed and anyone with a barcode reader can access the information. The authentication to read data is therefore "none", even though the information actually might be encrypted and require decrypting before it is of use¹.

Compare this with a MiFare RF media, whereby the reader must authenticate itself with the media by means of a cryptographic key before any of the information stored on the media can be read. In this case, the authentication to read data is "SHA1" or whatever the name of the cryptographic mechanism used to protect the information is. It is possible that once the information is retrieved from the media, it, too, like the barcode, may need further decrypting to be of use.

Barcode, CMB, Magnetic stripes, Optical stripes, RFID and SC: The scale has not yet been fully defined and improvements are needed.

4.1.10. Humidity (RH) and temperature (°C)

Filter threshold type: "between"

For all the four characteristics — Operational humidity, Operational temperature, Extreme humidity and Extreme temperature — the user can specify the range within which the media must survive. "Operational" characteristics refer to the range over which the media must be fully functional. "Extreme" characteristics refer to the range over which the media will not be destroyed, but might not be fully functional.

Barcode, CMB, Magnetic stripes, Optical stripes, RFID and SC: As with life expectancy (section 5.1.6), for now the tool uses the characteristics of the media to give the ranges, but in future, the ranges should be given by the overlapping region of the ranges of all the subcomponents of the media.

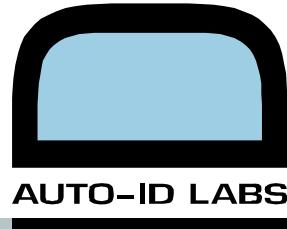
4.1.11. Physical area required (mm)²

Filter threshold type: "less than"

Barcode, CMB, Magnetic stripes, Optical stripes, RFID and SC: The physical area required for the media will be a characteristic of the media.

Direct Part Marking: The area will be a constraint of the part being marked and will be determined in phase 1, the question-answering phase.

¹ There are specialized secure barcodes that require the user to use a physical key as an overlay before the barcode can be read.



5. Technical Details

5.1. Software tools

The tool has been developed using free and open source software that are in the public domain. This section provides a short description of the technologies that have been used in the development of the tool.

Ruby on Rails (Rails) drives the dynamic web pages such as shown in Figure 1 and performs all the queries on the database. Radrails, the text editor for Ruby, was used as Integrated Development Environment (IDE) to write the tool's source code in Ruby. More discussion about Ruby on Rails are given in the following sections.

MySQL is the SQL database software used. The software used for this application is MySQL version 5.

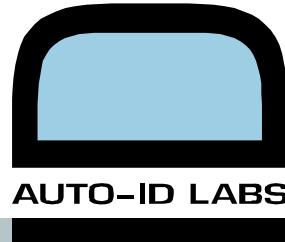
Macromedia Flash facilitated programming of the main user interface of the tool. In order to pass information between MySQL server and the Flash interface, Macromedia Remoting Connector and Weborb were used. The Remoting module, that is part of Macromedia Flash, allows functions in other programming languages to be called from within the native ActionScript code of Flash.

phpMyAdmin, a set of server side scripts that allows generation and maintenance of MySQL database, was used extensively to generate, enter and test databases.

5.2. Ruby on Rails

As the code of the tool was programmed in Ruby-on-Rails, this section provides a very brief introduction to this framework. Using any web development framework is going to be tricky, whether it is J2EE, ASP.NET, PHP or Rails. In long run though, you become more productive with Rails than with other tools, but it takes time to learn all the features.

Rails is a capable web application platform that has been around less than two years. It has gained significant popularity among web developers and programmers. The attraction stems from the strength of Rails. Rails uses a strict model-view-controller architecture that is well developed and organized. It is easy to build a basic system with Rails. That ease of development attracts many developers who have used PHP.



All that does not come free, Rails makes many assumptions about your database layout and application needs. It assumes that all tables use a single, non-compound, primary key. It does not support compound primary key. It does support multiple databases, but cannot coordinate transactions among them.

5.2.1. What is Ruby?

Ruby is a scripting language for quick and easy object-oriented programming. It is similar in many respects to Perl and Python. It is simple, straight-forward, extensible, and portable. Java and PHP programmers can pick up the language easily. Ruby was created in 1993, two years before Java and PHP. It has been less popular than some other programming languages because of lack of English documentations, books, web sites and blogs (originally all in Japanese). But that has changed recently when David Heinemeier Hanson introduced a new web development framework called Rails. Rails extend the Ruby programming language to a new level that is well suited for database backed web site development.

5.2.2. What is Rails?

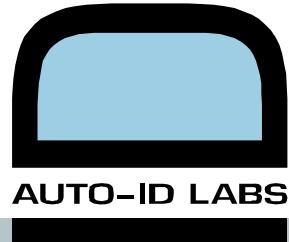
Rails is designed from ground up to create dynamic web sites that use relational database back-end. It adds functionality to Ruby programming language that makes web applications easier to configure. It can automatically generate a complete, but crude, web application from an existing database schema. Rails makes assumptions about database structure that if followed, makes generating a web site by only executing a single command possible.

All that said, you are able to override every database convention that Rails expects. Doing so, though, makes the Rails platform less productive.

5.2.3. What is Ajax?

The term Ajax stands for Asynchronous JavaScript and XML. The concept behind Ajax is to refresh only part of a page, not the entire page. At the heart of Ajax is something called the XMLHttpRequest object. When embedded in a Web page, this XMLHttpRequest object sends a request to a Web server. And the good news is that the request is asynchronous. Your browser doesn't stop what it's doing to wait for a response. Instead, the browser keeps doing whatever it was doing before issuing the request. At some future moment, when a response arrives back from the server, your browser processes the response and updates only a portion of the current page.

5.2.4. Installations



The easiest way to get Ruby on Rails running manually is to try one of the pre-packaged solutions. These include everything in one bundle: Web server, database, Ruby, Rails. For Windows, there's InstantRails that can be downloaded from <http://instantrails.rubyforge.org/>.

All you have to do is to download InstantRails, unzip it and copy everything to a directory on your C: drive. After installation you might need to change the default network socket port number for an Apache web server to something like 8000. You have to make the appropriate modifications to C:\InstantRails\conffiles\http.conf file. Now, you can access phpmyAdmin at <http://localhost:8000/mysql/> to configure your database.

Sometimes you might need to update your rails installation. The best way to do that is to run the following commands at a DOS prompt.

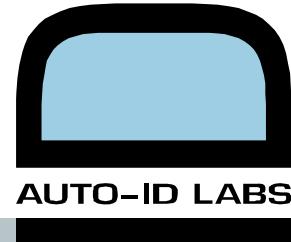
```
E:\Cam\Aero\idapp>gem install rails
Bulk updating Gem source index for: http://gems.rubyforge.org
Install required dependency activesupport? [Yn] y
Install required dependency activerecord? [Yn] y
Install required dependency actionpack? [Yn] y
Install required dependency actionmailer? [Yn] y
Install required dependency actionwebservice? [Yn] y
```

5.2.4.1. Create Database

The first step is to create a database for our application. Rails uses the concept of convention over configuration that means the name of the database must match the name of the project. Therefore we have to create a new database in MySQL named idapp_development. Open a DOS box and type the followings:

```
$mysql -u root -p
$create database idapp_development;
$grant all on idapp_development.* to 'ODBC'@'localhost';
$exit;
```

The next step is to use the script provided in the public/sql/idapp.sql directory to generate and fill in some initialization data into the database. That can be achieved using the import



option of phpMyAdmin. After the database is setup we can move to the next section that is installing the application itself.

Now open phpmyAdmin page and import the database schema into your database. On the left frame of the browser select idapp_developemnt database. In the right frame, click on the Import button at the top of the frame. Next, click on the Browse button to select your schema file and click on Go. That will create the necessary tables for your application and populates the database with initial values. Alternatively you can use the DOS box again and type in the following command:

```
$mysql idapp_development < idapp_schema.sql
```

5.2.4.2. Create the Web Application

Now it is time to create our application, to do that you will need to create a new Rails project as follows. Create a new project using the rails command line application (this will automatically create a new project directory).

```
C:\> rails idapp
```

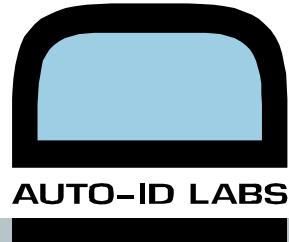
Go inside the idapp directory that the Rails has just created for your new project.

```
C:\> cd idapp  
C:\idapp>
```

Now that you are in the project directory, you might need to configure your project so the Rails applications will use the Rails-Mysql connection library. However the default database adapter in config\database.yml file is set to mysql.

There are a number of approaches to creating a Web application in Rails, but the simplest by far is to use scaffold, a command line feature in Rails since version 1.0.0, which will build the entire application in a single command.

```
C:\idapp_development> ruby script/generate scaffold idapp_tableName
```



Finally to run the web server type the following in the DOS window.

```
C:\idapp_development> ruby script/server --port 3000 --binding  
127.0.0.1
```

5.2.4.3. Upgrading Idapp

The IdApp application has been developed using RadRails from <http://www.radrails.org>, that is a framework for Ruby on Rails. The RadRails IDE is built on the Eclipse from <http://www.eclipse.org> RCP, and includes the RDT and Subclipse plug-in. The RadRails tools are also available as Eclipse plug-ins.

Use Rails to create a new empty Rails project. As by default RadRails sets the database name same as the name the project to save time and headache trying tweak the database settings in the application, the project name must be the same as the old one (idapp).

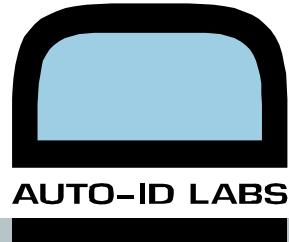
Then copy the following files to their new locations:

- everthing under app directory
- public/stylesheets
- public/images
- config/environment.rb
- config/routes.rb

Also copy the following plugins from the old project folder

- multiple_select, used in multiple selection scenario
- label_for, used in clickabletr
- nested_set_mod, for creating full path

Also it is necessary to modify enviroment.rb file and set the correct environmental parameters, and define singular and plural of any irregular word used as table name in the database (such as criterion, criteria)



It is necessary to delete the index file at public/index.rhtml, so any new connection will be re-routed to login page.

For Remoting and Weborb to function correctly, it is also necessary to copy the following files to their new location;

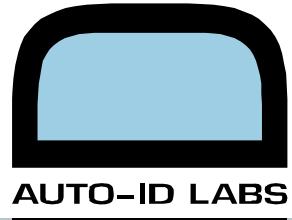
- app/services
- config/weborb-config.xml
- config/WEB-INF
- vendor/plugin/weborb

6. Conclusions

This project demonstrated that it is possible to characterize the majority of criteria on which technology selections are made. Many of the characteristics of an ID solution depend on the exact environmental in which the solution is used. The Tool does not model the environment but instead makes use of the experimental results to train it to choose the right technology.

The design philosophy of the Tool has been to make it open source and accessible. Thus others can easily use it and extend its functionality as required. As such we have chosen Ruby on Rails as our prime development software.

For it to be useful the tool needs to be kept up to date with proper information about media, manufacturers, readers, writers, standards, etc. This would require a continual effort from the users inputting latest information about the products and standards into the system.



References

- [1] European Commission. Regulation (EC) no. 1383/2003. Official Journal of 2 March 2003, L 196, page 7, 2003.