



*POESIA: Public Open-source Environment for a
Safer Internet Access*

DM Filtering Components Software
(Deliverable 8.1)
preliminary version

Project name:	POESIA
Project number:	IAP 2117/27572
Date:	January 27, 2003
Document Id:	POESIA-WP8-8.1
Version:	1.0
Deliverable:	8.1.
Related WP:	8 - Filtering Decision Mechanisms
Authors:	META: Stefan GUERRA, Alberto RAGGIOLI et al.
Status:	Public

Index

8.1 DM Filtering Components Software.....	3
8.1.1 Introduction.....	3
8.1.2. General Overview	3
8.1.2.1 Current development state	4
8.1.3. Architecture of the filtering Mechanism.....	4
8.1.3.1 Decision Maker politics	4
8.1.3.2 Bayesian Decision Maker (BDM)	5
8.1.3.3. Neural Network Decision Maker (NNDM)	6
8.1.3.4 Rule Based Decision Maker (RBDM)	6
8.1.3.5 Interactions between Monitor and DM	6

8.1 DM Filtering Components Software

8.1.1 Introduction

The filtering decision mechanism is the last link in the chain of the filtering process; it has to elaborate the information which it receives as input from the different and specialised modules, and must say the last word on the admissibility of the requested page.

Its input is constituted by score messages which come up from the:

- Language filtering components;
- Image filtering component;
- PICS filtering component;
- JavaScript filtering component;
- URL filtering component.

It is not a task of filtering components to submit messages to DM; actually, it is a primary task of the monitor component to submit to DM all scores that it receives from the over stated components.

The targeted filtering mechanism will respect the following major general characteristics:

1. reliability;
2. fastness;
3. configurability.

Since the first two points are obviously clear, the last point has to be explained: different users might have different needs concerning what has to be filtered and therefore the filtering mechanism cannot apply rules good for everyone, but has to be flexible and adaptive. Moreover the filtering mechanism should evolve according to the use and needs of the end-user; therefore we have chosen to implement the decision maker as a plug-in, which means that it might be possible, and it will be simple doing that, to add in a later stage a new decision mechanism which implements different filtering styles.

A major point concerning the decision mechanisms is how to handle the less of asynchronism of the responses coming up from the specific filtering components. At a first sight, this can really threaten the possibility of having a neural network to implement the decision logic, since POESIA development team, as the end users as well, believes that response time is, altogether with accuracy, the most important question that POESIA has technically to face. Waiting for all filters to complete their evaluation might lead to a significant overhead. Therefore, POESIA development team recognised the need of the DM to deliberate even if not every filtering component process is concluded.

8.1.2. General Overview

To respect the reliability request at least one of the filtering styles will be implemented using one of the standard machine learning models.

The mechanisms we have chosen to design and implement are the following different plug-in modules (other developers, outside of the POESIA consortium, could add more).

1. a Bayesian decision maker,

2. a neural network decision maker,
3. a rule based decision maker.

8.1.2.1 Current development state

At the moment the development team is targeting the first alpha release. In order to have it running it has recently been decided not to embed any complex decision mechanism in this very first release. This will instead be a simple two step decision mechanism based on a threshold. This component is about to be finished and will therefore be in use shortly.

8.1.3. Architecture of the filtering Mechanism

[version 1.0]

As any other filtering component the decision maker communicate with other POESIA components only through the monitor using three separate monodirectional channels (pipes) that are created by the monitor.

The input request channel (as seen by the DM) transmit requests from the monitor to the DM. In particular, it transmits data about the content to be judged. The request format is detailed below.

The output reply channel (as seen by the DM) transmit replies from the DM back to the monitor. In particular, decision scores are so transmitted to the monitor (that is in charge of storing and caching them).

The control input channel (as seen by the DM) transmit asynchronous control information from the monitor back to the DM. In particular, the monitor may abort a pending filtering activity. The DM, just like every other filtering component does, is expected to asynchronously watch for control input during its processing. The decision maker component is thus expected to be able to reset itself when so requested on the control channel by the monitor.

Each filtering activity (usually related to a single given Web content) has a unique numerical id (a positive integer). This id will correspond, e.g., to a single ICAP request received by the monitor. Request are terminated by a newline character, possibly followed by a request body of known size. Numbers are in decimal.

The possible input requests message from POESIA monitor to DM will therefore be as the following:

- SCORE req-number * label₁= score₁ * label₂= score₂* ... *label_n=score_n

[To be completed]

8.1.3.1 Decision Maker politics

[version 1.0]

As a general rule, DM will have to wait until it receives all messages regarding all different filtering component. In this sense, the monitor is in fact only propagating to the DM the different scores it

receives from the filtering components. Nevertheless, this rule has already proved to be too strict for most cases and therefore more flexible politics of decision are about be implemented.

The very first filtering politics which will reach the implementation phase is a Bayesian one. In this case and at least for the rule based filtering one, the response of the DM will be taken without waiting for the scores of every filtering component. In fact it might often happen that the decision maker component will be able to take a decision just on the basis of the URL or PICS filtering component scores.

A distinction has to be traced down for what concerns treatment of images. Since the embedded images in an HTML page can be, and often are, more than one, DM has to pre-process the different imaged tied to a request in order to reach an overall score for all picture embedded in the HTML page which is being analysed. This work is implemented in different way according to different DM mechanism, but in general it is clear that the particular politics that it implements may be different from the overall politics of the DM.

The following are the messages that DM issues as reply to the monitor:

1. ACCEPT req-number to accept a content.
 - a. Example ACCEPT 2345.
2. READY protocol-number.protocol-revision –
 - a. This reply is usually sent once by the filter, at initialization, to tell the monitor that the monitor is ready to accept requests. It is actually not a reply to a previous request, since it is a message sent as soon as possible by filters. The protocol described here (which is the same used by all other filtering component) has number 1 and revision 0, so the message is actually READY 1.0
3. REJECT req-number to reject a content.
 - a. Example REJECT 1234

[To be completed]

8.1.3.2 Bayesian Decision Maker (BDM)

[version 1.0]

Bayesian Decision Theory is one method used to solve Pattern Recognition problems, when those problems are posed in a particular way. The Bayesian decision maker receives from the specialised filtering modules (textual analysis module, image analysis module etc.) a vector of scores and translate this parameters in the $P(i)$ probability that the content received from the i -nth module is safe. All degree of safeness $P(i)$ are processed by a supervisor algorithm implementing a Bayesian rule which produces as its last outcome the error probability P_{err} . If this exceeds a fixed level, the page will be refused and the BDM will issue a REJECT message for the Monitor which could then give to the user instead of the requested one a denial page. This decision maker will be configurable in a way that the filtering system administrators can, through graphical interfaces: associate different levels of weight to the $P(i)$ values so to produce a more personal response by the filtering decision maker; modify the global filtering level P_{err} to act as more or less restrictive;

It is at the moment an open issue if rejected pages can automatically be added to black and white list of pages or URL which have to be permitted or denied.

The result of any new configuration should be tested on the fly on a recordset of previous filtering services that lies inside a certain degree of precision (e.g., all previous services where Perr level lied close to the Perr_max, the level over which the page is refused).

[To be completed]

8.1.3.3. Neural Network Decision Maker (NNDM)

[To be completed]

[At the moment since this WP started in November 2002 all effort are going to the realization of the BDM component; the next 8.1 deliverable will describe NNDM, if it will be recognised a a suitable option for Poesia.

8.1.3.4 Rule Based Decision Maker (RBDM)

[To be completed]

[At the moment since this WP started in November 2002 all effort are going to the realization of the BDM component; the next 8.1 deliverable will describe RBDM]

8.1.3.5 Advanced interactions between Monitor and DM

[version 1.0]

The possibility of having a more flexible system relies on the interaction between the monitor and DM. For instance, a denial page should be configured as to ask for permission to the filtering system administrators (in which case the page can be added to a white page list for a single person (if there is a selective login system) or for all users. For what concerns the Bayesian DM, on the basis of the administrators decision an overruling probability Poverrule, that takes into account the different initial P(i) values configuration, could be calculated. This Poverrule parameters would take its place among the other ones in the implementation of the Bayesian rule.

Advanced interaction between Monitor and the BDM, as described above, will be taken into account soon after the first implementation of the Bayesian DM.

[To be completed]