

Web Self-Controller: A Filtering Browser for Self-Control

Kyohei Yoshida

Graduate School of Engineering
Muroran Institute of Technology
Muroran, Japan

Email: s2124188@mmm.muroran-it.ac.jp

Shun Hattori

College of Information and Systems
Muroran Institute of Technology
Muroran, Japan

Email: hattori@csse.muroran-it.ac.jp

Abstract—Various Web contents exist on the Internet in recent years, and are used by many users. But some people are so fascinated by and absorbed in Web contents that their everyday life is affected. Such a person yields to the temptation because of her/his weak will, even if s/he tries to control her/his Web access by self-control. Moreover, it is difficult for such a person to overcome the temptation because there are few tools to help her/his self-control. Then, this paper proposes a method that applies the filtering technology conventionally-used for access restriction to self-control over Web access, and allows users to control their own Web access. Specifically, we create a prototype Web browser that has four functions to filter Web pages: a filtering based on their content, a filtering based on their ID (URL), a filtering based on their actions (buttons) such as purchasing and playing, and a time-based filtering.

Keywords—access control; browser; filtering; self control;

I. INTRODUCTION

By the recent development of the Internet, various Web contents have spread explosively. Conventionally the main use of the Internet was E-mail and Web search. However, it seems that the utilization rate of entertainment-type Web contents has increased and will also increase from now on. For example, Amazon of electronic commerce, YouTube of video hosting service, and Facebook of social networking service.

The growth of entertainment-type Web contents has brought about a social problem that not a few people worry about being disturbed in their daily lives, e.g., being absorbed in Web contents and so neglecting their studies. Such a problem that there are people who yield to the temptation of Web contents could be solved by a method that their access to Web contents is controlled by themselves, i.e., by their predefined access policies. Today, “Filtering” is used as a technology for access restriction.

General filtering technology is mainly used to clean up harmful Web pages for the purpose of youth protection. For a user’s “Self-Control” over access to Web contents with much temptation for her/him, all s/he has to do is set them as harmful by filtering technology. But various filtering rules are needed for her/him to set harmful specification. Because advanced knowledge is required, filtering rules are almost always described by network administrators of a government and a company. And so it is difficult for a general end-user

to describe a filtering rule. Of course, general users also have the possibility to set their filtering rules by filtering software. However, fine-tuned filtering rules for individual Web contents cannot be described, since all filtering software can only perform a simple setting. Therefore, a filtering system which is so excellent in operability that general users can set filtering rules easily is required for self-control.

Moreover, to overcome the temptation of Web contents, users need a compulsory filtering method that can bound their Web access. General filtering is often enforced to a user of network by not her/himself but its administrator. Therefore, the user is compulsorily restricted in her/his Web access. However, in general filtering for self-control, s/he can easily cancel her/his described filtering rule because s/he is concurrently both an administrator and a user of network. So, for self-control, we require a filtering system in which once a user sets a filtering rule, it cannot be canceled until specific conditions are fulfilled.

From the above, this paper proposes a novel filtering method that allows anyone to exercise her/his self-control over her/his own Web access, and verifies its usefulness by developing its prototype system. It has four functions to filter Web pages: a filtering based on their content, a filtering based on their ID (URL), a filtering based on their actions (buttons) such as purchasing, and a time-based filtering.

II. RELATED WORK

A. AntiSpoiler

AntiSpoiler [1], [2] which is a browser to filter spoilers dynamically is one of related researches. It is a filtering system to filter the result (e.g., victory or defeat) of a sports match. The system prevents spoilers by filtering text, i.e., “content” in the composition of Web pages. It prepares four kinds of methods for visualization of filtering spoilers, while this paper adopts the hiding method from among them to filter content of tempting Web pages for self-control. There are also browser extensions to filter text [3], [4].

B. i-filter

i-filter [5] is mentioned as another related research. The filtering software is developed and sold by Digital Arts Inc. When users beforehand register the URL of a Web page to be

intercepted by this software, it performs the filtering based on URL, i.e., “ID” in the composition of Web pages. This paper also implements the URL-based filtering function.

Moreover, i-filter has a function of the Internet Timer. This function enables filtering settings conditioned by time. But a concrete date cannot be specified because it allows users to condition their filtering rules only by the specific time zone in the specific day of week. In this case, it can filter only in seven days at the maximum, e.g., a temporal specification “from Sunday to Saturday” for the Internet Timer, but it cannot filter in the long term. Therefore, this paper aims at a filtering system with long-term filtering by adopting a filtering method based on date.

C. Comparison of filtering targets

AntiSpoiler filters Web pages based on their “content,” and i-filter filters based on their “ID”, while our proposed system for self-control can filter based on “action” as well as “content” and “ID” of Web pages. In addition, this paper implements “time specification” function, which is a different technique from i-filter’s Internet Timer. Table I shows the difference between our system and two related researches with respect to their filtering targets.

Table I
COMPARISON OF FILTERING TARGETS.

	content	ID	action	time specification
AntiSpoiler [2]	✓	-	-	-
i-filter [5]	-	✓	-	✓*
Proposed System	✓	✓	✓	✓

* ... i-filter’s time specification differs from our system.

III. PROPOSED METHOD

A. Creating a filtering browser

We have various ways to filter our own Web access. For example, we have a way of building a proxy server and restricting access to the Internet from our PC. Every time our PC tries to access the Internet, a proxy server intercepts and restricts the access. But it takes time and effort to learn about a proxy server, and this problem is an obstacle for those who want to filter their Internet access immediately. Moreover, a way of building a proxy server is not suitable for the purpose of this research “anyone can filter easily,” since it is difficult for general end-users to describe a filtering rule for a proxy server.

Hence, if a Web browser, which has familiar to many users, can be used for filtering, the filtering can be used easily because its setup is intuitive and intelligible. As another reason, a Web browser is easy to be extended with various functions requested by users from now on. This research develops an original Web browser (henceforth called “filtering browser”) in consideration of its high operability and extensibility. Our proposed filtering browser for self-control has the following four functions: content filtering function, ID filtering function, action filtering function, and time specification function.

B. Content filtering function

This is a function which filters the text on a Web page. The filtered character is replaced with a blank. To use content filtering function, it is necessary for a user to beforehand add the word(s) that s/he wants to filter out to a content filtering list. For example, it is necessary to register “baseball” to the content filtering list if users want to filter out the word “baseball” from the text on their accessing Web pages.

This function is convenient to prevent digression from pertinent details. For example, when users search the Web for necessary information about a target, there is some possibility of their wasting time by seeing unnecessary information about the target because there are various information on Web pages. In such a case, users can have a browsing environment where they can efficiently acquire their necessary information by hiding the word(s) that distracts them. The concrete example of usage is shown below. And an example is shown in Figure 1, the word “computer Go” which is under “GNU Octave” on an original Web page is replaced with a blank on the content-filtered Web page.

Example A: Mr. A likes computer go (game of go)

He decided to search about the “Monte Carlo method” by Wikipedia for study. However, the Monte Carlo method is a technique applied to the computer Go which Mr. A likes. Therefore, if the Monte Carlo method is searched for, descriptions of computer Go related to it may be seen. Mr. A thought that he would be distracted and his study would not progress if he would see the word “computer Go”, and he decided to filter out the word “computer Go” for 30 minutes by content filtering function.

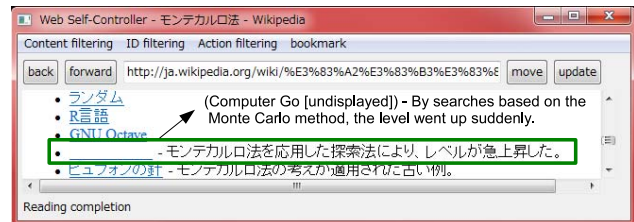


Figure 1. An example of content-filtered Web page.

C. ID filtering function

This is a filtering function based on URL. Not only match full of URL but also match partial of URL are available. In the case of match partial, if the URL of Web page that a user tries to access includes her/his specified domain for ID filtering, the access is blocked out. To use ID filtering function, it is necessary for a user to beforehand add the URL(s) that s/he wants to filter out to an ID filtering list. For example, it is necessary to register “https://www.google.co.jp/” to the ID filtering list if users want to filter out the Google’s homepage (https://www.google.co.jp/). The Google’s pages can be also filtered by registering its domain “google.co.jp” (match partial).

This function is convenient to concentrate on something. For example, because a user likes and is absorbed in watching a video clip on YouTube, there is some possibility of neglecting her/his study for an exam. In such a case, this function helps users to concentrate on their study for an exam by preventing them from connecting to YouTube until the end date of the exam. The concrete example of usage is shown below. And an example is shown in Figure 2, our browser blocks out the Web pages of YouTube if their URL contains the registered domain of “youtube.com.”

Example B: Ms. B likes watching a video clip at YouTube

Ms. B was going to watch a video clip on YouTube as usual. But she decided to watch a video after her homework was over because she remembered that she had still her homework to do. Ms. B began her homework, but she was not able to stop thinking about YouTube and concentrate on her homework. Then, Ms. B thought that she enjoyed watching a video while doing her homework, and so she accessed a Web page of YouTube at once. Although she could do her homework while watching a video at first, she ended up neglecting her homework and continued watching a video after all. Ms. B pulled herself together, and made up her mind doing her homework once again. It would take about 30 minutes to finish her homework if earnestly. Ms. B thought that she could be devoted only to her homework if she could not access YouTube for 30 minutes. She determined to filter out Web pages of YouTube for 30 minutes by ID filtering function.



Figure 2. An example of ID-filtered Web page.

D. Action filtering function

This is a function which filters out buttons with an action (e.g., add to cart, play, and post) on a Web page. The filtered button is undisplayed, and users become unable to push it. To use action filtering function, it is necessary for a user to beforehand add the action that s/he wants to filter out to an action filtering list. For example, it is necessary to register “add to cart” to the action filtering list if users want to filter out the buttons with the action “add to card.”

This function is convenient to enjoy window-shopping at an online shopping site. For example, users want to know the outline of goods and evaluation, and so they determine to check the goods information. There is some possibility of wasting money without thinking of the consequences, once users access an Electronic Commerce (EC) site. In such a

case, users can enjoy window-shopping at ease by preventing them from pushing buttons for buying on the EC site. The concrete example of usage is shown below. And an example is shown in Figure 3, our browser filters out the button of “add to cart” on Amazon.

Example C: Mr. C likes buying an item on EC

He wanted to buy comics on Amazon instantly, but he could not buy the comics to save the money for the next month. Since he could not purchase goods but wanted to check only the outline and reputation of goods, he decided to access the Web page(s) on Amazon where the Dragon Ball series vol. 1-42 were sold. However, there is some possibility of purchasing goods without thinking of the consequences since the goods can be easily purchased once he sees the goods on Amazon. Then, he determined to filter out buttons with the action “add to cart” on Amazon by action filtering function.

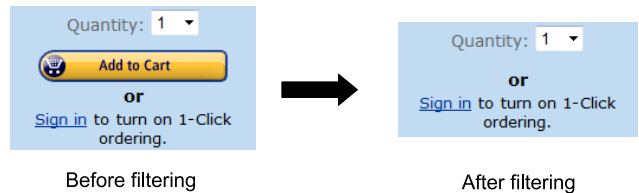


Figure 3. An example of action-filtered button.

E. Time specification function

Since our proposed method is a filtering for self-control, it is meaningless that users can cancel their filtering setting when they want to do so. Hence, our browser can combine time specification function with all three filtering functions shown above. Figure 4 shows the dialog box for setting of time specification function in content filtering function.

Time specification function enables a filtering setting not to be canceled until its specified time. When ID filtering function and time specification function are combined as an example, it is impossible to access a Web page whose URL includes a specific domain until the specified time, and it is also impossible to cancel the filtering setting. In addition, a filtering setting continues to be valid even after its specified time, unless it is explicitly canceled by users themselves.

IV. EXPERIMENT

Our developed system was tested by nine persons who use the Internet daily (incl. three Computer Science students, five non-CS students, and one working person). After that, we had the testees answer several questions about the operability and usefulness of our system. The procedure of evaluation experiment is shown below.

- 1) Each testee reads a questionnaire description. It mentions how to use our system and examples of use of each function, etc. Note that the usage examples for each function are the same as shown in Section III.

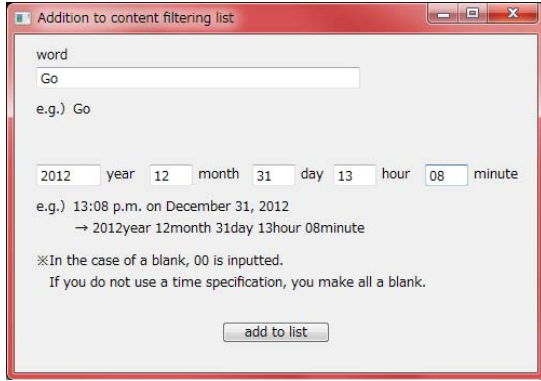


Figure 4. An example of setting dialog box for time specification function in content filtering function.

- 2) Each testee uses various functions of our filtering browser while the author is present at the experiment. If a testee questions, the author replies to the question. Filtering settings are the same as the usage examples shown in Section III. However, time specification function is set to 3 minutes after the current time to reduce testee's burden.
- 3) Each testee accesses the Web page for questionnaire, and answers the questionnaire.

A. The questionnaire result about the operability

The questionnaire about the operability of our filtering browser gives the following questions.

- To what extent could you understand our content filtering function?
- To what extent could you understand our ID filtering function?
- To what extent could you understand our action filtering function?

Each question has four levels of evaluation items: "I understood well," "I understood," "I hardly understood," and "I could not understand." Table II shows the breakdown and average of four-level evaluation for each filtering function about the operability. An average value is calculated by considering "I understood well" as 4 points, "I understood" as 3 points, "I hardly understood" as 2 points, and "I could not understand" as 1 point.

Table II
THE BREAKDOWN AND AVERAGE OF FOUR-LEVEL EVALUATION FOR EACH FILTERING FUNCTION ABOUT THE OPERABILITY.

	****	***	**	*	Avg.
Content filtering function	8	1	0	0	3.89
ID filtering function	8	1	0	0	3.89
Action filtering function	7	2	0	0	3.78

- **** ... I understood well
- *** ... I understood
- ** ... I hardly understood
- * ... I could not understand

From the result of Table II, it can be said that any filtering function is intelligible and excellent in operability. However testees were in an environment where it was easy to resolve their questions, because the author replied to their questions directly at the time of an experiment. Such an environment is considered as one of factors of the result of high understanding level. Therefore, the way of the evaluation experiment in this research cannot measure exactly whether or not the operability is good. To measure it exactly, a testee needs to use the system without an author's attending.

And our system is easy for a user to understand because it has a few functions now. But as various functions will be added from now on, the burden of a user's learning about our system will also increase. It is necessary to aim at more intuitive and intelligible operability for the future. Specifically, it is necessary to make a visually intelligible system by implementing a function that sounds an error-sound or displays a popup window when a user mistakes her/his operation.

B. The questionnaire result about the usefulness

The questionnaire about the usefulness of our filtering browser gives the following questions.

- Is our content filtering function helpful for your self-control?
- Is our ID filtering function helpful for your self-control?
- Is our action filtering function helpful for your self-control?

Furthermore, each testee answers the reason why s/he selects an evaluation item for each question. Each question has four levels of evaluation items: "It is likely to be helpful," "It is likely to be helpful a little," "It is hardly likely to be helpful," and "It is unlikely to be helpful." Table III shows the breakdown and average of four-level evaluation for each filtering function about the usefulness. An average value is calculated by considering "It is likely to be helpful" as 4 points, "It is likely to be helpful a little" as 3 points, "It is hardly likely to be helpful" as 2 points, and "It is unlikely to be helpful" as 1 point.

Table III
THE BREAKDOWN AND AVERAGE OF FOUR-LEVEL EVALUATION FOR EACH FILTERING FUNCTION ABOUT THE USEFULNESS.

	****	***	**	*	Avg.
Content filtering function	2	3	3	1	2.67
ID filtering function	6	2	0	1	3.44
Action filtering function	7	1	0	1	3.56

- **** ... It is likely to be helpful
- *** ... It is likely to be helpful a little
- ** ... It is hardly likely to be helpful
- * ... It is unlikely to be helpful

From the result of Table III, it turns out that the evaluation of content filtering function is low. We think that one of

factors of the problem is the way of the question, “Is our filtering function helpful for your self-control?” In such a question way, those who do not need the content filtering function for self-control give low evaluation. Accordingly, it is difficult to guess that content filtering function is useful for those who need the function.

Then, using only the result of those who need content filtering function, it is judged whether or not the function is useful. We reach a conclusion by hearing the reason why a testee felt her/his reply of the question, “Is our content filtering function helpful for your self-control?” And we exclude the persons who answered the reason that seems that they did not feel the necessity of the function from the questionnaire result. Specifically, the answer of “Because I do not use for self-control” etc. was removed.

As a result, five persons demanded the function and four persons did not demand it. Among the persons who answered that s/he wanted the function, two persons answered “It is likely to be helpful,” one person answered “It is likely to be helpful a little,” and the other answered “It is hardly likely to be helpful.” There was no person who answered “It is unlikely to be helpful.” By targeting only persons with the demand of content filtering function, Table IV compares the average of four-level evaluation only for those with demand and the average for all from Table III.

Table IV
THE COMPARISON OF THE AVERAGE EVALUATION ABOUT THE USEFULNESS OF THE CONTENT FILTERING FUNCTION WITH OR WITHOUT DEMAND.

	Number of persons	Avg.
Demand disregard (Table III)	9	2.67
Those with demand	5	3.20

From the Table IV, it turns out that the usefulness of the content filtering function is comparatively high for the persons who need the function. From the above consideration, it can be said that the reason of the low evaluation of the content filtering function in Table III was that the authors did not consider the testees’ demand. And it turns out that there is a person who evaluated “It is hardly likely to be helpful” among the persons who feel the necessity for content filtering function. This respondent expressed the opinion “Even if a tempting word is replaced with a blank, it can be guessed easily since the word was registered by a user itself, and so I do not feel the necessity of using the content filtering function.” This points out the problem of content filtering function in the present. It is thought that this problem can be improved by implementing a related word filtering function. Because not only a word specified by a user but also its related words are simultaneously filtered out by the related word filtering function, a blank on a Web page is not always hiding the word specified by the user itself. Therefore, it becomes difficult for the user to guess the word that is replaced with the blank and the respondent’s

trouble might be improved. From the above consideration, a related word filtering function needs to be implemented to improve the usefulness of this research.

C. The experimental accuracy of action filtering function

Besides the above-mentioned experiments, the experiment about the accuracy of our action filtering function was carried out. The experiment used three representative EC sites: Rakuten Market, Amazon.co.jp, and Yahoo! shopping. Using the commodity-sales pages on each EC site, the experiment investigated whether or not a button for purchasing a merchandise can be filtered out by our action filtering function. It used the top 50 goods on the comprehensive ranking of Rakuten Market and Yahoo! shopping, and the top 10 goods on each ranking of 5 genres of Amazon because there was no comprehensive ranking on Amazon.

First, the result for Rakuten Market is described. There are various named button for purchasing a merchandise on Rakuten Market, which has a lot of shops, because each shop owner can freely change a name of a button on her/his shop’s Web page. In this experiment, we added “putting into a shopping basket” to an action filtering list because it is the default name of button. Table V shows the experimental result of action-filtering the top 50 goods on Rakuten Market.

Table V
THE ACTION-FILTERING RESULT OF 50 GOODS ON RAKUTEN MARKET.

Filtering result	Number of goods
Success	32
Failure	8
Sold out	5
Impossible to investigate	5
Total	50

A tag of “Sold out” is assigned to a Web page for a goods where we cannot buy the goods because its stock has run out. Note that a button for buying a sold-out goods is not displayed and cannot be pushed without action-filtering the button. A tag of “Impossible to investigate” is assigned to a Web page which cannot be displayed well by our filtering browser because of its atypical style sheet and checked by the author.

From the result of Table V, the experimental accuracy of action-filtering buttons for purchasing a merchandise on Rakuten Market only by registering their default name “putting into a shopping basket” is 80% (= 32/40). Here, it is calculated without taking account of Web pages tagged by “Sold out” or “Impossible to investigate.” And the accuracy for action-filtering cannot be 100% because each shop owner can freely change a name of a button for buying a goods on her/his shop’s Web page. However, if a user continues to add action filtering rules, the accuracy can be brought close to 100%. In this experiment, the number of Web pages which our browser failed in action-filtering was eight, but they had

only four kinds of button names. Therefore, all users have to bring the accuracy close to 100% is to register the four kinds of button names as well as the default name in this experiment.

Next, for Amazon.co.jp, the author adds “add to shopping cart” to an action filtering list. As a result, our browser was able to perfectly filter out buttons for buying a goods if the goods had been already released and in stock. In the case of goods for reservation, their buttons are named by “pre-order: add to cart.” So, only the setting of “add to shopping cart” could not filter out the buttons for reservation because their name is different from the setting.

For Yahoo! shopping, the author adds “add to cart” to an action filtering list. As a result, our browser was able to filter out all buttons for buying 50 goods. So, it can be said that the action-filtering accuracy for Yahoo! shopping is 100%. This is because all the pages for 50 goods adopt quite the same button name. The result is also because the top 50 goods used for this experiment included no goods for reservation by chance. If a user wants to filter out buttons for reservation as well as for purchasing a merchandise, it is necessary to add a different button name “pre-order” to an action filtering list because their names are different from each other.

The action filtering function is a filtering based on HTML document, and users can filter out a targeted button of Rakuten Market absolutely by registering its name currently displayed to an action filtering list because its name on display and its name on HTML document are always the same.

On the other hand, because Amazon.co.jp and Yahoo! shopping use a picture button, its name on display and its name on HTML document may differ from each other. In this case, users cannot filter out a target button only by registering its name displayed to an action filtering list. However, since these sites have the constant kind of button, the filtering accuracy can be made 100% by registering beforehand the button name on HTML document.

To make the filtering accuracy 100% on Rakuten Market, a user theoretically has to describe infinite filtering rules beforehand. Although linguistic expressions with the meaning of purchasing a merchandise are not infinite in practice, users have to describe many rules compared with Amazon.co.jp and Yahoo! shopping. Therefore, for a shopping site which has many kinds of buttons like Rakuten Market, each user has too large burden to describe filtering rules. Then, we require a system which carries out a certain support to reduce the burden. For example, even if a user adds only the default button of “putting into a shopping basket” to an action filtering list, the function would identify whether or not a button on a Web page is for purchasing a merchandise by semantic similarity between words on buttons and similarity between positions of buttons, i.e., where a button is placed in a goods page.

V. CONCLUSION

In recent years, the utilization rate of entertainment-type Web contents has increased, e.g., Amazon of electronic commerce, YouTube of video hosting service, and Facebook of social networking service. As a result, new forms of communication and business by the Internet were produced, and people’s living environment changed dramatically. However, those who spend a lazy life of neglecting their studies appeared because they were absorbed in attractive Web contents. Web contents have no problem, if users can switch their mind in work and amusement well. But, for those who tend to escape to amusement because they have a weak will, Web contents not a little affect their real life. So, to prevent users from excessively devoting to Web contents, this paper has proposed “a filtering method for self-control” that can filter their own Web access, and we have developed a system based on the method.

The created system is a Web browser that has four functions for self-control: content filtering function, ID filtering function, action filtering function, and time specification function. By conducting an evaluation experiment to general nine persons, we have evaluated the operability and usefulness of its four functions.

For the operability, all filtering functions were evaluated high, but it would be because our system has few implemented functions. As various functions will be added from now on, the operation of our system may become difficult. So it is necessary to make a intuitively intelligible system by implementing a function that sounds an error-sound or displays a popup window according to the operation.

For the usefulness, only content filtering function was evaluated low a little. There was an opinion that “Even if a tempting word is replaced with a blank, it can be guessed easily since the word was registered by a user itself.” by the person who gave our system low evaluation. This problem could be improved by simultaneously filtering out related terms of the word specified by a user, and the content that a blank hides become difficult to be guessed.

In the future, we try to develop “Mobile Self-Controller” by applying our proposed method to mobile devices, because now many people access Web pages by smartphone, tablet, and so on.

REFERENCES

- [1] Satoshi Nakamura, “AntiSpoiler: An Web Browser to Filter Spoiler,” IPSJ SIG Technical Report, Vol.2010-HCI-139, No.17, pp.1–8 (2010).
- [2] Satoshi Nakamura and Takanori Komatsu, “Study of Information Clouding Methods to Prevent Spoilers of Sports Match,” Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI’12), pp.661–664 (2012).
- [3] Bad Words Filter for Internet Explorer : <http://www.softpedia.com/get/Tweak/Browser-Tweak/Bad-Words-Filter-for-Internet-Explorer.shtml>
- [4] WebAborn homepage : <http://webaborn.herokuapp.com/>
- [5] i-filter’s homepage : <http://www.daj.jp/cs/>