Building a Usable Hegelian Inquiring System

Richard Cornelissen
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
r.cornelissen@student.utwente.nl

ABSTRACT
The paper describes the design of a working prototype of the Hegelian Inquiring System, as described by Churchman [3], using the triangulation methods specified by Denzin [5]. This system will aid its users in finding documents, articles, blogs and other sources about a certain subject s/he queried and searches for other sources, searches for the author, gets information about the document’s website, and so on. The goal of this system is to give the user the means to check the reliability of found articles. We also test the system’s usability and whether users are satisfied with its user interface.

Keywords
Internet, information trust, information services, Churchman, Hegelian inquiring system, Denzin, rationality, system design, design research.

1. INTRODUCTION
The internet consists of a large quantity of virtual information, found in online news articles, blogs, online dictionaries, to whole encyclopedias. However, because of this vast quantity, users should always question the truth and reliability about what they are reading when using the internet, as presently anyone can post anything on the internet.

In order to answer the internet user’s question about whether an article, document, web page, and the like is “true” or not, a tool must be provided to aid him. This tool is the Hegelian Inquiring System [3], which will give the user the ability to search for other references about the topic, find the author of the article, and see what website is being read from, by triangulation, as proposed by Denzin [5].

Denzin outlines four types of triangulation, all of whom are meant to check the truth of statements. These triangulation methods are data, investigator, theory, and methodological triangulation. Respectively, these will give the user other sources about the same subject, information about the author and the website of the document, what theoretical bias is present in the document, and the kind of methodology in the document (e.g. empirical or historical).

Data triangulation:
Data triangulation can be realized by finding additional sources about the same topic. This can be implemented by searching for other sources, websites, and information about the author of the document. This can be done by creating a word cloud of the document, which shows what issues are addressed more often.

Investigator triangulation:
This type of triangulation can be realized by detecting the background of both the author and other authors in the same field of interest. This may include personal homepages or social media.

Theory triangulation:
Theory triangulation checks if certain issues are more dominant in a document to detect assumptions within the document. This can be done by creating a word cloud of the document, which shows what issues are addressed more often.

Methodology triangulation:
A document can use multiple methods to examine a phenomenon, for example empirical or historical. The document that is being triangulated can be checked for which methods apply to it.

Churchman [3] specifies that the Hegelian Inquiring System applies Hegel’s [10] dialectic logic. He states that knowledge is found through a process of thesis, antithesis and synthesis and sees thesis and antithesis as two different views about a single phenomenon. Synthesis solves the conflict between thesis and antithesis by forming a new proposition.

The Hegelian Inquiring System, as explained by Wijnhoven, Meertens, and den Engelse [12], starts with formulating a thesis and a query. A document found via the query will then be triangulated using implemented versions of the aforementioned triangulation methods. If the document passes the triangulations, continue with specifying an antithesis and repeat the triangulations, if not, modify the thesis.

After both thesis and antithesis pass the triangulations, they can be compared and a synthesis can be formulated.

This paper will continue the research of Wijnhoven, Meertens, and den Engelse by designing and testing a working prototype of the Hegelian Inquiring System, named the Hegelian Inquiry Tool (HIT). Therefore, the main research for this paper is: “Does the Hegelian Inquiry Tool provide a usable and user friendly solution to check the reliability of document on the internet?”

Below, we will first specify the research methodology, after which we will list the requirements for HIT and show any existing solutions. The paper will then continue with an explanation of how HIT works and how we will evaluate it using a questionnaire.

2. RESEARCH METHODOLOGY
According to Peffers [8], design science research consists of a number of activities that entail the design and development of an artifact. These activities are:
1. Problem identification and motivation, where the specific research problem is defined and the value of a solution is justified.

2. Define the objectives for a solution. According to Peffers, resources required for this include “knowledge of the state of problems and current solutions”.

3. Design and development, simply put, the creation of the artifact.

4. Demonstration, this could involve “use in experimentation, simulation, case study, proof, or other appropriate activity”.

5. Evaluation, which entails the observation and measurement of how well the artifact is a solution to the research problem.

6. Communication. To diffuse the resulting knowledge, “the problem and its importance, the artifact, its utility and novelty, the rigor of its design, and its effectiveness to researchers and other relevant audiences such as practicing professionals” must be communicated.

This paper will summarize the results of these activities and show the development of a working prototype of the Hegelian Inquiring System, which we will call the Hegelian Inquiry Tool (HIT). To do this, we use all knowledge about the Hegelian Inquiring System.

The first step of Peffers’ design science has already been completed with section 1, where we define the problem statement and the Hegelian Inquiring System.

3. REQUIREMENTS

Before starting the search for existing Hegelian Inquiring System tools, a set of requirements is specified to list all features of the Hegelian Inquiry Tool (HIT). This section is part of the second activity of Peffers’ design science [8], defining the objectives for a solution.

3.1 Requirements of HIT

1. Hegelian Inquiring System process:

1.1. After starting the triangulation process, the user will be able to specify a subject and a thesis about that subject.

1.2. After specifying a subject and a thesis, the user will be able to triangulate a found document through triangulation methods explained below.

1.3. If the thesis survives all triangulation methods, he may specify an antithesis.

1.4. After supplying an antithesis, the user will be able to triangulate a found document through the same triangulation methods.

1.5. If the antithesis survives all triangulation methods, the user will be able to specify a synthesis from his thesis and antithesis.

2. Document search requirements:

2.1. The user will be able to enter a query based on a (personally thought of) thesis or antithesis in order to search for a document which he or she wishes to triangulate.

2.2. The user will be able to enter a link to a known document on the internet which he/she wants to triangulate.

3. Triangulation requirements:

3.1. The user will be able to use data triangulation on the selected document, which will search and show comparable material, for example by searching using words from a generated word cloud as a query.

3.2. The user will be able to use investigator triangulation, which will give the user the ability to enter the name of the author and the domain of the website and will then search for any available information about these data, e.g. on social media and via a “who is” tool.

3.3. The user will be able to use theory triangulation, which creates a word cloud of the selected document.

3.4. The user will be able to use method triangulation, which will give the user the means to find out what methodology is used in the document.

3.5. For every triangulation, the system will give the user the option to add a reliability-factor to it.

3.6. After the four triangulation methods, the user will be able to specify a reliability-factor about the whole triangulation of the (anti)thesis.

4. EXISTING SOLUTIONS

The Hegelian Inquiring System [3], as described by Wijnhoven et al. [12], consists of parts that are already available to the user, such as a means to search for alternate sources (search engines such as Google), the means to find more information about an author (people search engines such as pîpl), or word cloud generators such as wordle.net.

However, a tool combining all features of the Hegelian Inquiring System, allowing users to quickly search for other sources, author information and so on, has never been developed.

With this section, we complete the second activity in Peffers’ design science [8] and move on to the third step, designing and developing the Hegelian Inquiry Tool.

5. HIT DESIGN

As this implementation of the Hegelian Inquiring System [3] should be able to allow any user to search or specify documents which he or she can triangulate, the Hegelian Inquiry Tool has been implemented as a web service, accessible through the user’s browser. The Hegelian Inquiring Tool consists of the following packages, which will be further explained along this chapter:

- The metasearch package, developed for HIT to use multiple search engines to find as many results when querying for documents.
- The peoplesearch package, which is able to search for links to information about the author of a document.
- A whois-application which is able to contact a whois-server to search for information about a domain.
- A wordcloud-package to create a word cloud of a document.
- The Triangulator application, which is made with the Django framework [6], which is able to triangulate documents and provides the web user interface.
The design and development of these applications, including putting them together which will be explained in section 5.6, conforms with the third step in Peffers’ design science [8], the design and development of the artifact HIT.

### 5.1 Metasearch

The metasearch-package has been developed for HIT itself. Its goal is to search multiple popular search engines for a given query and to return the found results such that they all use a common standard. Currently, metasearch can search from Google, Bing Beta and Yahoo! Web, but it can easily be extended to use another search engine when given an Application Programming Interface (API) and a small piece of software to parse the results into a usable standard.

During implementation of this package, some problems arose, most importantly a working Google Search API. As no other fitting custom API could be found, a (quick and dirty) API has been created which requests and parses results from Google’s website. To do this, a browser emulator has been included into HIT [7], which can easily download a page and return it as a piece of text, which can then be parsed for search results.

![Figure 1. Class diagram of metasearch-package.](image1)

### 5.2 Peoplesearch

The peoplesearch-package has been designed to search for a list of links to webpages about a given author of a document. To do so, the original idea was to search social media, such as Twitter, LinkedIn and Facebook for information about the author, or use a people search engine, such as pipl, to find information.

However, due to the fact that these social media and people search engines provide no interface with which a third party service, such as HIT, can communicate and request information, these services have been excluded from peoplesearch.

Because of this, peoplesearch has only been implemented to search for information through the metasearch-package. Additional engines can still be easily added, for instance when these interfaces will be made available.

### 5.3 Whois

To understand more about a document, the user should also look at the website where it is published. To do this, the WHOIS protocol can be used.

The WHOIS protocol has been specified in RFC 3912 [4] and can give more information about a domain, such as the company owning it. To use this protocol, the whois-application selects a WHOIS server corresponding to the Top Level Domain (TLD) of the domain (e.g., ".com"), connects to the server and requests information about the domain. The WHOIS server will then respond with all information it has about that specific domain. Among other purposes, users can use this information to find out what company runs the website where he or she found a document.

![Figure 2. Flow chart of whois-package, using WHOIS protocol [4].](image2)

### 5.4 Wordcloud

Theory triangulation of a document on the internet has been specified as creating a word cloud of the document. According to Wijnhoven, this will allow the user to detect the theoretical bias of a document [12].

To do this, the wordcloud-package is able to use the aforementioned browser emulator to download the document. As this document is written in HTML, the wordcloud-package will strip it of all its useless contents – any embedded JavaScript, embedded Cascading Style Sheets and all HTML tags will be removed.

After this, a tag list will be created, which consists of all unique words used in the article sorted by occurrence. This tag list will then be used to create the word cloud.

![Figure 3. Flow chart of wordcloud-package.](image3)
5.5 Triangulator
The triangulator package contains the triangulation tools. These include the data, investigator, theory and methodology (or method for short) triangulators. As mentioned before, these are the tools that will aid the user in finding sources that agree with the selected document, finding information about the author of the document and the website that publishes it, providing a word cloud so the user may see any bias in the document and select any methodologies used in the document.

For every triangulation method, the user is also able to specify some notes on the triangulation and a reliability-factor for every triangulation. This way, the user can later on review his findings about the triangulated document.

5.5.1 Data triangulation
Data triangulation [5] is meant to find additional sources about the same topic as the document. To be able to do this, the data triangulator uses the wordcloud- and metasearch-packages together to first create a list of most used words (the tag list) and then using that list as a query with metasearch. Metasearch will then return search results from various popular search engines.

5.5.2 Investigator triangulation
To find more information about the author, the user is asked to specify some data about the author and the website that publishes his document (the name, optionally a location and the domain of the website). The investigator triangulator [5] then uses the peoplesearch- and whois-packages to find information about the author and website.

As stated previously, the peoplesearch-package is unable to use social media or people search engines. To cope for this lack of information sources, a link to pipl.com, a people search engine, has been provided for the user to manually find more information.

5.5.3 Theory triangulation
As previously mentioned, theory triangulation [5] can be achieved by creating a word cloud of the selected document. To do so, the wordcloud-package is used to create a word cloud with the most used words of the document.
5.5.4 Method triangulation

Method triangulation [5] checks which methodologies have been used inside the selected document. This is visually checked and specified by the user himself. Some methodologies are already present in HIT, such as empirical and historical, however, if the user is missing some of them, s/he can add them to HIT himself manually.

5.6 Putting it all together

All sub packages of HIT have now been discussed, however, the general outline of the Hegelian Inquiring System is to first state a thesis and triangulate a document about that thesis, then doing the same for an antithesis and ultimately formulate a synthesis from the thesis and antithesis.

HIT also follows this path. After the user indicates that he wishes to perform a Hegelian triangulation, he is able to specify a subject (for administration) and a thesis. After doing so he is able to either search for a document to be triangulated by providing a query (metasearch is used for this) or by providing a direct link himself. After triangulations for the thesis, the user will be shown a summary of his findings and specify a general reliability-factor about the thesis. If he chooses to continue with formulating an antithesis, he is once again asked to select or supply a document for triangulation. After showing a summary of the antithesis triangulation and again providing a general reliability-factor, the user is shown a complete summary of both thesis and antithesis and is asked to formulate a synthesis.

A flow chart of this process is given in Appendix I.

6. EVALUATING HIT

As the usability of the Hegelian Inquiry Tool (HIT) will be tested by Wal [11], the evaluation in this paper will be about the user satisfaction with regard to the web interface of HIT.

To do so, a tool provided by Perlman [9], called the Questionnaire for User Interface Satisfaction (QUIS), will be used. This instrument is based on the article “Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface” [2].

This instrument will ask general questions about the software, such as difficulty to use, but will also address display, terminology used, how easy it is to learn how to use it, speed, and so on. An example can be found at http://hcibib.org/perlman/question.cgi?form=QUIS. The full list of questions can also be seen in Appendix II.

To evaluate HIT, a demonstration will be given to a group of students, after which they get the chance to try HIT out for themselves. They are then asked to fill in the survey to specify their opinion about the user interface. This will complete the fourth and fifth step in Peffers’ design science [8].

6.1 Results and Discussion

The questionnaire was developed such that the results can be a sum of the answers of each question, giving a final score which depends on the number of people that have filled in the questionnaire.

Unfortunately, only thirteen people have filled in the questionnaire of Appendix II, but the results of this questionnaire are still quite clear and very consistent, as can be seen in Table 1.

The only question that has a relatively high score is “Position of messages on screen”, which shows the consistency of the position of messages. These results show that the developed Hegelian Inquiry Tool (HIT) is still far from perfect and has a lot of room for improvement.

The speed of HIT and its progress reporting (which is virtually inexistent) return the worst results. Furthermore, the students specify they find the sequence of screens confusing and feel the tool lacks adequate power. They also react moderately to how dull or stimulating and how frustrating or satisfying HIT is, both having a total of zero points (questions three and five).

<table>
<thead>
<tr>
<th>#Q</th>
<th>Sum</th>
<th>#Q</th>
<th>Sum</th>
<th>#Q</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>10</td>
<td>-1</td>
<td>19</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>11</td>
<td>7</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>12</td>
<td>2</td>
<td>21</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>-2</td>
<td>13</td>
<td>12</td>
<td>22</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>14</td>
<td>4</td>
<td>23</td>
<td>-2</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>15</td>
<td>-3</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>16</td>
<td>4</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>17</td>
<td>9</td>
<td>26</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>18</td>
<td>4</td>
<td>27</td>
<td>4</td>
</tr>
</tbody>
</table>
7. CONCLUSIONS

The Hegelian Inquiry Tool is meant to answer the internet user’s question about whether an article, document, web page, and the like is “true” or not.

The research question of this paper was “Does the Hegelian Inquiry Tool provide a usable and user-friendly solution to check the reliability of document on the internet?” To answer this question, we look at both the results of Wal [11] and the questionnaire held in this study.

Wal studied the usability of HIT by demonstrating it to a group of students and ask them to fill in a usability survey. This survey included questions about each triangulation method (data, investigator, theory, and method triangulation) and general questions about end-user satisfaction.

In general, the users believed the data triangulator to be the most trustworthy, but they do not believe the theory triangulator reaches its goal; show the bias and content of a document. Another complaint that came forth from the evaluations was the slowness of the tool.

According to the questionnaire held in this study, the Hegelian Inquiry Tool has not been found user friendly. Results show the system should improve on a lot of points, especially its speed and the reporting of progress. To do so, the data triangulator’s speed should be improved dramatically, which is also reflected in Wal’s study. Furthermore, an extended study needs to be held to find out what sequence of screens makes the tool less confusing to users.

However, we believe further user design and usability research and improvements will overcome these problems.

Other possible research includes combining the idea of HIT with an online search engine like Google, giving HIT direct and fast access to resources Google has to offer, such as fast search results and high server capacities.

With this paper, we complete the sixth and final step of Peffers’ design science [8].

8. REFERENCES


[7] Krumins, P., Browser Emulator. 2009. A browser emulator which is able to download the html code of a webpage by telling the server it is some popular browser. Available from: http://www.catonmat.net/blog/python-library-for-google-search


9. APPENDIXES

9.1 Appendix I: Flow chart of the Hegelian Inquiry Tool

This flow chart shows the way the user will use the Hegelian Inquiry Tool. The user will be able to start a triangulation on the homepage, will search for or specify a document to triangulate will see the results of the four triangulation methods and will be shown a summary of the triangulation. After this, he may choose to start triangulation for an antithesis, modify his thesis or stop entirely. If he chooses to continue triangulating for an antithesis, HIT will again allow the user to search for or specify a document, show the user the results of triangulating this document and show the user a summary, where he may enter a synthesis he made from his thesis and antithesis.

9.2 Appendix II: Questionnaire

The questionnaire used to test the satisfaction of the user about the user interface consists of 27 questions. These questions can either be answered with a score of -2, -1, 0, 1 or 2, or by specifying N/A (none available) and are ordered in five categories. Note that this is an edited version, with all the questions included in the real questionnaire, but with another design.

Overall reaction to the software

<table>
<thead>
<tr>
<th>Question</th>
<th>Score 0</th>
<th>Score 1</th>
<th>Score 2</th>
<th>Score 3</th>
<th>Score 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Totally miserable</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Difficult</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Frustrating</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inadequate power</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dull</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rigid</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Screen

<table>
<thead>
<tr>
<th>Question</th>
<th>Score 0</th>
<th>Score 1</th>
<th>Score 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reading characters on the screen</td>
<td>Easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Highlighting simplifies task</td>
<td>Very much</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organization of information</td>
<td>Very clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence of screens</td>
<td>Very clear</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Terminology and webform information

<table>
<thead>
<tr>
<th>Question</th>
<th>Score 0</th>
<th>Score 1</th>
<th>Score 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of terms throughout webform</td>
<td>Consistent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terminology related to task</td>
<td>Always</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Position of messages on screen</td>
<td>Consistent</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prompts for input</td>
<td>Clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer informs about its progress</td>
<td>Always</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Error messages</td>
<td>Helpful</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Learning

<table>
<thead>
<tr>
<th>Question</th>
<th>Score 0</th>
<th>Score 1</th>
<th>Score 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning to operate the webform</td>
<td>Easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploring new features by trial and error</td>
<td>Easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remembering names and use of commands</td>
<td>Easy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performing tasks is straightforward</td>
<td>Always</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help messages on the screen</td>
<td>Helpful</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supplemental reference materials</td>
<td>Clear</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Webform capabilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Webform speed</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Too slow</td>
<td>〇 〇 〇 〇 〇 Fast enough</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Webform reliability</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unreliable</td>
<td>〇 〇 〇 〇 〇 Reliable</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Webform tends to be</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Noisy</td>
<td>〇 〇 〇 〇 〇 quiet</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Correcting your mistakes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Difficult</td>
<td>〇 〇 〇 〇 〇 Easy</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Designed for all levels of users</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Never</td>
<td>〇 〇 〇 〇 〇 Always</td>
<td></td>
</tr>
</tbody>
</table>