Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 131 (2015) 312 – 326

Procedia Engineering

www.elsevier.com/locate/procedia

World Conference: TRIZ FUTURE, TF 2011-2014

Brewing Free Beer: Using Ideality to develop a 'free-to-use' TRIZ Effects Database

Andrew Martin

Oxford Creativity, 6-7 Bankside, Hanborough Business Park, Long Hanborough, Oxford, OX29 8LJ, United Kingdom

Abstract

This paper describes how the TRIZ concept of Ideality was used to inform the development and deployment of a 'free-to-use' TRIZ database of scientific and physical effects. The TRIZ 'Time and Scale' framework (also known as Nine Boxes, Nine Windows or System Operator) is used to structure the narrative based on three phases: Pre- Development, Development and Post-Development.

In the Pre-Development phase it was recognized that most comprehensive effects databases tend to be commercial products providing excellent content but at a price presenting a potential barrier to their use, especially by those new to TRIZ. The few 'free-to-use' effects databases that were available tended to be less comprehensive than the best of their commercial rivals.

This led to an Ideal Outcome (or Ideal Final Result) for the effects database: "delivery of complete and perfect results to any user without restriction or cost". This highlighted a contradiction between the performance of the system (primarily represented by the quality and quantity of the results it produces) and the cost of delivery (primarily represented by the cost of developing the database infrastructure and collecting and collating the database content).

As the project moved into the Development Phase this contradiction was tackled using a three-pronged Ideality-based strategy of: focus on delivering only the essential Benefits, reduction of Harms that detract from the essential Benefits and finally reduction of Costs by using existing and/or low-cost resources.

Focusing on essential Benefits resulted in the selection of a simple database design that minimised the data required to populate the database. The principle Harm (where Harm is an output that is not useful) considered was that of an inaccuracy in the results delivered to the User. A taxonomy of database inaccuracies was compiled and each identified inaccuracy type considered in terms of its influence on the Ideality of the database system. This revealed, somewhat surprisingly, that inaccuracies were not only inevitable but were also not critical to the overall usefulness of the system. The principle resources used for cost reduction were: open-source database development tools, available internet resources and people.

The Post-Development Phase was triggered in October 2011 with the deployment of the database on the Oxford Creativity website (www.triz.co.uk). The principle activity since that time has been a series of content updates. In addition consideration has been given (somewhat belatedly) to other aspects of the Post-Development column of the Nine Box model. The User's perception of Ideality has shifted from "Can I have more results" to "There are a lot of suggestions here – can they be filtered in some way?" This is a useful reminder that (in common with most systems) the database falls short of delivering the Ideal Outcome of complete and perfect results. Some candidate routes towards the realization of this Ideal are suggested.

Peer-review under responsibility of the Scientific Committee of TFC 2011, TFC 2012, TFC 2013 and TFC 2014 – GIC
doi:10.1016/j.proeng.2015.12.395

Peer-review under responsibility of the Scientific Committee of TFC 2011, TFC 2012, TFC 2013 and TFC 2014 – GIC
Keywords: Effects Database, TRIZ; Ideality, System Operator;

1. Introduction (and Nomenclature)

This paper discusses how the development of a free-to-use database of engineering and scientific effects was informed by the TRIZ concept of Ideality.

Effects databases are an important and established part of the TRIZ tool kit, with paper-based examples appearing in many TRIZ reference works [1][2][3][4] and a number of computer-based versions available on the internet [5][6] or incorporated into TRIZ-based software tools.

This paper is structured using the TRIZ 'Time and Scale' framework [7] (also known as Nine Boxes, Nine Windows or System Operator) shown in Figure 1.

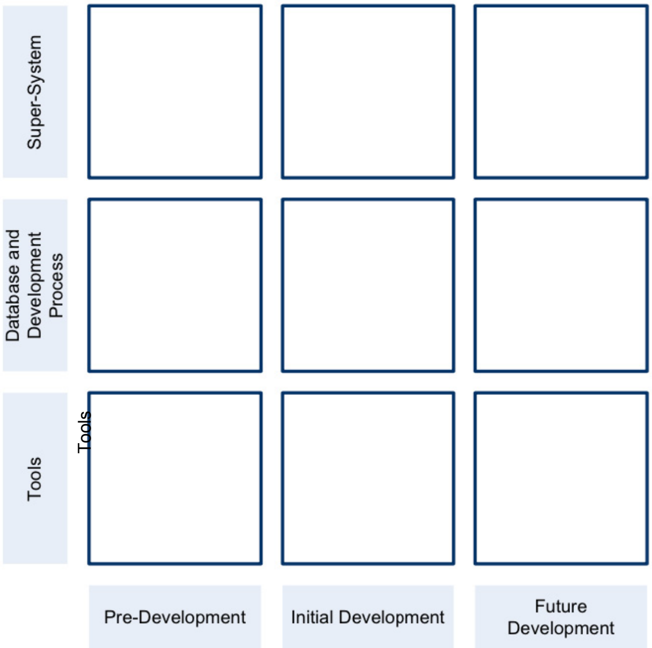

Fig 1. Time and scale (nine boxes) narrative framework

The narrative is based on three time phases: Pre-Development, Development and Future Development and three levels of system hierarchy/scale: Context, Database Development Process and Tools:

Nomenclature

Effect	A scientific or engineering effect
EDBS	The effects database system that is the subject of this paper
EDBS Query	A query made by the User to the EDBS
Query-Effect Relation	The link between an EDBS query and a relevant effect
The user	The person using the EDBS

2. Pre-development: super-system

2.1. Motivation for a free-to-use effects database

The motivation for the development of a free-to-use database of scientific and engineering effects was prompted by the author's dissatisfaction with the range of such databases available to TRIZ users. Available effects databases can be categorized as follows:

- Paper-based databases (from books or other TRIZ publications) generally having limited content
- Internet-based 'free-to-use' databases generally having more content than the paper-based databases
- Commercial software-based products generally providing the greatest content but requiring some form of payment by the user

What appeared to be missing was a database that came closer to the best of the commercial systems in terms of usefulness but that was also available free of charge to the user. Of particular concern are people new to TRIZ. Such users may be reluctant to invest in a commercial effects database product until convinced of the merits of the concept of an effects database. Confidence in effects databases (and maybe, by inference, in TRIZ itself) may be eroded if the low-cost database solutions fail to report all the relevant effects that a user is aware of.

2.2. Initial formulation of the ideal outcome

An initial Ideal Outcome (or Ideal Final Result) for the new free-to-use effects database system (EDBS) was formulated as:

"Delivery of complete and perfect results to any user in any circumstance without restriction or cost"

There is an apparent contradiction between the performance of the system (primarily represented by the quality and quantity of the results it produces) and the cost of delivery (primarily represented by the cost of developing the database infrastructure and collecting and collating the database content).

3. Pre-development: database and development process

3.1. Ideality-based strategy development

The TRIZ concept of Ideality was used to suggest a development strategy for the database. The Ideality of a system is defined as:

$$I = (B/(C + H) \qquad (1)$$

where:

I indicates Ideality

E indicates *benefits* (useful outputs)

C indicates *costs* (all inputs)

H indicates *harms* (non-useful outputs)

The *benefits* can be divided into two categories: the primary *benefit*, Bp and secondary *benefits*, Bs, giving:

$$I = (\uparrow Bp + Bs)/(C + H) \qquad (2)$$

From this we can infer that strategies for increasing Ideality include (amongst others): Decrease secondary *benefits* while disproportionately increasing primary *benefits*:

$$\uparrow I = (\uparrow\uparrow Bp + \downarrow Bs)/(C + H) \quad (3)$$

Reduce secondary *benefits* while reducing *costs* disproportionately more:

$$\uparrow I = (Bp + \downarrow Bs)/(\downarrow\downarrow C + H) \quad (4)$$

Allow *harms* to increase while reducing *costs* disproportionately more:

$$\uparrow I = (Bp + Bs)/(\downarrow\downarrow C + \uparrow H) \quad (5)$$

Find ways of reducing *costs* without having a significant effect on *benefits* or *harms*:

$$\uparrow I = (Bp + Bs)/(\downarrow C + H) \quad (6)$$

where:

$\uparrow$ indicates increase
$\downarrow$ indicates decrease
$\uparrow\uparrow$ indicates disproportionately greater increase
$\downarrow\downarrow$ indicates disproportionately greater decrease

In order to convert these conceptual strategies into something more tangible it is necessary to understand what a reduction in secondary *benefits* and an increase in *harms* means in practical terms and then to assess their significance on the Ideality of the system.

3.2. Taxonomy of effects database harms and insufficiencies

A simple taxonomy of *harms* and insufficiencies in database results was compiled in order to provide a framework for assessing their significance.

- Failure to report a relevant effect. In practical terms it is impossible to report every relevant effect due to the scope and ever-expanding nature of human knowledge and so failures of this sort are inevitable. However the primary purpose of an effects database is to provide knowledge of relevant effects and so any failure to report a relevant effect constitutes an insufficient benefit. It is also a harm in that it may mislead the user into believing that the unreported effect is not relevant. Given that the primary benefit of the database is the reporting of relevant effects and that an unreported relevant effect might be the very one that enables the user to solve the problem at hand, the significance of this inaccuracy was rated as high.
- Reporting an irrelevant effect as relevant. This constitutes a *harm* as the user will be misled (a non- useful output). It is likely to lead to an unnecessary *cost* as the user may waste time following unfruitful lines of development. However it is considered likely that such errors will be discovered relatively quickly. The significance of this inaccuracy depends upon the relative proportion of irrelevant effects reported as relevant to the number of correctly reported relevant effects. If this ratio is kept relatively low then the significance of the inaccuracy will be low.
- Providing inaccurate information about an effect. This constitutes a *harm* as the user will be misled and in many cases a *cost* as the user may waste time following unfruitful lines of development. The significance of this is rated as low as inaccuracies of this sort are likely to be discovered fairly quickly.
- Providing insufficient information about an effect. This constitutes an insufficient secondary *benefit*.
- Given the wide availability of information from other sources (such as the internet) the significance of this inaccuracy is rated as low.

The assessed significance of the outcomes of database inaccuracies is summarized in Table 1:

Table 1. Significance of Insufficiencies and Harms Inaccuracies in Database Results

Inaccuracy or Insufficiency	Significance of Outcome
Failure to report a relevant effect as relevant	High
Reporting an irrelevant effect as relevant	Low if proportion relative to correctly reported effects is low
Providing inaccurate information about an effect	Low
Providing insufficient information about an effect	Low

3.3. The three-pronged strategy

The above analysis suggests that a promising strategy for a free-to-use effects database is to:

- Focus on including as many effects, queries and query-effect relations as possible and at the same time reduce effort spent on providing the secondary *benefit* of providing supporting information about effects. This will also have the effect of reducing the *harm* of unreported relevant effects.
- Be prepared to tolerate a proportionately small number of irrelevant effects reported as relevant. In practice this means not spending effort obtaining definitive evidence that a particular effect is relevant to a query before including the associated relation in the database.
- Make use of available resources wherever possible to reduce *cost* (such as freely-available software tools or people prepared to give their time to data collection).

4. Pre-development: tools

Initial development of the EDBS was impromptu and informal. Effects were 'collected' in a manner analogous to the way a novice stamp collector takes his first steps in acquiring a collection. At first the process was conducted on a whim, with no attempt to incorporate structure in either the process or the data itself. A Microsoft Excel spreadsheet was used to hold the growing collection of effects and provided methods for sorting and collating them. The same spreadsheet was subsequently extended to provide a simple user interface enabling queries to be submitted and results displayed.

This precursor to the EDBS demonstrated the usefulness and possibility of a free-to-use effects database – albeit one that would take a considerable amount of time and effort to create. The bulk of this work would be required for data collection and much of this took place before the start of the formal development of the EDBS. Effects data and query-effect relations were collected and identified as and when free time was available, a few hours here, a few hours there, over a period of some years.

Once the amassed data had grown to a size that made the spreadsheet-based version of the database useful for practical problem solving, it was time to move to more formal development.

5. Initial development: super-system

5.1. Database trials during development

Two versions of the database were used in trial form during development:

- The original spreadsheet-based precursor system
- The emerging EDBS (running offline using a local web server)

The original spreadsheet-based precursor version of the database was used during development to assess the effectiveness of the Ideality-driven minimalist design in a real-world environment. Initially it was exposed to TRIZ trainers/facilitators at Oxford Creativity and subsequently to selected Oxford Creativity clients for feedback.

5.2. Knowledge and Awareness

From these informal trials it was realized that the effects reported by the database to a user could be categorized in terms of the user's knowledge and awareness of those effects at the time the query was made and in relation to the problem at hand. In this context knowledge and awareness are defined as follows:

- Knowledge : possession by the user of knowledge of the effect and/or knowledge of the Query-Effect Relation for the current query
- Awareness : awareness by the user that Query-Effect Relation is relevant to the current query

The distinction here between knowledge and awareness is important. A user may possess knowledge but not be consciously aware of that knowledge at a particular moment of time – such as when it is needed to solve a problem at hand. Without that awareness the knowledge is useless (on this particular occasion). The role of an effects database is not only to bridge gaps in knowledge, but also to bridge gaps in awareness - and in the context of a particular user attempting to solve a particular problem at a particular time. The distinction between knowledge and awareness is illustrated in Figure 2.

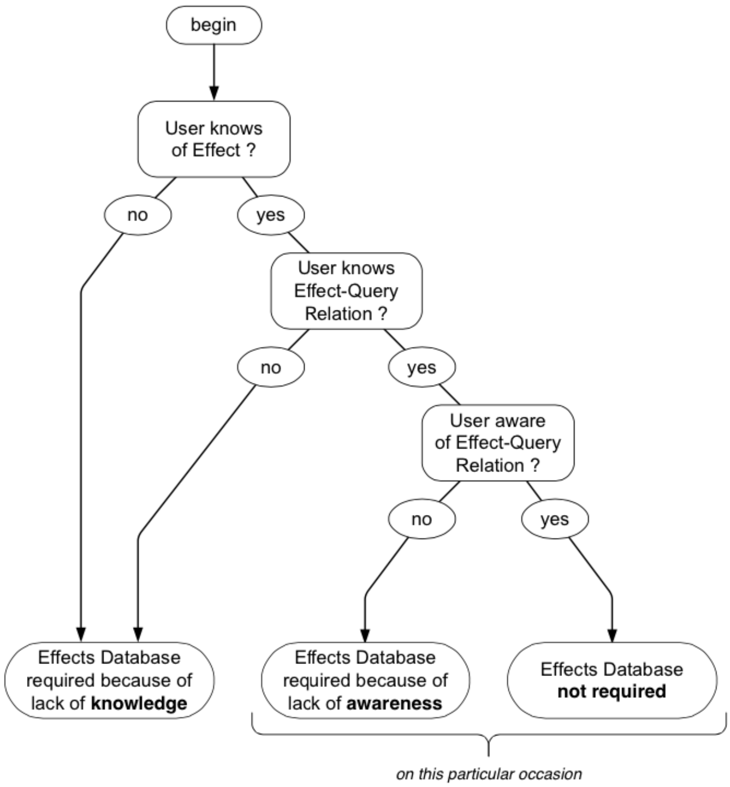

Fig. 2. Graph of user knowledge and awareness states

Use of development versions of the database highlighted that in many cases the usefulness of the database was in providing awareness of knowledge of effects, especially for queries that generated a large number of reported effects. So, for example, the EDBS (at the time of writing) generates over 200 results in response to the query: Function: Move Solid (i.e. "How to move a solid?"). Most of these are commonly known, especially to those from a scientific or engineering background. However, asked to name ways of moving a solid, most individuals working alone and unaided are unable to suggest more than 20-40 suitable effects. Yet, when the results from the database are revealed, very few of the suggested effects fall into the users' "not known" category. Although these "not known" results are an important output of the database, many more potentially useful effects fall into the 'known but not aware of' category. These observations resulted in:

- A realization that a significant benefit of using an effects database is providing the user with awareness of knowledge that he/she already possesses. This is analogous to (although not the same as) psychological inertia in that the user has a subconscious barrier hindering his/her efforts to solve the problem being tackled.
- A realization that much of the useful content of the database – especially in terms if the Query-Effect Relations, could be generated without the need for specialized knowledge – provided that the person(s) compiling the data was somehow immune from the 'awareness blindness' effect. This led to the development of a systematic approach to adding Query-Effect relations to the EDBS described in Section 6.3.
- A re-casting of the ideal outcome of the EDBS in terms of awareness rather than knowledge:

"The user is aware of all effects relevant to the solution of the problem at hand, without cost or any other restriction"

6. Initial development: database and development process

6.1. Database implementation principles

The implementation of the EDBS was based upon the following principles:

- A client server-solution with the database hosted on a web-server and accessed via a standard web browser without the need for any add-ons or plug-ins
- Use of open standards wherever practical
- Tools and standards should be freely available, well established and with good prospects for long-term availability
- User interface to be data-driven – i.e. expansion of the database should, wherever possible, not necessitate a change to the scripts responsible for presenting the client-side user interface

6.2. Database structure

6.2.1. Queries and query-effect relations

The heart of the database is the Query-Effect Relations Table. This is principally a linking table that holds the many-to-many relationships between Queries and Effects.

The query part of a Query-Effect Relation is composed of three components:

- A Mode, which defines the general nature of the query and the relationship between the Task and Target parameters (see below)
- A Task, which forms the first part of the query and identifies the nature of the task component of the query
- A Target, which forms the second part of the query and identifies what the task will operate upon

Examples of queries:

If the user's query is "How to move a liquid?", then:
The Mode is "Function" (the query relates to performing a function) The Task is "Move"
The Target is "Liquid"

If the user's query is "How to increase temperature?", then:
The Mode is "Parameter" (the query relates to operating on a parameter) The Task is "Increase"
The Target is "Temperature"

Thus an entry in the Query-Effect Relation table consists of these fields:

- Mode (defining the type of query)
- Task_ID (defining the task component of the query using a reference into the Effect Table)
- Target_ID (defining the target component of the query using a reference into the Effect Table)
- Effect_ID (defining the effect using a reference into the Effect Table)
- Usage_Note (optional text providing the user with an explanation of how or why the effect is relevant to the query

6.2.2. Formal structure description

The Effects Database structure is comprises five tables:

- Effects Table
- Modes Table
- Tasks Table
- Targets Table
- Relations Table

In addition a sixth table (the Query Log) is used to hold a record of the queries made to the database.

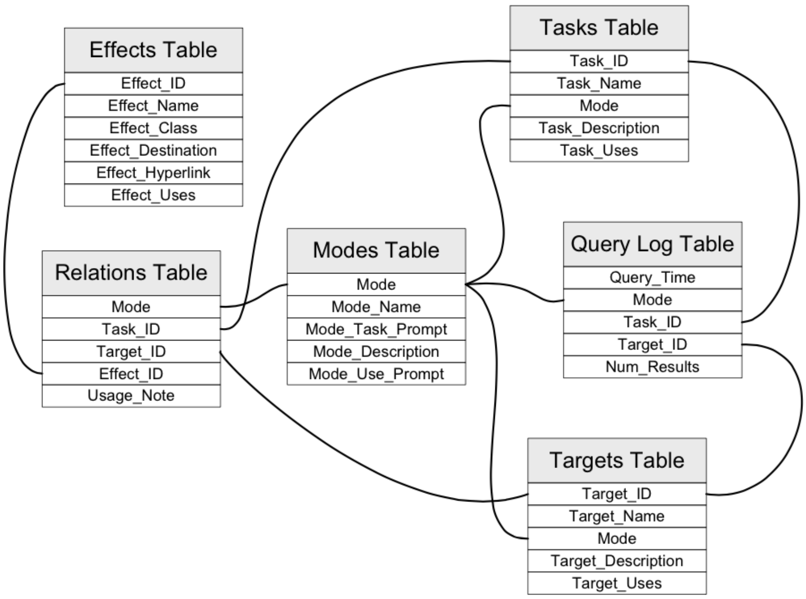

Fig. 3. Database structure diagram

More details of the database tables can be found in Appendix A.

It is recognized that this is very much a minimalist approach to the database design and reflects the database's original implementation as a spreadsheet as well as an the underlying philosophy of delivering sufficient functionality to make the database useful whist reducing the overhead of populating and managing it. The database design also reflects the data-driven approach to the user.

6.3. Data collection

Three types of data collection are required to support population and updating of the EDBS corresponding to the main types of data:

- Effect Data Collection
- Query Data Collection
- Relations Data Collection

The number of database table entries in the EDBS (as of July 2012) gives an indication of the relative magnitude of the data collection task for each data type:

Table 2. Summary of EDBS Table Sizes

Data Type	Number of Database Entries in EDBS (July 2012)
Queries	350
Effects	887
Query-Effect Relations	17028

This suggests than an efficient method of collecting and including Query-Effect Relations data is especially critical to the overall efficiency of populating and maintaining the database.
It was noted earlier in section 5.2 that the relevance of an effect to a query is usually clear once a user has both knowledge and awareness of the effect. Thus it is possible for a person adding a new effect to the database to overcome the problem of effect 'awareness blindness' by deliberately and systematically considering the applicability of the effect to every query in the database using the process shown in Fig. 4.

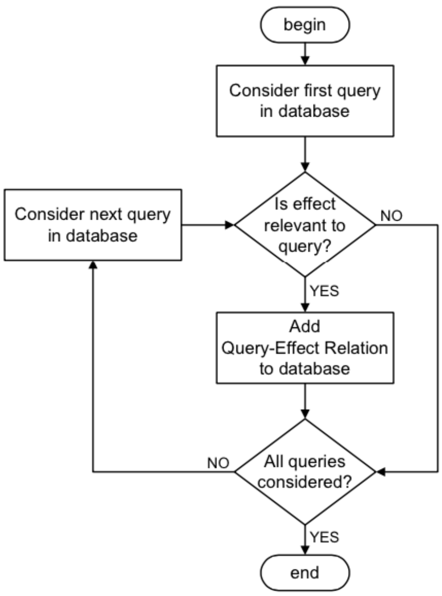

Fig 4. Process for adding a new effect to the database

Similarly, when adding a new query, a systematic process is used to consider each possible combination of that query with each effect in the database, as shown in Fig 5.

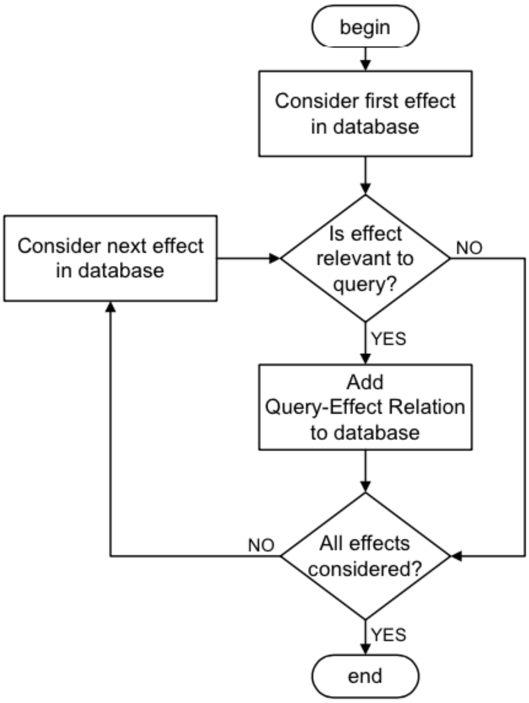

Fig 5. Process for adding a new query to the database

7. Initial development: tools

7.1. Tools: database build

Database build tools were selected against two criteria:

- Low cost of ownership (ideally free to use)
- Long-term availability

The tools selected were:

- Database Engine: MySQL
- Client side style sheet language : CSS (Cascading Style Sheets) (http://www.w3.org/Style/CSS/)
- Server side scripting language: PHP
- Scrip Editor: Arachnophobia
- Local web server: Apache HTTP Server
- Local web server distribution kit (Apache, MySQL and PHP) for use on development system: XAMPP
- FTP Client (used up upload scripts to the online web server): Filezilla

7.2. Tools: data collection

A Microsoft Excel spreadsheet is used for data collection. Embedded VBA (Visual Basic for Applications) macros are used to:

- Manage database sheets that act as interactive data entry forms to facilitate the systematic consideration of all query-effect combinations in the compilation of Query-Effect Relations.
- Export data in the form of text comma separated value (CSV) files

7.3. Tools: database publishing

A suite of VBA macros were created to export the data from the same Excel spreadsheet used for data collection. Data is exported as text files in comma separated value (CSV) format. The files are then uploaded to the database web server using myPHPadmin.

8. Future-development: super-system

8.1. Usage profile

Formal usage logging of the EDBS system was started on 9th June 2012. As of 25th July 2012 (i.e. over a period of 40 days) a total of 873 queries had been submitted to the EDBS, an average of nearly 22 per day. Patterns in the query log indicate that a systematic attempt to harvest the database may have taken place during that time. Adjusting for this event the number of normal queries was 523 giving an average rate of 13 queries per day. It is anticipated that this rate of usage may increase as the existence of the database becomes better known.

The most popular queries, based upon an analysis of the query log, are Function: Move Solid and Function: Move Liquid which account for 4.3% and 3.9% of the total number of queries respectively.

8.2. User feedback

Informal feedback has indicated that users would like to see facilities for filtering results, particularly in the case of those queries such as Function: Move Solid which (as of July 2012) generates in excess of 200 results.

Proposals for filtering under consideration include:

- Filtering between pure effects (such as evaporation) and applications of effects (such as a heat pipe)
- Filtering according to physical scales at which effects can be practically used – e.g. atomic, nanometer, meter, kilometer etc.

9. Future-development: database and development process

9.1. Additional query modes under consideration

A number of additional query modes are under consideration:

- An energy-conversion query mode. Query example: "How can mechanical energy be converted to heat energy?"
- A separation principle query mode, for use when solving physical contradictions. This would answers queries related to separation according to selected parameters, such as time, position, temperature etc.

9.2. Multi-language support

There are no plans to present the EDBS in languages other than English. It is acknowledged that this may restrict some potential uses of the EDBS – and hence falls short of the Ideal Outcome.

9.3. The inevitability of incompleteness

The EDBS can never be complete. The task of collecting and compiling the Query-Effect relations for all scientific and engineering knowledge, even for a finite number of queries is arguably impossible. The problem is made far worse if all possible useful queries were to be included and worse again if the ever- expanding scope of scientific and engineering knowledge is accommodated.

9.4. Technical evolution of the system

In many respects the EDBS is a technological dead-end that will be rendered obsolete by systems that are more advanced in evolutionary terms. Strong candidates to render the EDBS obsolete by providing a superior free-to-use or very-low-cost effects database include:

- A crowd-sourced solution, analogous to Wikipedia, which would allow an extensive (and potentially unrestricted) community of nominally unpaid contributors to populate and extend an effects database system.
- An extension of a free-to-use search engine, such as Google, including facilities for automated mining/harvesting of effects, queries and relations from on-line data sources (such as patent databases).
- Free-to-use or very low-cost versions of existing commercial products in which required revenue product would be raised from sources other than the user.
- An 'App-style' marketing model in which the sales volume is sufficiently high to make the cost to each individual user insignificant.

10. Future development: tools

A broader and more collaborative data collection system is being considered. This would enable a larger community of people to contribute to data collection and also provide a stepping stone towards developing a crowd-sourced data collection system.

11. Conclusion

The ultimate assessment of the usefulness and/or Ideality of the EDBS should be left to users of the system. Furthermore, it is difficult to both define and evaluate a metric suitable for comparing effects databases. Reasons for this include:

- There is considerable variation in the nature and structure of the database content.
- Many aspects of an effects database are hard to quantify, such as the quality of effect descriptions.
- Computer-implemented databases generally restrict users to problem-solving queries (e.g. How to move a liquid?") rather than ones needed to access the database scope (e.g. "Tell me everything you know").
- Effects databases form part of a wider problem solving process that incorporates human beings (with all the uncertainty that that implies) interacting with a potentially infinite number of possible problems.

Despite this, and whilst acknowledging the limitations of a simplistic 'numbers based' measure of database content, a crude comparison has been made between a number of existing free-to-use effects databases and the EDBS (as of June 2012) in terms of:

- Number of Effects in the database

- Number of Queries allowed by the database
- Number of Query-Effect Relations in the database

The following databases were analyzed:

- The paper-based database published as Appendix 3 of Creativity as an Exact Science (CAAES) [1] which appears in an almost identical form as Appendix 3 of Engineering of Creativity [4]
- The paper-based database published as “Index of Physical Effects” in Tools of Classical TRIZ (TOCT) [2]
- The paper-based database published in Chapter 15 of Hands-On Systematic Innovation (HOSI) [3] which appears to have the same or very similar content (in terms of numbers of effects, queries and relations) to the internet-based database available at the Creax website (www.creax.com) [5]
- The internet-based database available at TRIZ Korea website (www.triz.co.kr) [6]
- The EDBS (the database described in this paper)

The results are summarized below in Table 3.

Table 3. Results of analysis of selected free-to-use effects databases

Effects database	Effects	Queries	Relations
CAAES	130 (approx.)	30	224
TOCT	199	30	296
HOSI / www.creax.com	400 (estimate)	222	1000 (estimate)
www.triz.co.kr	141	51	398
EDBS	887	350	17028

These numbers suggest that the EDBS may have been successful in providing a more useful free-to- use effects database. However, as noted above, numbers cannot tell the whole story. The reader is encouraged to try the EDBS (at http://ftp5.dns-systems.net/~wbam2244/EDB_Welcome.php or via the Oxford Creativity website at www.triz.co.uk) and make his/her own judgment.

Appendix A. Details of EDBS Database Tables

A.1. Effects table

The Effects Table holds data defining and describing each effect. The Effects Table has six fields as follows:

Table 4. Effects Table Fields

Effects Table Field	Description	Type
Effect_ID	A unique index to identify the effect.	Unsigned Integer
Effect_Name	The name of the effect.	Text
Effect_Class	The class of the effect. Currently only two classes are used: Effect and Application.	Text
Effect_Description	A short text description of the effect.	
Effect_Hyperlink	A hyperlink to an internet-based resource that provides additional information about the effect.	Text
Effect_Uses	The number of uses (Relations) of the effect included in the database (strictly speaking this is redundant information as it can be obtained by querying the database).	Unsigned Integer

A.1. Modes table

The Modes Table holds data defining and describing each of the query modes of the database. The Modes Table has six fields as follows:

Table 5. Modes Table Fields

Modes Table Field	Description	Type
Mode	A unique character identifying the usage mode of the database.	Text
Mode_Name	The name of the usage mode, e.g. "Function".	Text
Mode_Task_Prompt	A short text prompt that informs the user of the type of task used in this mode, e.g. "Required Function".	Text
Mode_Target_Prompt	A short text prompt that informs the user of the type of target used in this mode, e.g. "Object".	
Mode_Description	A short text description of the mode suitable for guiding the user when selecting which database mode to use. In practice this has been used to hold an example of the usage of the mode, e.g. :"For example: Move a Liquid".	Text
Mode_Use_Prompt	A short text prompt that provides the user with information when using the database in this mode, e.g.: "Select an Operation and the Parameter on which the Operation is to be performed".	Text

A.2. Tasks table

The Tasks Table holds data defining and describing each of the query tasks. The Tasks Table has five fields as follows:

Table 6. Tasks Table Fields

Tasks Table Field	Description	Type
Task_ID	A unique character identifying the task.	Unsigned Integer
Task_Name	The name of the task, e.g. "Move".	Text
Mode	The mode in which this task is used.	Text
Task_Description	A short text description of the task that may be used to guide the user.	Text
Task_Uses	The number of relations that use the task (strictly speaking this is redundant information as it can be obtained by querying the database).	Unsigned Integer

A.3. Targets table

The Targets Table holds data defining and describing each of the query targets. The Targets Table has five fields as follows:

Table 7. Targets Table Fields

Targets Table Field	Description	Type
Target_ID	A unique character identifying the target.	Unsigned Integer
Target_Name	The name of the target, e.g. "Liquid".	Text
Mode	The mode in which this target is used.	Text
Target_Description	A short text description of the target that may be used to guide the user.	Text
Target_Uses	The number of relations that use the target (strictly speaking this is redundant information as it can be obtained by querying the database).	Unsigned Integer

A.4. Query-effect relations table

The Query-Effect Relations Table holds data defining and describing each of the query-effect relations. The Query-Effect Relations Table has five fields as follows:

Table 8: Query-Effect Relations Table Fields

Targets Table Field	Description	Type
Mode	The mode for which the relation applies.	Text
Task_ID	The index of the task component of the relation. Used to obtain information from the Tasks table.	Unsigned Integer
Target_ID	The index of the target component of the relation. Used to obtain information from the Targets table.	Unsigned Integer
Effect_ID	The index of the effect component of the relation. Used to obtain information from the Effects table.	Unsigned Integer
Usage_Note	A short text description of how the effect identified in this relation is relevant to the query matched by this relation.	Text

The Query-Effect Relations Table holds *references* to the Mode, Task, Target and Effect data rather than the data itself. In contrast the Usage Note data is held as part of the Query-Effect Relation record as it is potentially unique to each relation.

A.5. Query log table

The Query Log Table holds information about each query made to the database. The Query Log table has five fields as follows:

Table 9. Query Log Table Fields

Query Log Table Field	Description	Type
Query Time	The date and time on/at which the query was made.	Datetime
Mode	The mode used for the query.	Text
Task_ID	The index number of the task specified for the query.	Unsigned Integer
Target_ID	The index number of the target specified for the query.	Unsigned Integer
Num_Results	The number of results returned by the database.	Unsigned Integer

References

[1] Altshuller, G.S. ,*Creativity as an exact science. Studies in Cybernetics:5* (trans. Anthony Williams), Gordeon and Breach; 1984.
[2] *Tools of Classical TRIZ*. Ideation Internatonal Inc. 1999.
[3] Mann, Darrell L, *Hands-On Systematic Innovation.* CREAX Press; 2002.
[4] Savransky, Semyon D., *Engineering of Creativity*. CRC Press; 2000.
[5] *Creax Function Database*. Web. 30th July 2012. http://www.creax.com/function_database.htm;
[6] *Triz Tool - Effects*. Web. 30th July 2012. http://www.triz.co.kr/TRIZ/frame.html;
[7] Gadd, Karen, *TRIZ for Engineers*. Wiley; 2011, Chapter 4.