Patent Mining (PatMine v0.1) ©

User Manual

PatMine © is developed by Manal Helal, Lecturer in Arab Academy for Science, Technology, and Maritime Transport. This is a working document to develop a user manual to be updated as the development progresses.

Contents

1. Installation: 2
2. FAD Visualisation: 5
3. Annotating using FAD 8
4. Searching & Browsing the PatMine Patents Database 10
5. Comparing the Emerging Design with the patents in the PatMine Database 12
6. Importing from Excel 14
7. Bug Reporting and feature requests 16

1. Introduction

Patents provide a rich source of information about design innovations. Patent mining techniques employ various technologies, such as: text mining, machine learning, natural language processing, and ontology-building techniques among others. A new graph data modelling method is proposed for building a semantic database of patents from functional representations of mechanical designs. The method has several benefits: The schema-free characteristic of the proposed graph modelling enables the ontology it is based on to evolve and generalise to upper ontologies across technology domains, and to specify to lower ontologies to more specific engineering domains. Graph modelling benefits from enhanced performance of deep queries across many levels of relationships and interactions, and provides efficient storage. Graph modelling also enables visualisation libraries to use the graph data structure immediately avoiding the need for graph extraction programs from relational databases. Patent/Designs comparisons is computed by search queries using counting of overlaps of different levels and weights. This work has produced the PatMine SolidWorks Add-in ©, which compares annotated CAD designs with patents and highlights overlapping design concepts. The annotation extracts a functional analysis and its structure is represented as geometric features interactions. Additional features such as full text search and semantic search of the PatMine patents database are available, and graph analytic methods and machine learning algorithms are enabled and will be implemented as plug-ins in the near future.

The Patent Mining approach presented in this project enable the designer to check the innovation score early enough in the design process. The published patent's annotations extracts the Functional Analysis Diagrams (FAD models), using fixed naming convention such as Reconciled Functional Bases from the literature (RFB). These annotations are visualised graphically, and similarity scoring is presented to the designer. The details of the overlap in the emerging design is presented to the user along with the corresponding patent documents for further checks and redesign. The cycle continues until an acceptable innovation score is reached, and overlap with existing designs are addressed properly based on the design knowledge domain. Figure 1 summarise this cycle.

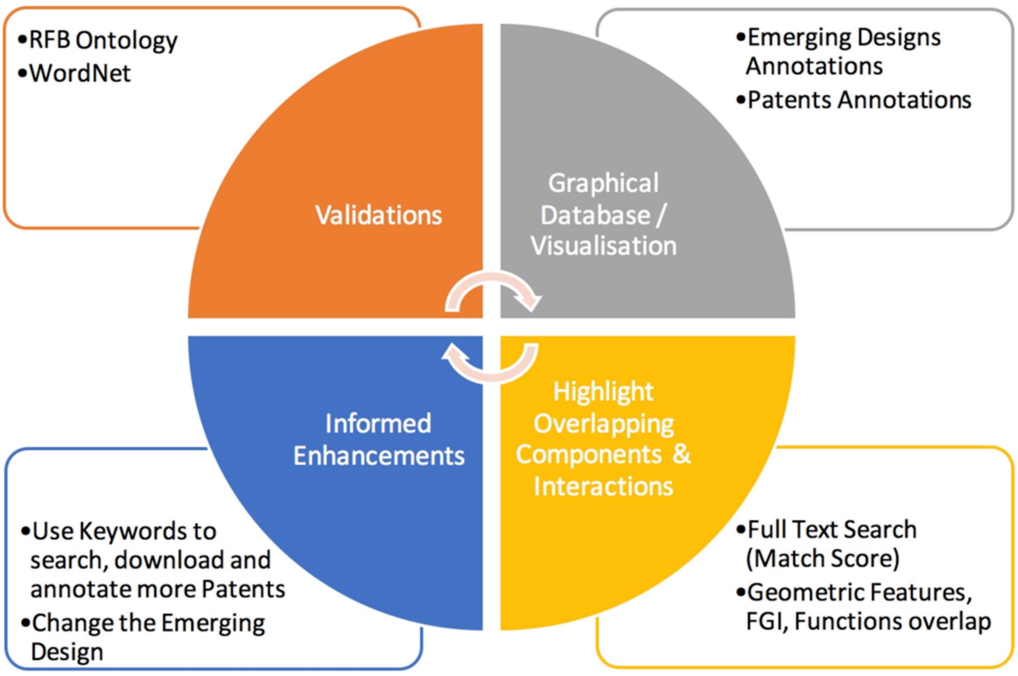

Figure 1: Design Innovation Cycle

The annotation of the emerging designs and the published patents need to include the products included in the design, the claims, the functions achieved by the design, the geometries included in the design, and Functional-Geometric Interactions (FGIs) that achieve the announced function. These requirements are captured in Figure 2.

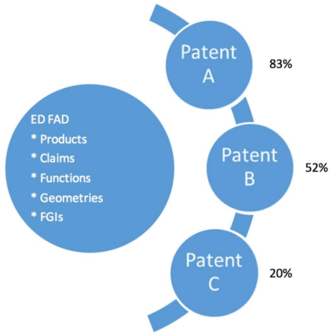

Figure 2: FAD Annotation Requiremenets for ED (Emerging Design) and Patents

The database contents are a list of FAD annotations for emerging designs and published patents, and the visualisations of these annotations, similarity scoring, and identification of the overlapping contents. The scoring is an iterative search of various components of annotations of the ED and the patents in the database with assumed initial weights. This is captured in Figure 3.

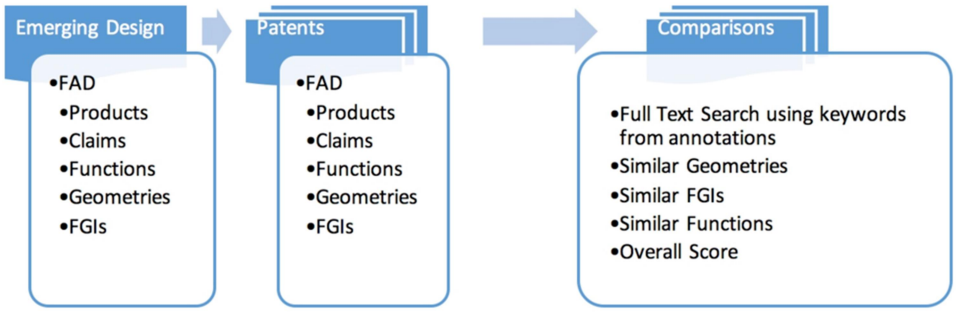

Figure 3: Scoring ED to each patent in the database.

The technologies integrated in the production of PatMine are described in Figure 4.

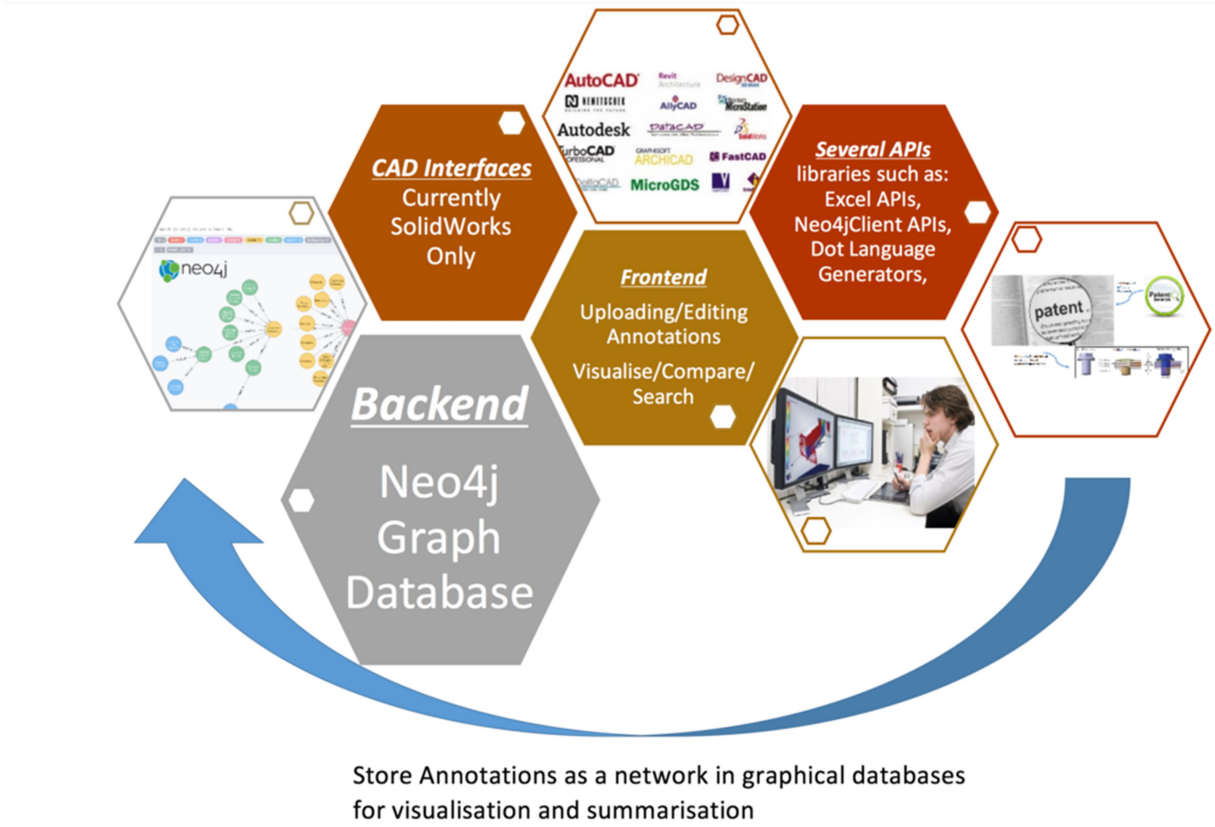

Figure 4: PatMine Components

2. Installation:

The tool is available for download as an add-in for Solid-Works as an example CAD tool, or as a stand-alone application that can interface with a running instance of Solid-Works. It is developed using Visual Studio C# APIs. The tool is a front end that connects to the database backend which is Neo4j graph database. The backend is a schema free graph database, and similarly the front end was built to maintain this feature. Although the main entities are: 1) patent or emerging design, 2) product, 3) claims, 4) geometries, and 5) FGIs. The first label given to a node, determines its type. In this case the basic node types are the five main entities just mentioned. Higher level abstractions are included as extra labels, such as geometries conceptual higher abstracted types. Columns headers in the excel sheets to upload to Neo4j are used as properties names, and their row contents as their values. This enables dynamic extraction of information as our knowledge of the patent document structure increase, or as more domain experts enrich the graph ontology with more object types, relationship types and taxonomy.

The tool adds a toolbar that contain button to invoke the functionalities offered by the system. First: an interface to define the FAD model components for the emerging design. Second functionality is to search and browse the FAD models for the patents stored in the database. A third functionality is to compare the emerging design's FAD with all patents' FAD and display the comparison results, in terms of charts of match ranks, overlapping geometries and interactions, corresponding patent document retrieval with the FAD annotations highlighted, and image thumbnails of the important drawing in the patent document. A fourth functionality expose all environment variables and system settings to be updated by the user to enable dynamic behaviour of the system as much as possible.

3. FAD Visualisation:

The Neo4j graph database is used in the backend. Neo4j Server browser employs a simple force-directed graph drawing algorithm that uses character co-occurrence as the force guiding a physical simulation of charged particles and springs. This algorithm places related characters taken for example from node labels and properties' values in closer proximity, while unrelated characters are farther apart. An Example corkscrew design and Cork–screw apparatus Patent as visualised in the Neo4j Server Browser is shown in Figure 5 and Figure 6 respectively.

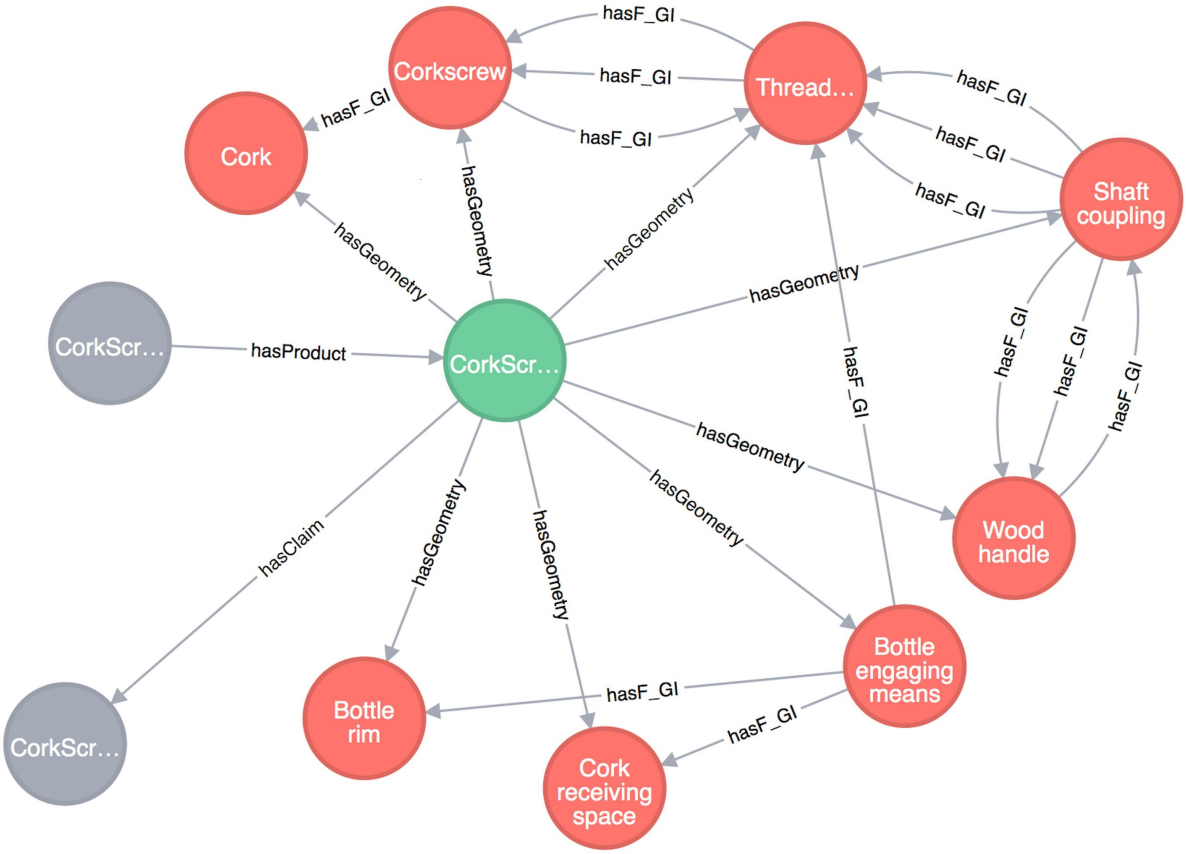

Figure 5: Corkscrew Design FAD visualisation in Neo4j Server Browser

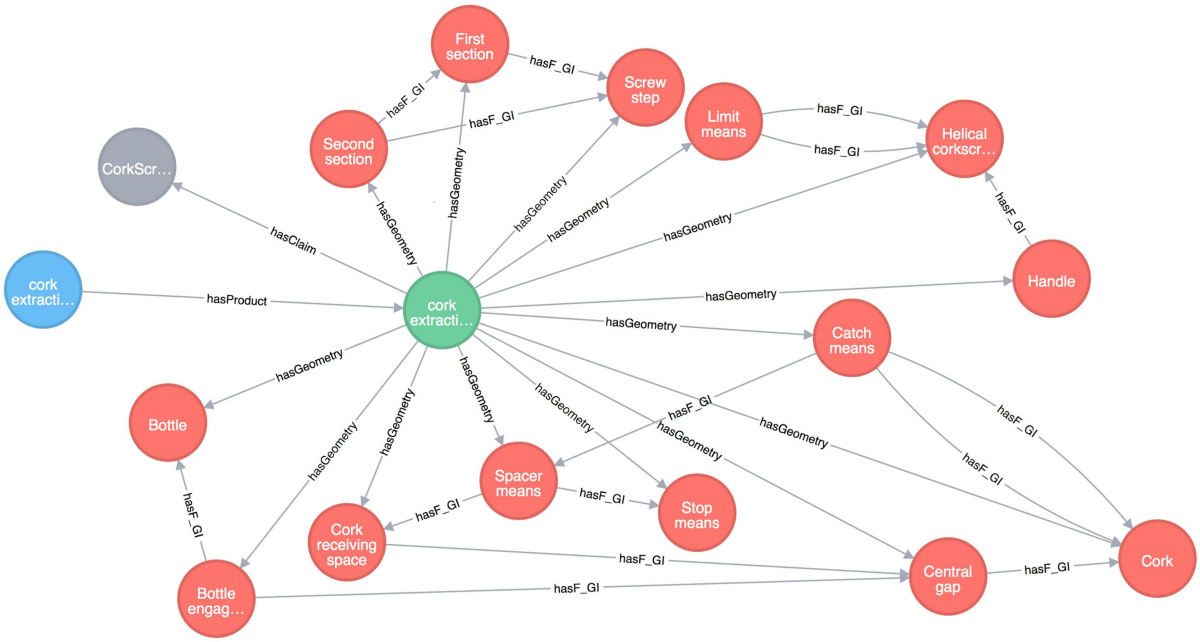

Figure 6: Corkscrew Apparatus Patent FAD visualisation in Neo4j Server Browser

The visualisation from inside the tool is generated using the dot engine. The corkscrew design vs. the corkscrew apparatus patent examples used above are visualised in PatMine SolidWorks Add-in © screens in Figure 7 and Figure 8 respectively.

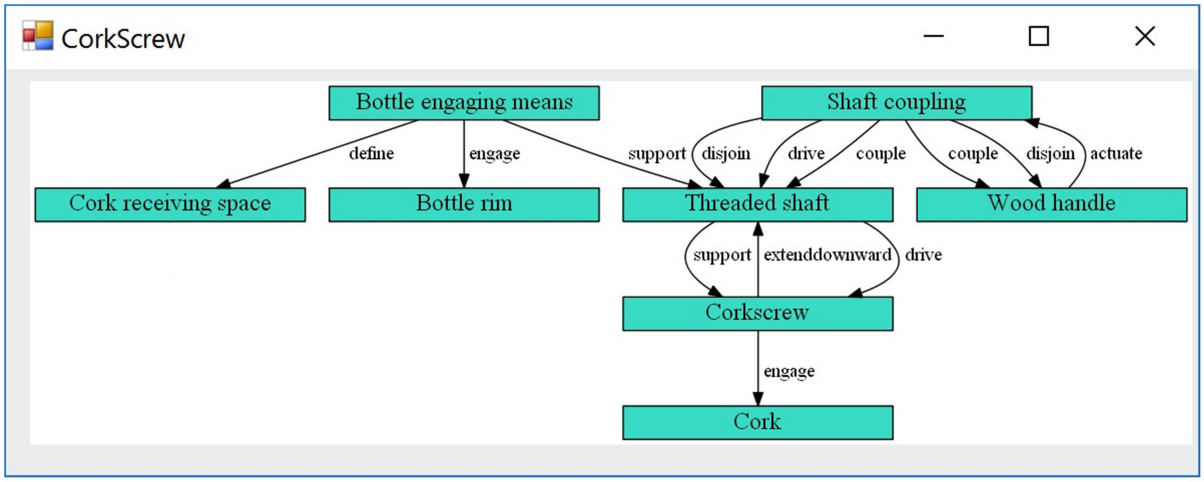

Figure 7: Corkscrew FAD visualisation in PatMine SolidWorks Add-in ©

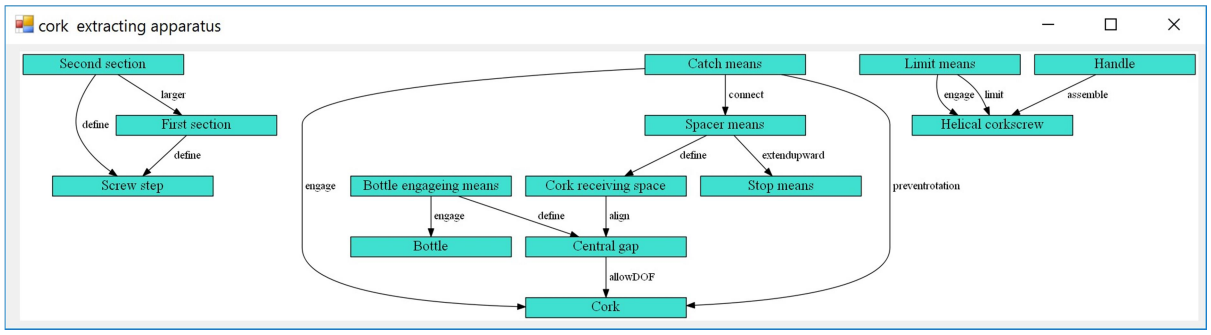

Figure 8: corkscrew apparatus patent FAD visualisation in PatMine SolidWorks Add-in ©

4. Annotating using FAD

This screen in Figure 9 allow the designer to annotate the emerging design using the FAD annotation model. This is divided in four tabs: 1) the design and product names and the claims it provides; 2) The functions provided by the design, 3) the list of geometric features the design contains; and 4) the list of FGIs between the defined geometries that achieve the intended functions. The annotations can be imported from a formatted excel workbook.

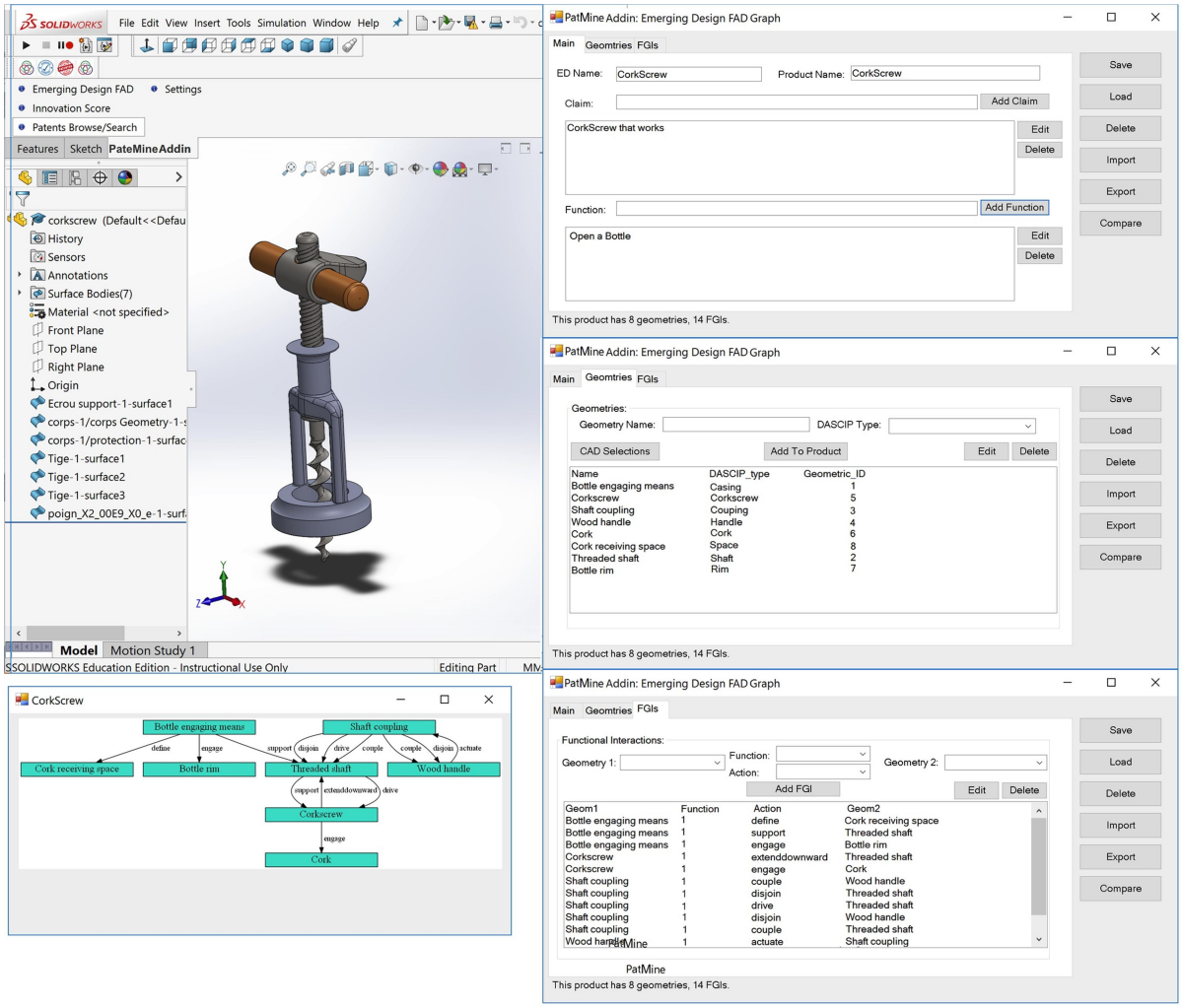

Figure 9: PatMine SolidWorks Add-in © Corkscrew design FAD model Definition screens and visualisation

Whether SolidWorks is running or not, you can start the program standalone as shown in Figure 10 to traverse the stored annotations for emerging designs and patents. The program has functions for adding, deleting, updating the existing annotations, visualising the FAD diagram for comparisons and identifications of overlapping regions.

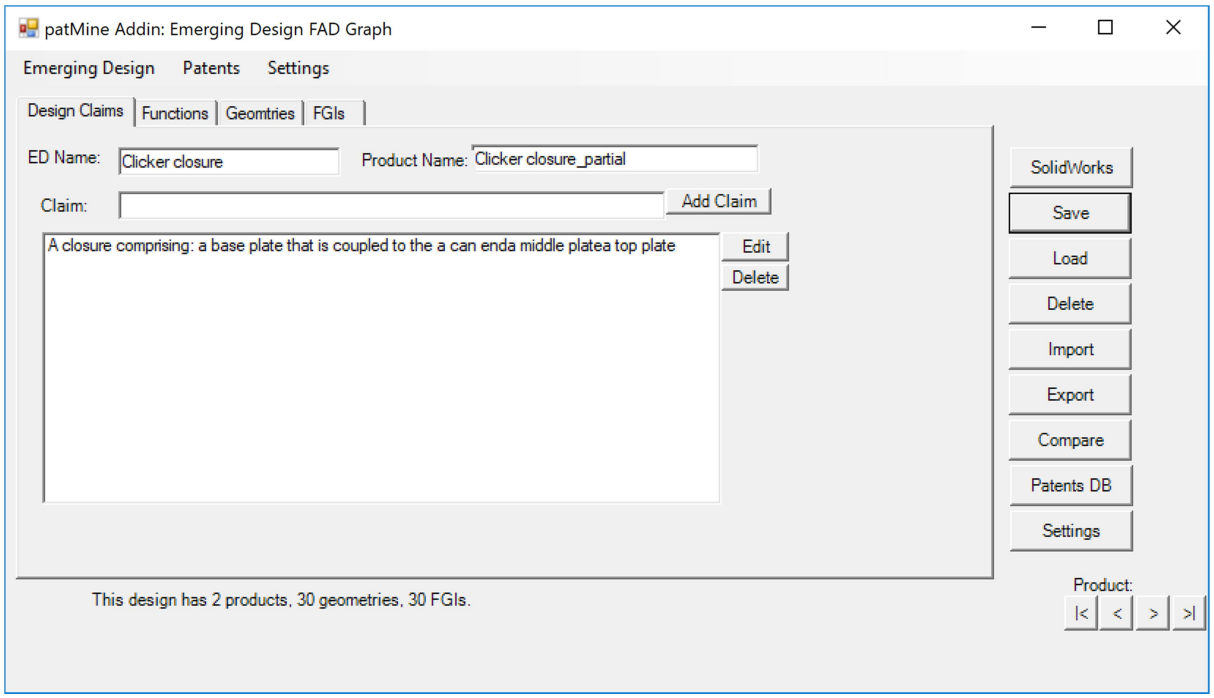

Figure 10: Standalone version of PatMine

5. Searching & Browsing the PatMine Patents Database

This screen enable the user to search and browse the FAD models stored in the patMine patents database. Three types of search queries are currently implemented. The first is a semantic search query that accepts query keywords to search in the title, product, function, action and geometry values. A second full text search is used to generate a dynamic query as mentioned in section five. A third query option is to enable experienced users to write their own cypher statements. Figure 11 illustrates the three query options user interface. The resulting patents are displayed graphically with zooming functionality to investigate the interactions. Navigation buttons (First, Previous, Next, and Last) are used to move around between patents in the Database. Double-clicking FAD image shows a thumbnail of the patent's FAD stretched to be seen completely in smaller area such as the one shown in Figure 8. This user interface invokes the first user interface to update the FAD model for every patent in the database to change its model contents, add and/or remove patents. Figure 12 shows the cork extracting apparatus patent displayed as a result set for an example search.

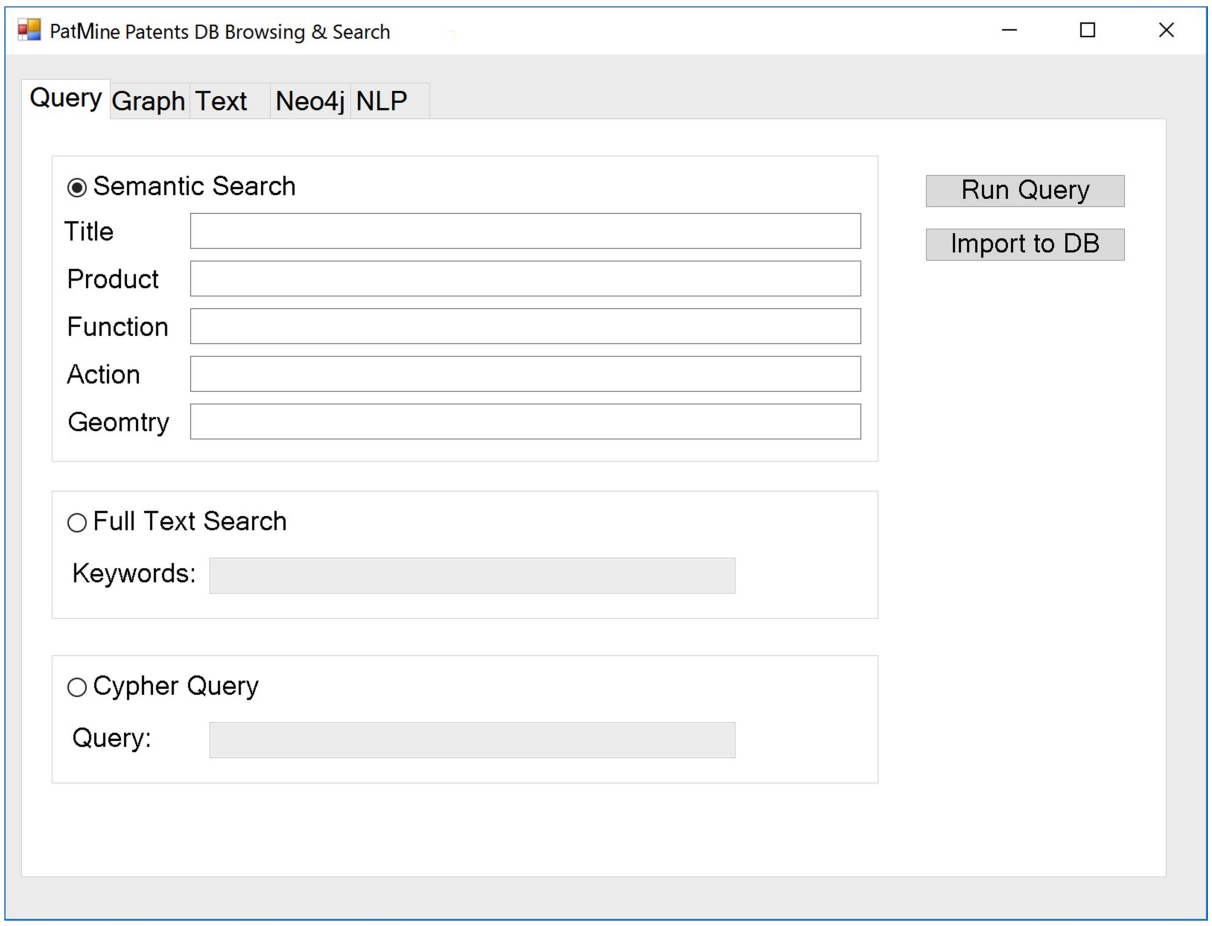

Figure 11: PatMine SolidWorks Add-in © Patents Database Query Generation

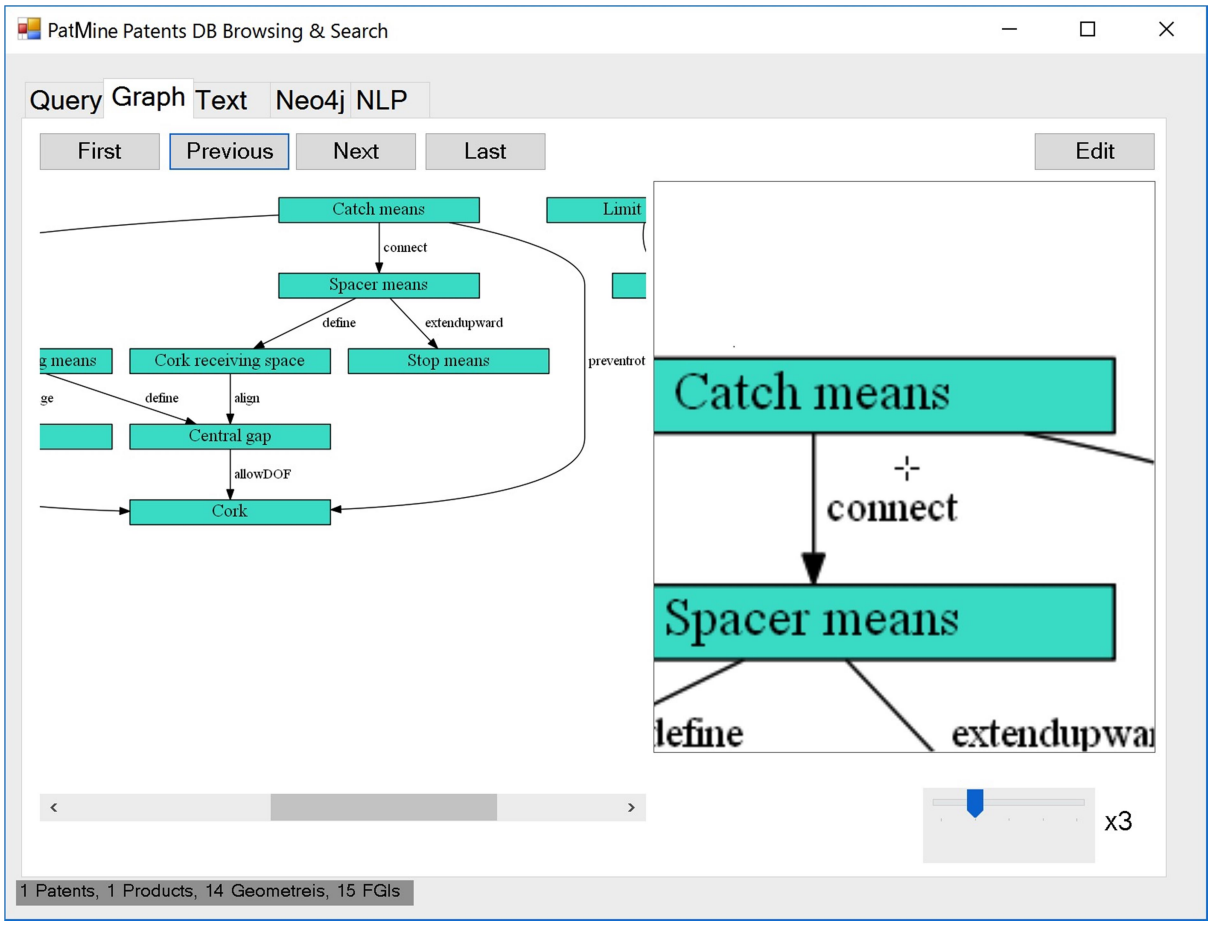

Figure 12: PatMine SolidWorks Add-in © Patents Database Query Results Interface

6. Comparing the Emerging Design with the patents in the PatMine Database

A third functionality is to compare the emerging design's FAD with all patents' FAD and display the comparison results, in terms of charts of match ranks, overlapping geometries and interactions, corresponding patent document retrieval with the FAD annotations highlighted, and image thumbnails of the important drawing in the patent document. The illustration in Figure 13 shows some screens of the scoring results for the corkscrew example.

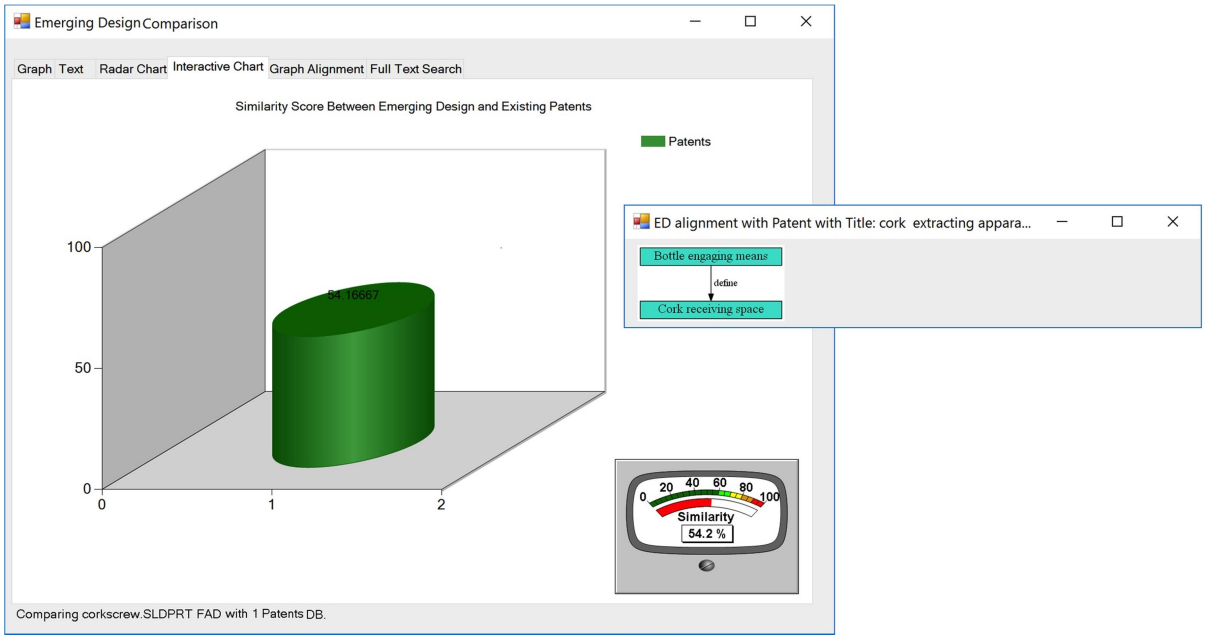

Figure 13: PatMine SolidWorks Add-in © Corkscrew example design scored against a 54% similar Patent, and overlapping interaction identified

Another example is shown in Figure 14 and Figure 15.

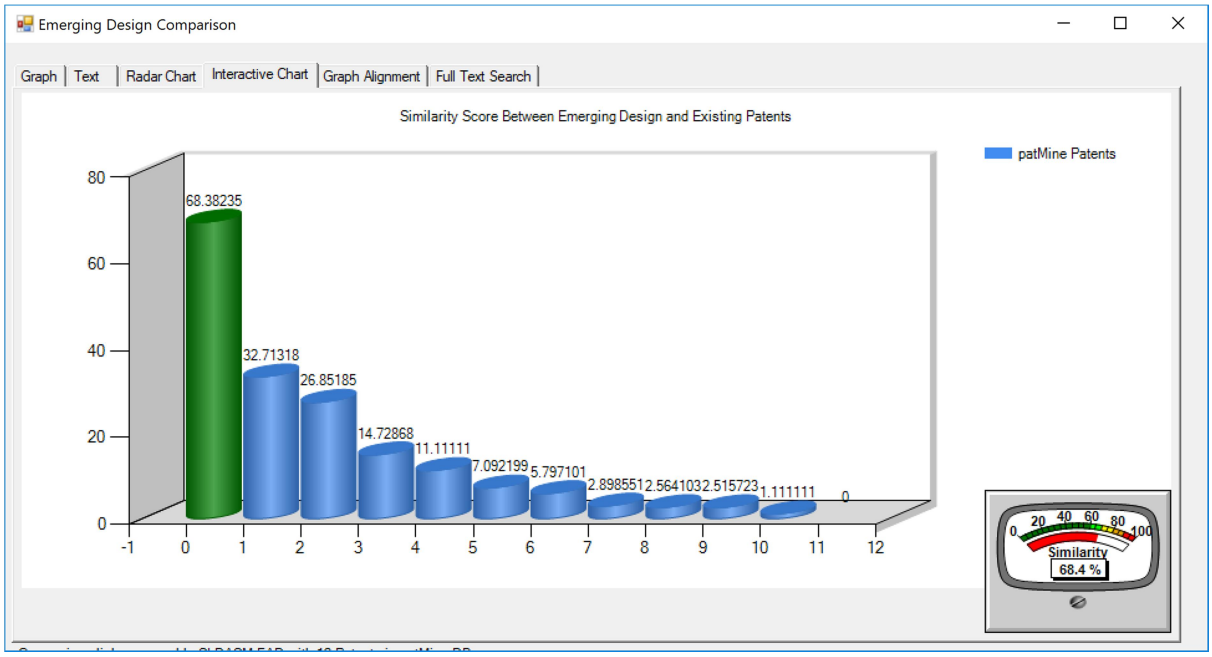

Figure 14: Scores for ED with all patents in DB

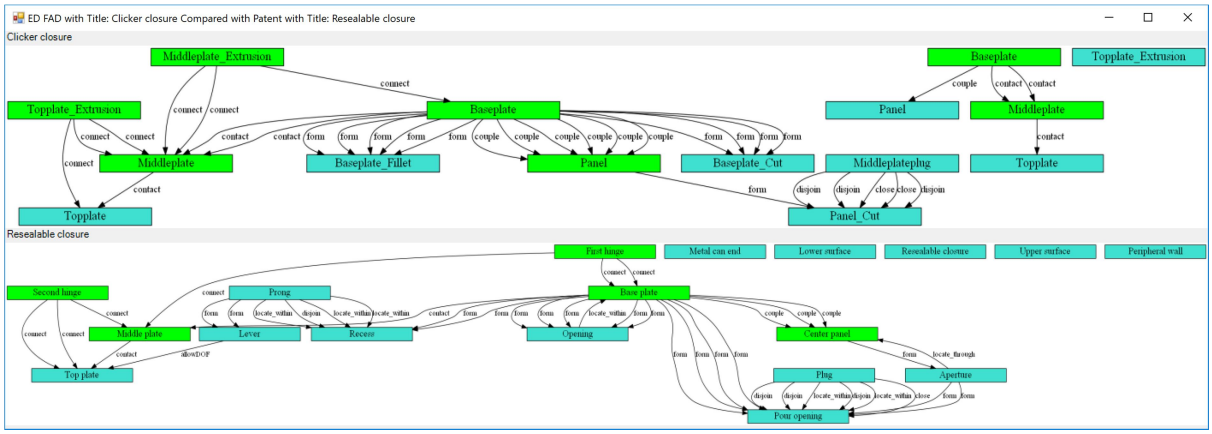

Figure 15: Identifying overlapping Geometries and interactions, and highlighting them from one FAD annotation to another

A fourth functionality exposes all environment variables and system settings to be updated by the user to enable dynamic behaviour of the system as much as possible, such as path to patent documents, path to dot engine. The names used in files (patents, annotated patents, thumbnails) need to be added exactly the same in the properties of patent and emerging design nodes in the DB. The database is populated from excel sheets describing the FAD annotations uploaded directly to the database back end. Direct loading of CSV into the Neo4j server using cypher statements is performed sometimes, or through the PatMine SolidWorks Add-in © User Interface. The Neo4j database backend is available on a single instance server, to which people can connect to and upload their FAD modelling of patents, and emerging designs. An interface to change the ontology taxonomy, glossaries, and relationships will be integrated soon to enable voting for the best way to model the various domains. If the database grows, distributed deployment of Neo4j on a reasonably sized cluster will be investigated. These future steps will enable a collaborative ontology building approach to widen our understanding of the upper ontology across all design engineering domains, and a lower and more specific ontology for sub domains.

7. Importing from Excel

The database is populated from excel sheets describing the FAD annotations uploaded directly to the database back end. Direct loading is performed sometimes, or through the PatMine SolidWorks Add-in © User Interface.

The file should contain 6 sheets. First sheet must contain the following order of column headers to describe the patent, with rows describing the contents. Extra Columns can be added after the listed required ones, for any extra desirable property to maintain, given that the first row will contain the property name and the following rows will contain the values for the corresponding patents:

Table 1: Patent Sheet Contents & Order of Columns

Patent_ID	Name	filename	thumbnail	Annotated File
"EP 0 344 377 B1"	"cork extracting apparatus"	EP0344377B1.pdf	EP0344377B1.png	EP0344377B1_annotated.pdf

For an emerging design, the filename needs to be the solidworks file name:

Table 2: Emerging Design Sheet Contents & Order of Columns

ED_ID	Name	filename	thumbnail
1	CorkScrew	corkscrew.SLDPRT	

Please note that to upload to the same database, All IDs belonging to the same entity (such as patent, product, claim, ...etc.) need to be unique across all patents and emerging designs. Such that we don't repeat Product ID 1 for different Patents for example.

The second sheet need to list the products the patent or emerging design describe, and repeat the patent ID for referential integrity, followed by any list of properties as required in extra columns:

Patent_ID or ED_ID	Product_ID	name
1	2	cork extracting apparatus

Table 3: Products Sheet Contents & Order of Columns

The third sheet is to list the claims, and again repeat the product ID for referential integrity and also list any extra properties as extra columns:

Table 4: Claims Sheet Contents & Order of Columns

Product_ID	Claim_ID	name
2	2	CorkScrew with a guide frame provided for guiding a corkscrew into a cork and for limiting down- ward movement of the corkscrew with respect to the bottle.

The fourth sheet is to list functions (repeating product ID, and all extra properties):

Table 5: Functions Sheet Contents & Order of Columns

Product_ID	Function_ID	name
2	2	down- ward movement.

The fifth sheet contain the geometries, again repeating the product ID, and again making sure all geometries across all products and patents have different ID number:

Table 6: Geometries Sheet Contents & Order of Columns

Product ID	Geometric Feature ID	Geometric Feature Name	Geometric Feature Type
2	1	Handle	Handle
2	2	Helical corkscrew	Corkscrew
2	3	Bottle engageing means	Casing

The sixth sheet should contain the FGIs, and also repeat the product ID, and also both geometries IDs then action, FGI:

Table 7: FGI Sheet Contents & Order of Columns

Product_ID	FGI ID	Geometric ID	Functional Interaction	Geometric ID	Function IDs
2	"cork extracting apparatus"	1	assemble	2	1, 2
2	"cork extracting apparatus"	3	engage	14	1
2	"cork extracting apparatus"	3	define	7	2
2	"cork extracting	7	allowDOF	5	

	apparatus"				

Please Note: For all columns string columns such as filenames and names are checked for quotes, and if not there, will be added by the code. For the Extra properties added in every sheet, the default is numeric (no quotes are added), but when the content of the cell contain a space, quotes are added and treated as string property.

Please Note: Column Names must not contain spaces!

Please Note: Same entity IDs from the patents file, or emerging design file, can not overlap. They will be uploaded to the same database at the end. When they overlap, it will mean it is the same entity and no new node will be created, just a relationship to the existing node. An overlap in geometries IDs between patents and other patents or emerging designs, will make an interaction meant to be between two geometries in one patent appear in other patent or design where only the overlapping geometry existed but not the interaction! In the automatic generation of diagrams, the added interaction will drag a foreign geometry to the design, added only for the interaction, and it will not be formatted equally like other geometries that genuinely belong to the design and was properly formatted. If this weird diagram is generated, review your IDs first thing.

Using the correct order of sheets, and columns, will guarantee successful automatic uploading the emerging design excel sheet from the FAD UI. The patents will be uploaded from a different excel sheet using the PatMine Database search and browse UI. Therefore we need two different excel sheets to fully complete a cycle. The concept of full annotation of the emerging design, can be an extra column, and a radio button in the FAD UI.

8. Bug Reporting and feature requests

The project wiki is maintained in patmine.manalhelal.com. Please contact Dr. Manal Helal (manal.helal@gmail.com) for bug reporting and/or feature requests.