A Digital Ecosystem-based Framework for Math Search Systems

Mohammed Q. Shatnawi
Computer Information Systems Department
Jordan University of Science and Technology
Irbid, Jordan

Qusai Q. Abuein
Computer Information Systems Department
Jordan University of Science and Technology
Irbid, Jordan

***Abstract*—Text-based search engines fall short in retrieving structured information. When searching for x(y+z) using those search engines, for example Google, it retrieves documents that contain xyz, x+y=z, (x+y+z) =xyz or any other document that contain x, y, and/or z but not x(y+z) as a standalone math expression. The reason behind this shortage; is that the text-based search engines ignore the structure of the mathematical expressions.**

Several issues are associated with designing and implementing math-based search systems. Those systems must be able to differentiate between a user query that contains a mathematical expression, and any other query that contains only a text term. A reliable indexing approach, along with a flexible and efficient representation technique are highly required. Eventually, text-based search systems must be able to process mathematical expressions that are well-structured and have properties that make them different from other forms of text.

Here, in this context we take advantage from the concept of digital ecosystems to refine the text search process so it becomes applicable in searching for a mathematical expression. In this research, a framework that contains the basic building blocks of a math-based search system is designed.

Keywords-component; digital ecosystem; math search; information retrieval; text-based search engines; structured information; indexing approach; representation technique.

I. INTRODUCTION

A mathematical expression has many equivalent expressions [1]; this makes the process of searching for a mathematical expression is different than searching for other types of information. For example, the expression x^-1 is mathematically equivalent to 1/x. Traditional search engines do not differentiate between mathematical expression and any other types of information. Google treats both expressions as text-based ones.

In fact, the mathematical expression has certain properties that make it far different from other types of information. Actually, the structure of the mathematical expressions conveys their correct interpretation [2][3][4].

II. PROBLEM STATEMENT

Currently, traditional search engines are not able to search for math expressions or even recognize math notations and symbols. Thus, to search for a certain math expression, users need to consider the following [3]:

- How to enable those search engines to recognize math symbols?
- Do those search engines understand the equivalency in math?
- Do those search engines understand the structure of math expressions?

All of the above need to be considered in order to enable those search engines to satisfy the user needs when he/she searches for math contents as well as other types of contents. The specific needs of users will be investigated in further details in the following sections.

III. RESEARCH GOAL

Building a math-based search system can be achieved using two different approaches. The first approach is to take advantage of the text-based search systems and tailor them to be adequate for math-based search quires. The second one is to build math aware search systems from scratch, based on the new emerging technologies. Either one has its pros and cons [2].

The goal of this research is to design a framework based on digital ecosystem properties [5] [6] to support math search on the web [7]. The proposed framework consists of several components that are needed to support search activities on math-based web data with a high precision. The detailed description of the proposed framework will explain all related issues of math-based search systems.

IV. ACCESSING MATH EXPRESSIONS ON THE WEB

Virtually all searches are text-based [8] [9], thus, there are problems associated with accessing math expressions on the Web. Those problems can be summarized as follows:

- Unless we have an agreed upon technique that should be understood by both users and search engines, a user needs to know the best search terms and the best way to write a query to be used in searching for any mathematical expression.
- When a user searches for a mathematical expression, there would be non-alphabetical symbols that are not understood by current search engines (e.g. Log10x+y^2).

- The same expression can be rewritten in many different, yet equivalent ways (e.g. 1/x and x^-1).
- Text-based search engines do not consider the syntax of a mathematical expression as one of its main features.
- The used approaches to search for equivalent text terms (i.e. thesaurus to search for synonyms) are not feasible for searching for an equivalent mathematical expression.

Relatively speaking, "the text is the only data type that lends itself to a full functional processing" [8].

A. Current Search Engines and Math Search Issues

Text-based search engines cannot search efficiently for different types of mathematical constructs (e.g. axioms, formulas, etc). Mathematical expressions have some distinct properties that make current search engines inadequate to search for such expressions. There are issues that the current search environment has never had to face. Three of them will be mentioned according to what authors of [2] mentioned:

- Searching for a mathematical expression is usually combined with non-alphabetical symbols (e.g. x^3 dy/dx, x**2, etc).
- Different types of mathematical constructs are structured and the structure itself conveys the meaning of these expressions.
- The more challenging issue, is that the same expression can be represented in many different ways. For example, 1/3 mathematically is the same as 3^-1.

V. Mathematical Expression As Search Terms

Mathematical expressions are a distinct type of information. Searching the Web for a mathematical expression is not a well-defined process; the result of the search is unexpected most of the time. The inaccurate result is due to the nature of the mathematical expression search process, which is not based on clear and structured rules. In addition, the available techniques are not applicable to such expressions but they are designed and tailored to work with normal text along with different kinds of documents (e.g. multimedia).

In this paper, the concentration will be on the main three components of the proposed framework, which are:

- The Mapping component
- The Representing component
- The Indexing component

VI. The Mapping Component

Theoretically, a mathematical expression may be represented in different number of ways and sometimes in infinite number of ways. Therefore, we need to come up with a reliable technique to solve out that problem by mapping the different representations into a unique format to be used during the search process thereafter.

One major problem of not being able to retrieve relevant items is the inconsistency between the author's vocabulary and the user's vocabulary. Therefore, the user may search for a term that is not provided by the author. This problem has been studied in text search, and there are some proposed solutions; such as searching for the synonyms during the search process using thesaurus lookup. A similar problem related to equivalency exists when you search for a mathematical expression, because the term y+x is the same as x+y mathematically,

Although the current search engines are equipped with tools to enhance their ability in retrieving items that contain a certain type of mathematical expressions, they still fail in retrieving the documents that contain variants of that mathematical expression. Therefore, there is a need for a way to retrieve the documents that contain, not only the expression itself, but also the expression's equivalent forms.

The online-reasoning systems can, in theory, be used to check for equivalence between query expressions and content expressions. Those systems would take prohibitively a long time to check whether a query expression is equivalent (or not) to the expressions in the contents.

Another important reason for the failure of current search engines in retrieving mathematical expressions is that search engines do not understand mathematical structures, but they well-understand text because a word in an unstructured text is simply a word with no data type definition and no conceptual definition.

Mathematical expressions are well structured, and the structure itself holds their correct interpretations. For example, in math there is a difference between 2*(x2-x3) and 2*x2-x3. However, if we were doing text retrieval there would be no difference between both expressions.

A. Definition of Mapping

Mapping is a sequence of transformations that is concerned with transforming an original expression form one algebraic/structural form into an equivalent one. According to this definition, the Mapping is divided into two types: algebraic and structural Mapping. In algebraic Mapping, the process of Mapping is done on the expression in its algebraic form. Therefore, the algebraic form changes after mapping.

The same Mapping can be called structural mapping when the structure of the parse-tree representation is changed after applying the Mapping process. In structural Mapping, the expression's parse tree [10] [11] structure will change after Mapping the mathematical expression to its equivalent one. For this reason, we call it structural.

For example, once the expression x+z+y is mapped to its equivalent form x+y+z, this mapping is called algebraic; because the algebraic form of the expression has been changed. In addition, the parse tree for the expression x+z+y, before the mapping, is shown in Figure 1.

The structure of the parse-tree representation shown in Figure 1 can be changed to the parse-tree that is shown in Figure 2

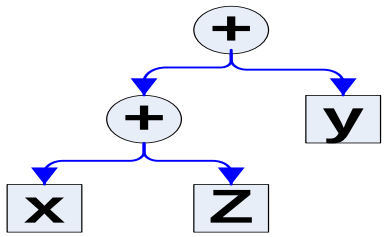

Figure 1. Parse-tree representation for x+z+y before the mapping

.

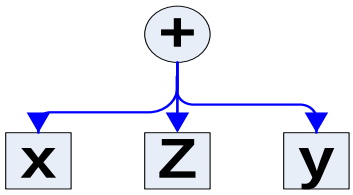

Figure 2. Parse-tree representation for x+z+y after the mapping

Notice that the structure of the parse tree has changed from one form to another after applying the mapping process.

B. Equivalence Detection and Mapping

The Equivalence Detection and Mapping (EDM) aims to transform the expression tree into a normalized one. This tree is equivalent to the original tree, but it is an agreed upon representation, based on some rules, to facilitate the search process. Therefore, the normalized tree should be the common form between the searchable database and the mathematical expression as a search term. The proposed framework should be able to update the query and the database content dynamically, so that they are both transformed into a common form.

A detailed description about this component can be found in [1], and [12]. The authors of [12] outline the main component of this subsystem. The Mapping rules are built based on a context free grammar (CFG). The authors built the component that contains the rules that are responsible for the mapping process. The mapping process verifies that a set of expressions are equivalent. For example, this component verifies that the x+y and y+x are equivalent or not. This approach is different than the theorem proving systems. The theorem proving systems verify whether the given two expressions are equivalent; whereas the EDM finds all the equivalent forms of a certain mathematical expression [13]. The process done by the EDM is faster than the one that is applied in the theorem proving systems.

In order to detect the equivalency of math expressions, a context free grammar is built to verify the format of the added mapping rules. This component is built based on the properties of the digital ecosystems [5], in which the system is able to update itself based on different user specifications, and based upon any added rules. This component is able to normalize the database content and the user query based on a list of mapping rules. There is no need to modify the system, because the mapping component is reacting automatically.

The mapping system is built based on the grammar that is responsible for verifying the format of any added mapping rule. The rules are added to check the equivalency of math expressions. The purpose of the grammar is to constraint the format of the added mapping rules. Any added rule that does not comply with the predefined grammar, is send back to the user in order to reformat it again. Notice that, the above rules can be written in a different way based on the way the user writes the grammar. Thus, the grammar decides how those rules are written and decides the further steps to be taken thereafter.

C. WildCards in Math Search Systems

The wildcards have been used in math search systems to achieve several purposes. The authors of [14] have used the wildcards to extend the current math query languages. The introduced three sets of wildcards are used for more precise structural search, and multi-level of abstraction. The authors of [14] introduced wildcards for several math operations, such as matrices, partial differentiation, and for function composition.

The query language that is introduced by the authors of [15] contains a set of wildcards. The implementation of this query language maps the queries written in that language into Xpath/Xquery quires [16][17]. The authors of [15] assumed that the math content is in MathML.

The introduced framework in this research can benefit from the proposed wildcards in [15] by providing a set of wildcards that can be used in the mapping process. In addition, the wildcards can be used during the search process in which the math query language in [14] can be tailored to be used with the proposed framework; especially the math expressions in [14] are assumed to be represented as parse-trees.

D. Generic Mapping

Based on the mapping component (i.e. the grammar), the system administrator should be able to add any valid mathematical equivalence rule. The Mapping system should be able to detect equivalency in math expressions. The rules tell whether two or more expressions are equivalent or not. In addition, the areas of math that our system has provided equivalence detection for must be determined.

Also, the system should be able to determine which group of users is targeted. Algorithms are developed to detect equivalency for any added rule that conforms to the grammar; any added rule to the generic Mapping (GM) system is derived from a general principle in which a rule is admissible, if and only if, there is a corresponding transformation on the parse-tree [12].

The GM processes a massive amount of math content. Thus, there are difficulties associated with searching such content using current search engines as mentioned before. Consequently, this research adapts the concept and properties of digital ecosystems trying to enhance the ability of GM system in increasing the precision and/or recall when searching math content.

Accordingly, the GM system has been developed to be:

- Able to be incorporated in different environments, i.e. web-based systems, math-search systems, etc.
- Designed as a separate component that can cooperate efficiently with other ecosystems.

- Flexible in which a user can choose whether to apply the GM or not. Users can notice the benefits of the GM after it has been used.
- Scalable in which the GM can be easily expanded to include all related math content.

Any added Mapping rule is validated in order to verify whether it is compliant with the grammar or not. This process is implemented using javaCC [18].

Figure 3 summarizes the detailed components of the Mapping sub-system.

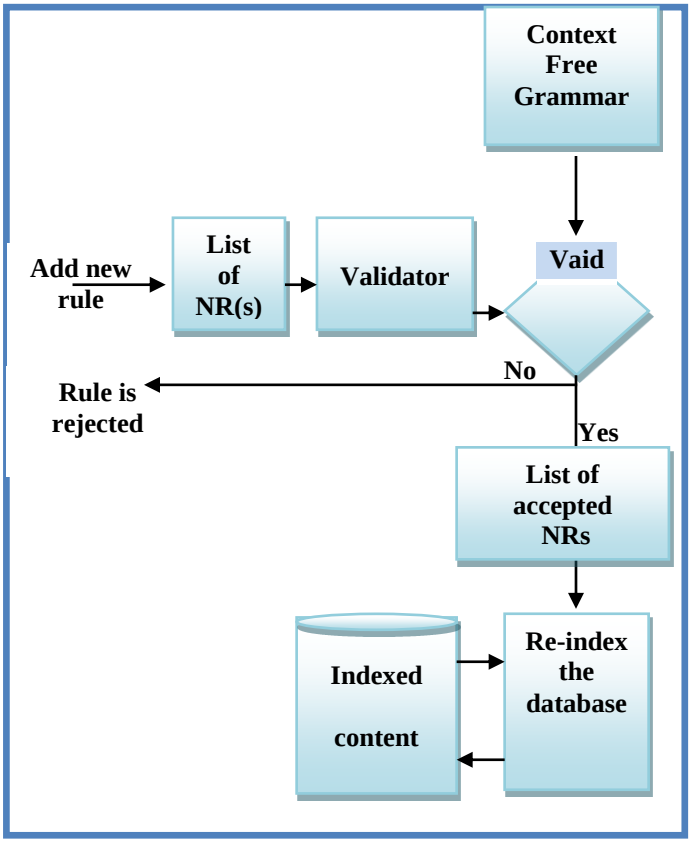

Figure 3. General Mapping System Based on a Conetxt Free Grammar

The framework that appears in Figure 3 can be enhanced by adding an intelligent component. This component can achieve several tasks on behalf of the user and the system. For example, the intelligent component can automatically check the correctness of the user's query, and at the same time, provides help for the user trying to figure out the query. The intelligent component can contain several well-known math functions and properties (axioms, trigonometric functions… etc). Notice that the context free grammar is defined by the user, in order to enforce certain format for the mapping rules. The mapping rules are responsible for mapping an expression to all of its equivalent forms.

VII. The Represening Component

Before getting into the proposed indexing approach, there is a need to discuss the different approaches to represent mathematical expressions.

Currently, there are many available representation techniques that have been used to represent mathematical expressions. For example, text-based mathematical constructs, XML-based math content, tree-based representation, and other representation techniques such as Box model [4].

In the designed framework, the parse tree representation is adapted as an efficient and reliable representation technique. For example, the tree operations are efficient; comparing sub-trees operators (i.e. sub-expressions) is easily implemented. The more important feature is that the structure of a mathematical expression conveys the correct interpretation of those expressions, and the tree representation can hold the structure of those expressions. In fact, using the parse trees to represent the mathematical expressions maintain the properties of math expressions. For example, the operator precedence is maintained when representing the expressions using parse trees.

As new math constructs are being added to the web, the use of parse trees is highly recommended. The already math-based content can be converted into their parse tree representation. The conversion process has some technical issues that can be handled once. Those different representations can be mapped to their parse trees' representation.

The sub-system that is responsible for converting the different representation into a unique one can be implemented based on the digital ecosystems properties. For example, the conversion can be done automatically based on the existing representation of an expression. The component that does the conversion can be enhanced with some rules, knowledge, or any other model that might contribute to the correct conversion.

VIII. The Indexing Component

There is not a well-defined indexing approach that can be used to index math-based content. Even with the extensive indexing of Metadata, users can only search for the math expression itself [4] [19]. A mathematical expression can be found in different equivalent forms. The existing techniques are not mature enough to fulfill the special requirements that are associated with the nature of mathematical constructs.

Based on the parse tree representation, a new promising technique has been proposed to index math-based content. The whole approach depends on assigning an agreed upon values for each parse tree node. Those values can be taken from a lookup table. Certain calculations can be performed on those values to extract a unique one to be used to index the whole parse tree. This approach is similar to the hashing technique in which the idea of the function that is used in our proposed indexing technique is similar, somehow, to the hashing function.

An ongoing experiment is being implemented to test the result of this technique. The preliminary result is impressive and the more clarification and details about this technique might be available soon.

IX. Arabic Math Expression

Thus far, most of the researches that work on math expressions are interested in English-based math expressions.

The number of researches that work on processing Arabic-based math expression is relatively few. For that reason, it is recommended in this research to work on Arabic-based math expression, and expand the framework to work for both language-based math expressions.

Arabic-based math expression maintains the same structure and math properties of the English-based math expressions. The main distinction between the both of them is in the used language.

In order to expand the framework there is a need to do further steps on the current framework. For example, the mapping rules will be different because the language is different. Accordingly, the grammar that checks on the format of the mapping rules is different as well.

Several modifications are needed in order to enable the current framework to process both Arabic and English-based math expressions.

X. The Proposed Framework

The proposed framework is depicted in Figure 4, in which the math query is transformed into its equivalent parse tree representation. The steps thereafter depend on the user demand whether to search within the un-mapped database or to search within the normalized one. Once the user choses to search within the un-mapped database; the system takes the user right to the un-normalized database. The user using this system can chose to do the Mapping, and then the query is mapped into the normalized one based on a set of predefined rules of mapping. The search process after that completes as it does in any other search systems in which the indexing process is performed on the normalized database. Notice that the mapping is performed on both, the user math query and on the searchable math content (i.e. math database). The designed framework enables the user, after applying the mapping process, to search the un-normalized database as well. In this way, the user has the ability to compare the results of the search process under different scenarios.

XI. Significant Contributions

This research makes the following significant contributions to the field of math search.

- A new indexing approach that can be utilize to index math-based content.
- A proposed approach for representing different types of math constructs and an approach to convert the already existing math constructs to the new proposed representation.
- Introducing the Arabic-based math expression processing in which the same operations of English-based math expressions can be applied on the Arabic-based ones.
- Introducing the intelligent component that can be developed and enhanced with several math-related functionalities.

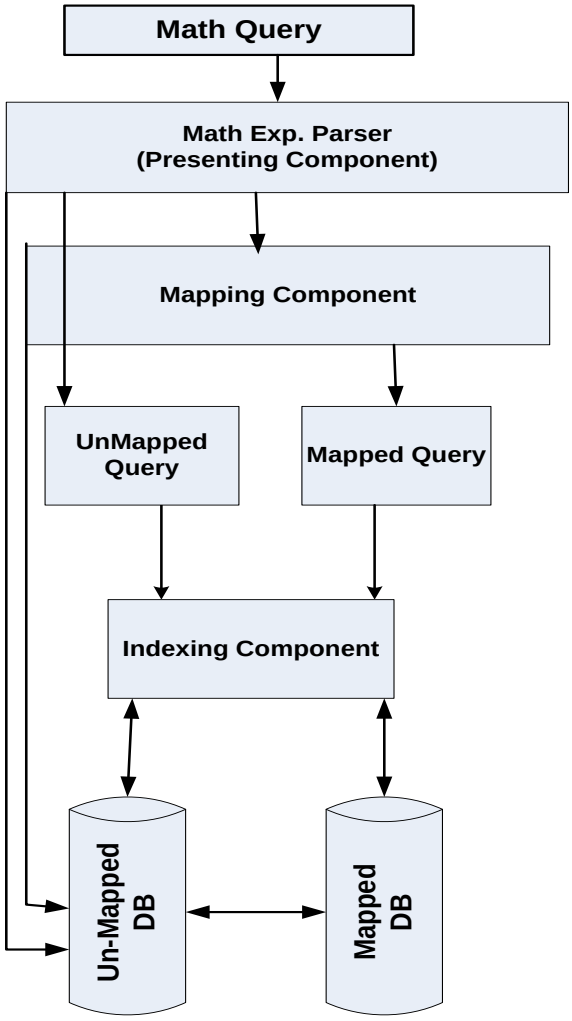

Figure 4. The proposed framework

XII. Conclusion And Future Direction

This research introduces a framework for math-based search systems based on digital ecosystems properties. The proposed framework consists of three main components.

The Mapping component shows that we have achieved some progress in searching for a mathematical expression (e.g. y+x).After applying the Mapping and equivalence rules, the recall and even precision of our search will be increased. Since we are transforming different equivalent mathematical expression into a common form, this common form will be compared against the searchable database, which contains the normalized form of that expression as well. According to that, the comparison process will end up finding most of the items that have the common mathematical expression.

The Representing component which represents the math expressions using parse tree, and convert the already existing expressions to that form. The output of this component is a math content represented using parse tree.

The reason behind this component is because the existing math content is represented in different formats, which make it more difficult to design a specialized system to search this content. Therefore, adapting the concept of digital ecosystems to standardize the way a math expression is represented minimizes the difficulties of searching such content.

The third component is the Indexing one. This component indexes the math content using an approach of assigning a unique value for each parse tree, and then uses that unique value to search for a specific tree. In addition, this approach allows for sub-expression comparison which enables the searching for a sub-expression within an expression.

The researchers still have too much to do on this field, such as:

- Combine the text-based indexing approaches with the proposed approach to a get better result when the user search for text mixed with a math expression.
- Add extra component to the Representing component (e.g. image) to be able to convert multimedia-based representations to the proposed representation.
- Expand the mapping rules, the grammar that verifies the format of the mapping rules, and the framework to work for Arabic-based math expression as well as English-based math expressions.
- Develop a comprehensive math query that can be used to search efficiently for a math expression based on a parse tree representation.

REFERENCES

[1] Mohammed Shatanwi, Abdou Youssef, "Equivalence Detection Using Parse-tree Normalization for Math Search ", ICDIM 2007 Lyon-France Oct 28-31-2007.

[2] Youssef, A. "Information Search And Retrieval of Mathematics Contents: Issues and Methods", The proceeding of the ISCA 14th International Conference on Intelligent and Adaptive Systems and Software Engineering (IASSE-2005), July 20-22, 2005, Toronto, Canada.

[3] Youssef A., "Roles of Math Search in Mathematics,"Spring-Verlag volume the Lecture Notes in Artificial Intelligence series. Also, an invited paper, the 5th Int'l Conf. on Mathematical Knowledge Management, August, 2006, UK, pp. 2-16.

[4] Abdou Youssef, Bruce R. c "Technical Aspects of the Digital Library of Mathematical Functions, Annals of Mathematics and Artificial Intelligence, Volume38, pp. 121-136, 2003.

[5] H. Boley and E. Chang, "Digital ecosystems: Principles and semantics," in Proceedings of the Inaugural IEEE International Conference on Digital Ecosystems and Technologies, 2007, pp. 398–403.

[6] P. Dini, N. Rathbone, M. Vidal, P. Hernandez, P. Ferronato, G.Briscoe, and S. Hendryx, "The digital ecosystems research vision:2010 and beyond," European Commission, Tech. Rep., 2005.

[7] Michael Kohlhase , "MATHML Presenting and Capturing Mathematics for the Web", Carnegie Mellon University, http://docbu.com/2011/09/08/mathml-presenting-and-capturing-mathematics-for-the-web/, last access in September 2010.

[8] Kowalski, Gerald J., Maybury, Mark T. "Information Storage and Retrieval Systems: Theory and Implementation", Springer, 2nd edition, 2000.

[9] Sergey Brin, Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", Proceedings of the 7th international conference on World Wide Web, Brisbane, Australia, 1998.

[10] Derivations and Parse Trees, http://www.cs.nuim.ie/~jpower/Courses /parsing/node24.html , last access in March 2011

[11] Christopher W. Fraser, Robert R. Henry, Todd A. Proebsting, "BURG -- Fast Optimal Instruction Selection and Tree Parsing", December 1991.

[12] Abdou Youssef, Mohammed Shatnawi, "Math Search with Equivalence Detection Using Parse-tree Normalization", The 4th International Conference on Computer Science and Information Technology, April 2006, Amman, Jordan.

[13] Automated Theorem Proving, http://www.cs.miami.edu/~tptp/OverviewOfATP.html

[14] Moody Altamimi, Abdou Youssef, "Wildcards in Math Search, Implementation Issues," 19th International Conference on Computer Applications in Industry and Engineering, November 7-9, 2007, San Francisco, California USA.

[15] Abdou Youssef (Jointly with Moody Al-Tamimi), "A more canonical form of content MathML to facilitate math search", The 2007 Extreme Markup Languages conference, Montréal, Canada, August 7-10, 2007.

[16] World Wide Web Consortium, "XML Path Language (XPath) Version 2.0," 2005. http://www.w3.org/TR/xpath20/.

[17] World Wide Web Consortium, "XQuery 1.0: An XML Query Language," 2007. http://www.w3.org/TR/xquery/

[18] Java Compiler Compiler, https://javacc.dev.java.net/, last access in February 2011.

[19] Lozier, D. W., Miller, B.R., and Saunders, B.V., "Design of a Digital Mathematical Library for Science, Technology and Education". Proceeding of the IEEE Forum on Research and Technology Advances in Digital Libraries; IEEE ADL '99, Baltimore, Maryland, May 1999.

AUTHORS PROFILE

Mohammed Q. shatnawi received his undergraduate degree in computer science from Yarmouk University/ Jordan in June 1995. Shatnawi joined the Ahli National Bank/ Amman as programmer in 1999 for 6 months. After completing his master and D.Sc. studies at the George Washington University/ DC, in January 2007 he joined the Faculty of Computer and Information Technology/ Jordan University of Science and Technology. He is currently working for the computer information systems department. His research interests are in information retrieval, supply chain management systems, CRM, data mining and algorithms.

Qusai Abuein received his B.Sc. in coputer science from Yarmouk university/ Jordan in June 1993. Abuein has joined the Yarmouk un is an assistant professor at the Computer Information Systems in Jordan University of Science and versity computer information center as a programmar and system analyst for five years. Abuein completed his master and Ph.D. in computer science from Japan and currently he is an assistant professor in Computer Information Systems Department in Jordan University of Science and Technology. His research interests are in computer cryptography, information retreival and web technolgies.