An Expertise Recommender System for Web Cooperative Production

Muhammad Aslam[1], Ana Maria Martinez Enriquez[2], Muhammad Tariq Pervez[3], Zakia Saeed[4]

[1]Department of CS & E, U.E.T., Lahore, Pakistan
[2]Department of CS, CINVESTAV-IPN, Mexico
[3]Department of CS, Virtual University Shadman Campus, Lahore, Pakistan
[4]Faisalabad Institute of Cardiology(FIC), Faisalabad, Pakistan
tariq_cp@hotmail.com

Abstract: This paper focuses on providing dedicated expertise recommender system to enhance awareness among group members, working in a distributed cooperative environment. Normally, coauthors lack the information about the production capabilities of their colleagues. As a result of this lack, when they need assistance for the production of complex objects (formulae, figures, style sheets, etc.) they ask their colleagues for help, consequently the authoring process is disturbed. On the other hand, personal referrals may not be useful due to human biasing, liking, and disliking. The existing expertise recommender systems work on user profiles containing user qualification, experience, and history of solved problems. These systems require manual database updation which can be performed by only skilled person. We treat the issue by developing an expertise recommender system which is in-charge to seamlessly observe user activities and to auto detect a possible human expert of elaborated productions on the basis of a generic criterion. Whenever, a participant is deduced as a novice having some production problem, the developed system recommends him/her the presence of an expert with whom the novice can communicate. The entire goal is to enhance awareness coordination among collaborator activities and hence to generate a consistent shared production. [Journal of American Science 2010;6(7):106-112]. (ISSN: 1545-1003).

Keywords: Knowledge based systems, presence awareness, collaborative information filtering, recommender systems.

1. Introduction

In the term of Computer Supported Cooperative Work (CSCW), the CS interpretation deals with the design and development of suitable computer systems, including both hardware and software, to support the work produced by a group of people, distributed over local or wide area networks whereas, the CW interpretation concerns with the fundamental concepts of those disciplines that study the ways people work in groups. These fields are sociology, psychology, ergonomics, organizational theory, and management sciences. Thus, CS enforces technical requirements of such environment whereas CW emphasizes social factors of group users (Grudin, 1994) (Schmidt and Bannon, 1992).

In order to produce in groups, users are assigned roles and designated tasks. While assigning roles and tasks, knowledge and expertise level of each participant is taken into account. In a scenario, if a group member has to perform a task in which he has little or almost no experience he may frequently disturb his colleagues asking for assistance. This disturbance, not only affect the user efficiency but also affects the cooperative production process.

The assistance provided to users depends upon their familiarity with the system, the nature of problem, time constraints to complete the task, and so on. These factors give reasons to highlight four kinds of assistance like quick reference, task specific help, full explanation, and tutorial. All these supports require users to have some knowledge to put inquiries, their proper formats, syntax and semantics. These requirements may frustrate users loosing their interest in the actual task (Babin et al., 2009). Users are assisted by means of *wizards* that guide a user step by step to complete the task. The user can perform complex tasks safely, quickly, and accurately. However, wizards may constraint with some information that user may not have (Gutwin and Greenberg, 2002). An assistant is software tool which observes the sequence of user actions on the basis of which suggestions or hints are given to complete the task.

User can hide the assistant any time. But, this kind of help becomes irritating for its users as an assistant interrupts user work unnecessarily unless disabled.

On the other hand, the CSCW systems have no function integrated into them by means of which a user can be detected when he is in problem. Additionally, if a user faces a difficulty in common production, no information is provided to him/her about his/her participants who can assist in solving the problem. Thus, he/she leaves with no other means to use personal referrals to complete the tasks. Such references have their own limitations: human biasing, personality clashes, liking, and disliking.

Thus, it is needed to improve inter user assistance by automatically deduction of their expertise and the notification of available tools and human experts who help a beginner to reach their objectives. The investigated problem and the provided solution take place as part of the presence awareness. The developed system qualifies the user production on the basis of predefined criteria and deduces a user as possible expert. We validate our approach in designing and prototyping in a distributed coauthoring application that allows users to produce shared structured documents.

Expertise Recommender (ER) (McDonald and Ackerman, 2000) assists the user in trouble and makes a request by giving the area for expertise (e.g. technical support, programming development support), the choice from social network (e.g. students, professors, advisers) and the error code produced by the compiler. Users have the possibility to access the database of experts, and know their status. In this way, users can contact these experts, taking into account the topic area and "social network" level. The Expertise Finder (EF) (Hughes and Crowder, 2003) uses the record of the organization and the knowledge of the people to recommend the expertise. It provides a list of the documents and the experts with their contact numbers. Dynamic Expertise Modeling from Organizational Information Resources (DEMOIR) (Yiman and Kobsa, 2000) develops and tests expertise modeling algorithms. It recommends the documents and the experts that are locally or remotely available.

We present materials and methods of the expertise recommender system in Section 2. Section 3 presents results and discussions. Finally, we conclude our research (Section 4).

2. Materials and Methods

The developed system, Presence Awareness Recommender System (PARS) helps in finding experts and assists the beginners to complete their work. In the studied test bed application (Decouchant et al., 2001), the writing actions performed by authors on the produced document are captured in the form of events by means of a Distributed Event Management Service (DEMS) (Decouchant et al., 2002). An event represents a state change of a shared entity. A shared entity may be a document, hardware/ software, and participants. However, many actions not necessarily cause a change within a resource, such as select and copy or highlight any part of the document, and they are also perceived as events. DEMS acts as a communication mechanism between the producer and consumer applications of events. A producer generates events and can be configured for extending or restricting broadcasting of some events depending upon their scope. A Consumer subscribes to DEMS to receive events. There are some rules to filter events by category and their sources. When an author wants to stay intensively focused on his/her production, he/she may allow those events to be received which are resulted from coauthors who annotate the part of document he/she produces, whereas, other event notifications are restricted. Producers and consumers are uniquely identified by DEMS to control the event broadcasting, and different meta-data is associated to events:

- Entity user who produces events (login, his ID, the working site from where the user works, as well as the cooperative application).
- Entity resource within the cooperative environment (text, figure, formulae).
- Entity action is any performed action within the environment using cooperative application.

For example, in an event: galaxy_tito_ writing-editor_MathML_select, “tito” user works from the “galaxy” site, and selects a mathematical statement. Due to the “select” action DEMS identifies who is on line and his working site.

The developed system, an event consumer, uses rules written in the first order predicate logic. A rule is composed of the premise and the action part. The premise part consists of facts like performed action, user authoring role, nature of produced object. In the action part, the system deduces new knowledge or trigger actions to provide a dynamic user environment. When a user starts a session, the system is automatically launched until session is closed. It follows these steps: a) information catching, b) the deduction of new facts, and c) the proposition of actions (Martinez et al., 2002).

Our authoring application uses MathML (Carlisle et al., 2003) to write, represent, and interpret mathematical statements. Specific events are generated during the production or selection of an expression. As a result of selecting, the author is notified about whether a formula is well-structured (wsf), implicit contextual functions related with (keyboard shortcuts), rewriting it as infix, prefix, or postfix form. In our approach, a criterion is established to evaluate the complexity of a formula that can be modified by editing the specific rule. A weight to the following steps is assigned (see Table 1):

- the way by which it is created: a sequence of symbols or by using the menu,
- steps taken to produce it: sequence of characters from the standard input (keyboard) or using menus dialog boxes or palette,
- elapsed time to complete it.

Weights assigned to a constant and a variable. These values are used to calculate the associated complexity of each MathML structure. Due to the different roles played by users within the cooperative environment, the evaluation of the collaborative production is necessary. Taking into account this evaluation, the developed system qualifies a user as an expert, so it is a dynamic evaluation.

3. Results and Discussions

3. 1. Evaluation of Mathematical Expressions

The evaluation refers to the analysis concerning: a) the kind of elements included in the shared document: text, tables, links, formulae, graphical. For instance, a figure is defined by the vertical and horizontal alignment, its appearance, etc. b) the number and the complexity of these elements, and c) the elapsed time to produce them.

In the case of the mathematical production, the evaluation is based on: a) the number of well-formed formulas produced and their associated production complexity, b) the elapsed time to produce them, and c) the performed actions during their production (undo/redo, copy/cut/paste, etc.). Once a formula is concluded, the system verifies if it is well-structured formula (wsf). A MathML formula is said to be WSF if all components of MathML structure/pattern are filled a well-formed formula in mathematical context. Afterwards, its complexity is calculated.

a. The Complexity of a Formula

The complexity of a formula depends on the way by which a formula is created (either using "Type Menu" or "Math" icon), the steps taken, and the elapsed time to complete it: number of times the "type menus" are selected, as well as the math dialog box, and the math palette are used (see Table 1). A MathML pattern can be characterized by the number of components to be fulfilled:

- Single Component (e.g. square root). The complexity is equal to the complexity of the single component base expression, multiplied by 1.5. For instance, giving expression 1, and applying the rules #1, #2, #3, #4, #5, #9 and #10 from Table 1, the resultant complexity is 17.40.

$$\sqrt{x^2 - 2 * x * y + y^2} \qquad \text{….1}$$

Table 1: Complexity of mathematical expressions

MathML Pattern	Rule No.	Complexity	MathML Pattern	Rule No.	Complexity
Constant	1	1.0	x_y	8	comp (x) + comp (y) + 0.8
Variable	2	1.5	x^y	9	comp (x)+comp (y) + 1.0
$x+y$	3	comp (x) + comp (y)	$\sqrt{x}$	10	comp (x) * 1.5
$x-y$	4	comp (x) + comp (y)	$\sqrt[y]{x}$	11	comp (x)+comp (y) * 1.8
$x*y$	5	comp (x) + comp (y) + 0.3	$\int_x^y zdz$	12	(comp (x)+comp (y)) * comp (z)
$x=y$	6	comp (x) + comp (y)	$\sum_x^y z$	13	((comp (x)+comp(y)) * comp (z)) + 1.5
$\frac{x}{y}$	7	comp (x)+comp (y) + 0.5	$\prod_x^y z$	14	Comp (x) * comp (y) * comp (z)

- Twofold component (e.g. fraction, under root). The complexity depends on the basic expression associated to the pattern as we see in Table 1, rules #7, #8, #9, and #11. The complexity of Expression 2 is 23.58.

$$\sqrt[x]{x^2 - 2*x*y + y^2} \quad ....2$$

- Threefold component (Sub-superscripts, and under-overscripts). The complexity of these patterns is related to the complexity of the "z" base component, (rules #12, #13, and #14 from Table). The computed complexity of Expression 3 is 10.00.

$$\int_1^4 3*x^2 dx \quad .....3$$

The complexity of a formula is calculated by the following rule:

```
Startrule "Computing complexity"
If author(fragment_1) = x
   role(x) = "Writer"
   nature(fragment_1) = "formula"
   update(fragment_1) = "true"
   evaluate_wff(fragment_1) = "true"
                                  /* formula is wff */
Then
   announce(x) <-
computed_complexity(fragment_1)
       /*Complexity of the formula is calculated */
Endrule
```

A user can ask the system explicitly to calculate the production complexity of a generated formula. Table 1 establishes the complexity of basic expressions, and these patterns are given as input data to the system, thus, it is possible to change these values without modifying the code of the system. We argue that this is dynamic criteria and can be updated as desired. In addition to complexity, other factors are also considered to evaluate the user's production.

b. Undo/Redo Actions

We performed an experiment with 10 users, 5 of them had knowledge how to write elaborated large documents (expert). We asked them to reproduce some formulas, like binomials, integrals, and quadratic ones. In our experiment, experts performed 20 actions and reverting 2 actions on the average to produce expression 4. While, beginners generated same expression with 38 actions while reverting 18 actions on the average. The counted actions are: starting of the formula (selection of "Math" button), insertion of each pattern (square root, superscript, etc.), positioning the cursor in the pattern, producing the greek letters from math palette, and insertion of each number, identifier, and operator from the standard input (keyboard).

$$\frac{-b \pm \sqrt{x^2 - 4*a*c}}{2*a} \quad ...4$$

The following rule is applied to count do/undo/redo actions:

```
Startrule "Actions-time-size"
If action(x) = "start_formula"
  nature(fragment_1) = "formula"
Then
total_actions(x)<- start_count_total_actions(x)
                              /* actions are counted */
undo_actions(x)<-start_count_undo_actions(x)
                       /* reverted actions are counted*/
init_time(fragment_1)<-reg_curr_time(fragment_1)
                        /* Starting time is recorded */
initial_size(doc)<-reg_init_size(doc)
                        /* doc initial size is recorded*/
Endrule
```

When a user starts a formula, the number of reverted actions, the current system time and the size of the document are registered. This information helps to evaluate the user production when he/she conclude the expression. When a user reverts more than half of his/her actions, he/she is notified of presence of experts with whom the former can communicate with.

When users produce a formula, their performed actions are observed, if they are in trouble, the system proposes them to contact experts using synchronous communication. The well-formed formulas produced within a shared document during a session are counted by another rule. Depending on the number of produced well-formed expressions and their complexity, the system can classify a user as a possible expert in writing expressions (see Section 5).

c. Elapsed Time

The time during which a user writes a well-formed expression in one session, without performing other activities is the "elapsed time". This time is calculated by taking the difference between the registered time (the “Actions-time-size” rule) when a user starts to produce a formula and when he/she completes it. The system memorizes the time when the formula is concluded. The following rule is applied:

```
Startrule "Final time-Final size"
If author(fragment_1) = x
  role(x) = "Writer"
  nature(fragment_1) = "formula"
  update(fragment_1) = "true"
Then
```

```
   final_time(x) <- register_current_time(x)
                                /* the time is recorded */
   final_size(fragment_1) <-
                              reg_final_size(fragment_1)
                                      */ size is recorded */
elapsed_time(x) <- final_time(x) - initial_time(x)
                              */elapsed time is computed */
calculate_change_size(fragment_1) <-
   final_size(fragment_1) –
                               finitial_size(fragment_1)
                         /* change in size is computed */
Endrule
```

From the group of 5 experts and 5 beginners, we studied that initially the beginners were spending more time to insert mathematical patterns (fractions, exponents, subscripts, etc.). The duration was decreased when the expertise was grown to insert data and produce elaborated expressions. The observed average spent time was:
Production time (seconds)= a*(no. of mathematical patterns)+ b*(no. of greek letters)+ c*(characters from standard input)+ (elapsed time for the selection of menus, math box, math palette, and position to insert characters), where, a=b=5, c=2.
These values were obtained according to the experimentation. We observed that experts take less than 5 minutes (average) to produce a formula containing mathematical patterns and greek letters (e.g. quadratic formula). Therefore, when a user spends more than 10 minutes to produce a formula, whose complexity is less than 16, he/she is informed about the existence of an expert. The following rule applies:

```
Startrule "Large elapsed time"
If  author(fragment_1) = x
    nature(fragment_1) = "formula"
    update(fragment_1) = "true"
    elapsed_time(x) > 10   /* time in minutes */
     computed_complexity(fragment_1) < 16
     status_open(authors_definition_db) = "true"
Then
    announce(x) <- MathML_expert_exists"
Endrule
```

The results obtained in the experimentation made for producing mathematical expressions is shown in Table 2. The produced formulas, their complexities, and the elapsed time by each expert and beginner are presented. The average spent time of experts and beginners is computed. For instance, to produce a quadratic formula (at serial 2), experts spent 97 seconds on average whereas beginners took 625 seconds.

Table 2: Time elapsed by experts and beginners

Expressions	Production Complexity	Time spent by Experts (sec) / Time spent by Beginners (sec)					Average time spent (E /N)
		E1/N1	E2/N2	E3/N3	E4/N4	E5 / N5	
$a^2 - b^2$	8.0	14/142	15/90	18/80	14/65	14/150	15/102
$\frac{-b \pm \sqrt{x^2 - 4*a*c}}{2*a}$	16.6	177/734	40/617	96/540	83/616	91/617	97/625
$\sum_{n=0}^{\infty} a_n * t^n$	16.2	118/526	70/430	80/152	91/420	86/223	89/350
$\int_{-\Pi}^{\Pi} f_N^2(x)dx$	26.0	135/425	70/356	104/183	109/265	99/225	103/290
$n! \sim \sqrt{2*\pi} * n^{\left(n+\frac{1}{2}\right)} e^{-n}$	21.7	93/360	65/282	73/125	79/124	89/221	80/222
$e^{-C*x} \frac{1}{x} * \prod_{n=1}^{+\infty} \frac{e^{\frac{x}{n}}}{1+\frac{x}{n}}$	50.6	100/585	102/425	121/747	113/494	111/280	109/506
$\int_a^b (1-t)^{x+y-1} \left(\frac{t}{1-t}\right)^x dt$	45.0	122/ 495	85/430	104/670	117/398	113/722	106/543

d. Evolution of the Production

The size of the shared document is registered when: 1) a user starts to produce a formula, and 2) he/she completes a well-formed version. The difference between registered sizes gives the variation in the size of the production in bytes. When an expert generate a formula, the document's size gradually increases, whereas, the change in size was very slow when a beginner produces it.

In our study, the average increased size is approximately 50 bytes per minute in case of expert's production whereas a beginner produces less than 15 bytes per minute. Thus, when the difference in size of the production is less than 10 bytes per minute, the user is asked if he/she is in trouble.

We emphasize again that knowledge/rule base of our PARS recommender system is entirely independent of its code. The weight/values like production weights in Table 1, related complexity, number of WSFs, the complexity of an expression used for bench mark, etc. can be seamlessly changed to any new value.

e. Expert Presence Awareness and Assistance

"*The information about the presence of collaborators, their attitudes during a communication session, when they are busy and how frequently they produce objects of a particular nature*" (Gutwin and Greenberg, 2002) is defined as presence/social awareness. The PARS system continuously observes the user actions. Whenever, it concludes based on irrelevant/undo actions, document evolution, errors, complexity that a user is in trouble, it notifies him/her about the presence of experts. The notification box offers three buttons: - "Information" to get data about expert like: login name, working site, and availability; - "Contact" to synchronously communicate with them; and - "Cancel" to close the notification box.

In a scenario, Stephan performs irrelevant and undo actions frequently while producing a mathematical formula. The PARS system informs him/her the expert existence, Paul who normally produces mathematical expressions. Stephan may start a communication session with him, and ask how to produce a quadratic expression, as we see in Figure 2. By means of the synchronous communication, offered by our PARS system, Paul explains Stephan step by step how to do it.

Once Stephan finishes the formula, he can send its position within the shared document, by selecting the formula. It means, a) the DEMS service captures the selection event and recovers the unique identifier(s) of the selected section(s): "focus of discussion"; b) using this unique identifier, the developed system displays the "focus of discussion" into all concerned coauthor's environments (Martinez et al., 2002). When the nature of the fragment that a user produces, does not match with his/her common production, he/she is also informed about the existence of experts. Following, we explain how to define experts.

A user who produces elaborated figures, well-formed formulas, and continuously adds different kind of elements, can be considered as an expert. He/she avoids loss of recent changes by saving the document after a gradual period of time By contrast, beginners take a lot of time to perform a simple task doing irrelevant and unnecessary actions because they do not know how to proceed. When a coauthor produces five well-formed formulas in the same session (dynamic evaluation) and the complexity of each one is greater than 16, he/she can be considered a writing mathematical expert. The following rule is applied:

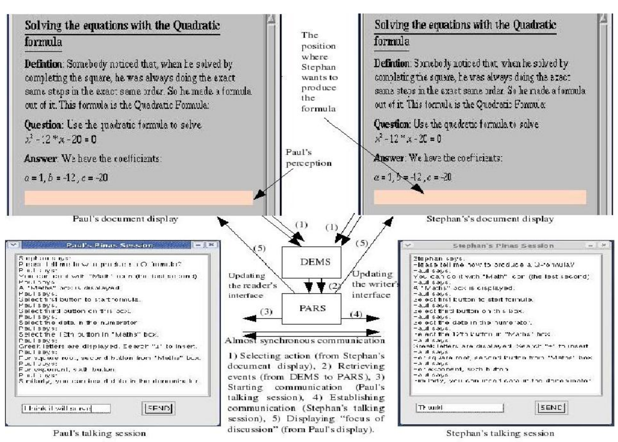

Fig: 2: The discussion between an expert and a beginner

```
Startrule "Defining expertise on writing formulas"
If author(fragment_1) = x
    role(x) = "Writer"
    nature(fragment_1) = "formula"
    update(fragment_1) = "true"
    complexity(fragment_1) > 16 /* from Table 1 */
    summary_wff(formula) > 5
                                /* formulas are counted */
    status_open(authors_def_data_base) = "true"
                                /* author's file is open */
Then
   author_definition(x) <== "expert_Math"/*
                 'x' is an expert in writing formulae*/
Endrule
```

The expertise of a user is saved in the user definition database, as well as, his/her login, user name, working site, and the availability to communicate with. Precise information of an ad hoc expert is suitable information, in case of trouble.

4. Conclusion and Future Perspectives

In order to provide a friendly cooperative working environment to users, it is necessary to inform them the productiveness of their colleagues. The productivity of users on writing mathematical formulas can be determined on predefined criteria. By means of this evaluation, the developed expertise recommender system defines a user as a possible expert. The evaluation criteria can be modified at any time in the rule base interpreted by the system. It enhances presence awareness by notifying the presence of experts to users in trouble. Users can communicate with experts to ask for assistance and hence, they can coordinate their activities.

In future, we intend to establish criteria to evaluate the graphical production. For this purpose, the history of the different versions of the shared documents will be necessary, in order to determine the evolution of the graphical objects.

Acknowledgements:

Authors are grateful to the Department of CS & E, UET, Lahore, for financial support to carry out this work.

Corresponding Author:
Muhammad Tariq Pervez
Department of CS, Virtual University,
Shadman Campus, Lahore, Pakistan
E-mail: tariq_cp@hotmail.com

References

1. Babin L. M.,Tricot A., Mariné C., “Seeking and Providing Assistance While Learning to Use Information Systems”, Journal. of Computers & Education. 2009; 53: 1029–1039.
2. Carlisle, C., Ion, P., Miner, R., Poppelier, N., “Mathematical Markup Language MathML Version 2.0 (2nd Ed.)", W3C, 2003.
3. Decouchant, D., Favela, J., Martínez Enriquez A. M. “PIÑAS: A Middleware for Web Distributed Cooperative Authoring”, SAINT,2001;187-194.
4. Decouchant, D., Martínez, A. M., Favela, J., Morán, A. L., Mendoza, S., Jafar, S., “A Distributed Event Service for Adaptive Group Awareness”, MICAI’LNAI 2313, Mexico, Springer-Verlag, 2002;506–515.
5. Grudin, J., “Computer Supported Cooperative Work: History and Focus", IEEE Computer, 1994;7(5):19-26.
6. Gutwin, C., Greenberg, S., “A Descriptive Framework of Workspace Awareness for Real-Time Groupware", CSCW, 2002;11(3-4):411-446.
7. Hughes, G., Crowder, R., “Experiences in Designing Highly Adaptable Expertise Finder Systems", Proc. of DETC'03, USA, Sep. 2-6, 2003.
8. Martínez, A. M., Muhammad, A., Decouchant, D., Favela, J., “An Inference Engine for Web Adaptive Cooperative Work”. MICAI’2002;526–535.
9. McDonald D. W., Ackerman M. S., “Expertise Recommender: A Flexible Recommendation System and Architecture", ACM CSCW, P.A., 2000; 231-240.
10. Rizzi, C. B., Alonso, M. C., DE Seixas, L. M. J., Costa, J. S. Tamusiunas, F. R., DA Rosa Martins, A., “Collaborative Writing via Web - EquiText", Informatic Education, 2000.
11. Schmidt, K., Bannon, L., “Taking CSCW Seriously: Supporting Articulation Work", CSCW: An Int. J, 1992; 1(1):7-40.
12. Yiman, D., Kobsa, A., “DEMOIR: A Hybrid Architecture for Expertise Modeling and Recommender Systems", IEEE 9th International WET ICE'00, Mar. 14-16, 2000.

08/04/2010.