The Journal of Systems and Software 86 (2013) 2184–2195

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Performing and analyzing non-formal inspections of entity relationship diagram (ERD)

Nergiz Ercil Cagiltay[a], Gul Tokdemir[b], Ozkan Kilic[c,*], Damla Topalli[d]

[a] Atilim University, Software Engineering Department, Ankara, Turkey
[b] Cankaya University, Computer Engineering Department, Ankara, Turkey
[c] Middle East Technical University, Informatics Institute, Ankara, Turkey
[d] Atilim University, Computer Engineering Department, Ankara, Turkey

ARTICLE INFO

Article history:
Received 18 March 2011
Received in revised form 16 March 2013
Accepted 29 March 2013
Available online 25 April 2013

Keywords:
Defect detection
ERD
Eye tracking

ABSTRACT

Designing and understanding of diagrammatic representations is a critical issue for the success of software projects because diagrams in this field provide a collection of related information with various perceptual signs and they help software engineers to understand operational systems at different levels of information system development process. Entity relationship diagram (ERD) is one of the main diagrammatic representations of a conceptual data model that reflects users' data requirements in a database system. In today's business environment, the business model is in a constant change which creates highly dynamic data requirements which also requires additional processes like modifications of ERD. However, in the literature there are not many measures to better understand the behaviors of software engineers during designing and understanding these representations. Hence, the main motivation of this study is to develop measures to better understand performance of software engineers during their understanding process of ERD. Accordingly, this study proposes two measures for ERD defect detection process. The *defect detection difficulty level* (DF) measures how difficult a defect to be detected according to the other defects for a group of software engineers. *Defect detection performance* (PP) measure is also proposed to understand the performance of a software engineer during the defect detection process. The results of this study are validated through the eye tracker data collected during the defect detection process of participants. Additionally, a relationship between the *defect detection performance* (PP) of a software engineer and his/her *search patterns* within an ERD is analyzed. Second experiment with five participants is also conducted to show the correlation between the proposed metric results and eye tracker data. The results of experiment-2 also found to be similar for DF and PP values. The results of this study are expected to provide insights to the researchers, software companies, and to the educators to improve ERD reasoning process. Through these measures several design guidelines can be developed for better graphical representations and modeling of the information which would improve quality of these diagrams. Moreover, some reviewing instructions can be developed for the software engineers to improve their reviewing process in ERD. These guidelines in turn will provide some tools for the educators to improve design and review skills of future software engineers.

1. Introduction

As Genero et al. reports, data requirements of agile companies is rapidly changing (2008), which requires the database designers quickly understand the old conceptual data model and adapt the new requirements and changes appropriately. For database design and system analysis purposes, the entity-relationship (ER) model and its accompanying ER diagrams (ERD) are the ones used mostly (Song and Froehlich, 1995). ERD provides developers with an overall grasp of the data requirements, modeling and database structures of the information system before the implementation phase. Therefore, information system designers need to understand the ERD notations to perform tasks for information system analysis and design (Rochfeld and Bouzeghoub, 1993; Bouzeghoub and Rochfeld, 2000; Gabay, 2001), for schema integration and manipulation (Christensen et al., 2001), for reverse engineering (Akoka et al., 1999), and also for better maintaining current systems (Genero et al., 2008). The ongoing improvements in software development processes have resulted in radical changes and shifts in software design approaches like object-oriented methodology

* Corresponding author. Tel.: +90 5374923542.
E-mail addresses: nergiz@atilim.edu.tr (N.E. Cagiltay), gtokdemir@cankaya.edu.tr (G. Tokdemir), okilic@ii.metu.edu.tr, ozkankilic@gmail.com (O. Kilic), dtopalli@atilim.edu.tr (D. Topalli).

0164-1212/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2013.03.106

(Fichman and Kemerer, 1992). Even in these new approaches, engineers need to understand and analyze the previously developed systems and make necessary modifications as required by the business model.

It is known that depending on the usage, diagrammatic representations can be more powerful than sentential representations (Larkin and Simon, 1987). They capture, communicate, and leverage knowledge that is indispensable for solving problems; they also act as *cognitive externalizations*[1] (Zhang, 1997). Diagrams are important illustration tools to represent information systems since they are helpful for conveying and transferring information in a condensed manner. They are used extensively in design, development, and maintenance processes in software engineering and may offer cost savings and a better understanding of a software problem. When the developers read and translate the designers' notations into programs, they can simultaneously view the overall system in the diagrams that provide a map from the problem domain to the visual representation by supporting cognitive processes that involve perceptual pattern finding and cognitive symbolic operations (Larkin and Simon, 1987). After the initial perceptions, information system engineers perform problem-solving activities like checking system consistency, identifying defects and misrepresentations in the diagrams. Discovering defects in such diagrams requires a deep understanding of the phenomena. Hence, defect detection performance of a software engineer shows his/her level of understanding of the diagram. Additionally, discovering these defects early in the design phase could decrease the cost of system development dramatically.

Main motivation of this study is to analyze the defect detection process of software engineers during their ERD reviewing process. Through this analysis, it is expected to find measures for the difficulty level of the defects seeded in an ERD and the performance of the software engineers during defect detection process. We believe that such measures are helpful tools for the researchers, educators and software companies to provide guidelines to improve the ERD design skills of software engineers and to improve their understanding level.

2. Background and Aim of the Study

Diagrammatic reasoning refers to how diagrammatic (or pictorial) representations can be used in problem solving and reasoning (Chandrasekaran et al., 1995). The active sketches held in the working memory during a problem-solving task with diagrams are called mental images. This allows diagrammatic problem solvers to create, modify and even animate these mental images to aid cognitive activities such as diagrammatic reasoning through visual information.

Opposite to the linguistic or algebraic representations, diagrammatic reasoning helps to understand the concepts and ideas by the use of diagrams and imagery (Glasgow et al., 1995). It not only allows people to gain insight into the way they think, but also it is a potential base for constructing representations of diagrammatic information that can be stored and processed by computers (Glasgow et al., 1995). In software development process, graphical representation together with textual information have been used to clarify the discussions of the design ideas to model entities or objects to be coded as computer programs (Rumbaugh et al., 1999).

There are several empirical studies in the literature which analyze and compare different database modeling approaches according to their diagrammatic reasoning process and design quality. For instance, Shoval and Frumermann (1994) compared entity-relationship (ER) model and object-oriented (OO) approach and searched which of these approaches were easier to comprehend. They performed an experiment with users in two groups who had been trained to use either ER or OO. Users' conception was measured through answers given to questions. Another study (Liao and Pavlia, 2000) compared relational model (RM), the extended entity-relationship model (EERM), and the object-oriented model (OOM) in terms of their quality in data modeling. They considered differences in two concepts: differences in user performance between those three approaches and differences in user performance between the RM and the relational conversions of the EERM and the OOM models. Similarly, Shoval and Shiran (1997) compared EER and OO approaches in terms of design quality. They measure design quality through correctness of the designed conceptual schemas, time required to finish the design task and designers' preferences. Bock and Ryan (1993) compared extended entity relationship (EER) and Kroenke's object oriented model. They measured modeling correctness through an experiment of two groups of trained users.

Additionally, there are several studies on cognitive processes in software development activities and defect detection patterns during software development processes (Klemola and Rilling, 2002; Stenning and Lemon, 2001; Porter et al., 1998; Runeson and Wohlin, 1998; Wohlin and Runeson, 1998). Some studies investigate the cognitive activities during the diagrammatic reasoning process. For instance, Klemola and Rilling (2002) modeled developers' comprehension processes and cognitive activities in software development. Stenning and Lemon (2001) discussed the semantic and computational efficacy of diagrams for reasoning. Moreover, Hungerford et al. (2004) reviewed software diagrams while identifying defects in order to improve software quality. Ericsson and Simon (1993) investigated the search patterns in diagrams through verbal protocol analysis.

Design defects such as problems of correctness and completeness with respect to the requirements, internal consistency, or other quality attributes can be very crucial for software which can directly affect the quality of, and effort required for the implementation (Travassos et al., 1999). Detecting and removing defects before they propagate to subsequent development phases where their detection and removal become more expensive is very important for software development process which requires a need for systematic reading techniques that tell inspection participants what to look for and, more importantly, how to examine a software document (Laitenberger et al., 2000). In other words, defect prevention is an important activity in any software project (Kumaresh and Baskaran, 2000). In this concern, Wohlin and Aurum (2003a,b) have performed a checklist-based reading to improve defect detection performance of software engineers in ERD. Similarly Laitenberger et al. (2000) studied reading techniques of UML diagrams.

Studies on the defect detection process of diagrams reported that experience and expertise improve this process (Porter et al., 1998; Chandrasekaran et al., 1995; Hungerford et al., 2004). As Kumaresh and Baskaran (2000) suggested, it is advisable to make measures that prevent the defect from being introduced in the product right from the early stages of the project. They report that analysis of the defects at early stages reduces time, cost and the resources required. This creates a need to use more direct measures to obtain insight about the perceptual processing of software engineers. In response to this need, eye movement data can provide real-time measures of cognitive processing during diagrammatic reasoning. Just and Carpenter (1976) reported that the location of eye fixation shows what the participant is processing at that time. Besides the location of the eye fixation, its duration for an instance is also important: the duration is associated with how long it takes to process that particular information (Henderson, 2007). According to Rayner

[1] Intertwining internal and external processes means extract information from the external world to enhance thinking.

(1998) the difficulty of the current task is related to mean fixation duration. Other studies have shown that in highly demanding jobs, the mean fixation duration is higher (Loftus and Mackworth, 1978; Underwood et al., 2004). To this end, eye tracker data could provide valuable insights about diagrammatic reasoning of software engineers in ERD representations.

Although several studies have been established to better understand the cognitive process during software development, most of them have been focused on software code. As Wohlin and Aurum (2003b) also state, there are few studies focused on the inspections of ERD. Although there are several researchers examined the process of understanding ERD in a comparative way (De Lucia et al., 2009), the ERD reasoning process is still not clear. Furthermore, theoretical assumptions of these studies were mostly based on indirect measures such as verbal protocols and video recordings. The studies found in the literature neither provide measures that can be used in the evaluation of software engineers' diagrammatic recognition nor use tools that would create rich and quantitative data for deeper analysis. Additionally, there are not many studies established to understand the behaviors of the individuals during analyzing the ERD and their success and failure rates during this process. Hence, there are not many direct objective measures to be used for comparing different settings of these diagrams.

As Runeson and Wohlin (1998) have reported, during software code review process some reviewers are better than others and also some defects are easier to be found than others (Runeson and Wohlin, 1998). If some reviewers perform better than the others, then it means that there are some factors that affect performance of reviewers. In order to better understand these influencing factors of review performance we need direct measures. Additionally, if some defects are easier to be found, then there should be a measure to be used for detection difficulty level for a specific defect. In the literature no specific measures have been found to quantitatively describe the defect detection performance of an individual and defect detection difficulty level of a specific defect. This study proposes two measures for the defect detection difficulty level (DF) and defect detection performance of a practitioner (PP) by analyzing the defect detection process of participants.

During software inspection process, several reading approaches were studied, like Scenario-based reading, Ad hoc and checklist-based reading (Laitenberger and Atkinson, 1999; Rombach et al., 2003). Rombach et al. (2003) compared Ad hoc and checklist-based reading techniques in their study. It is pointed that there may be a potential connection between ERD reading approach and the inspection performance (Rombach et al., 2003). This impact can be explored comprehensively by analyzing eye-tracker data of software engineers' defect detection process. Hence this study also explores successful defect search techniques of software engineers.

3. Methods

Main research questions of this study are:

1. Are there any measures to measure difficulty level of defects seeded in the ERD and performance of the software engineers in ERD defect detection process?
2. Is there a relationship between the search patterns of software engineers and their defect detection performance?

To answer these questions, this study is designed as a qualitative research to get better feedback from the insights of the participants' review processes. The qualitative research, which is inductive and exploratory (Glaser and Strauss, 1967; Bogdan and Biklen, 1992), deals with the findings from data in their natural settings, which is well suited to the purpose of this study. Contrary to quantitative research's approach, which summarizes large amounts of data, dealing with many subjects and reaching generalizations based on statistical projections, qualitative research excels at "*telling the story*" from the participant's viewpoint, providing the rich descriptive detail that sets quantitative results into their human context (Merriam, 1998, p. 5). Similarly, this qualitative study focuses on the perspectives of the participants of the study in order to uncover the complexity of human behavior in such a framework and present a holistic interpretation of what is happening in this context. Therefore, in this study the critical point is not the number of subjects but the depth and the quality of presenting the subjects' learning experience with the system. From Donmoyer's (1990) and other qualitative researchers' articles, it is seen that the best way to understand any phenomenon is to view it in its context. Nielsen and Landauer (1993) also report that the best usability results come from testing no more than four or five users and this number of users are enough to explain more than 80% of the phenomena. They report that after this point, the same findings start to be observed repeatedly. In this study, the participants' behaviors are analyzed in depth from different dimensions. Each participant is studied individually. The data is collected by using different methods. In the following subsections, the participants, the apparatus used, the materials that are provided to the participants, the procedure that is implemented as well as the analyses techniques used in this study are described. However, in order to better understand the correlation between the proposed measure values and eye-tracker data experiment-2 is conducted with another five participants with the same procedure. Information about the participants and results of experiment-2 are provided in Section 5.5.

Table 1
Participant information.

Participant	Years in the field	Experience with ERD
P_1	8	8
P_2	7	2
P_3	12	2
P_4	12	3
Average	9.75	3.75

3.1. Participants

Participants have been selected from the same company. There were four professional software engineers in the company who were native Turkish speakers and all of them were voluntarily participated in this study.

All the participants all have at least 7 years of experience in the field of software development. The company produces software systems like personnel management system, stock management, accounting and some specific software based on analyzing huge amount of data for providing information for the decision makers. In that context, the first participant (P_1) has declared that she has been involved in four projects having a database component as a software developer. The second participant (P_2) also declared that he has been involved in one project as database designer and user interface developer. P_3 reported that he has been involved in two big projects in the company as database designer and software developer. P_4 declared that he has been involved in at least ten such projects having database components as a software developer. They all declared that they were using ER diagrams to better understand the database design and develop the software accordingly. All participants were holding a university degree from the computer engineering departments of different universities. Table 1 below summarizes the information about the participants.

3.2. Apparatus

The apparatus consisted of an IBM-compatible PC with stereo speakers and a 17-inch monitor with a resolution of 1024×728. A Tobii Eye Tracker integrated into the casing of the monitor collected the eye movement data of the participants non-intrusively. This binocular tracker has a field of view of approximately $20\,\text{cm} \times 125\,\text{cm} \times 20\,\text{cm}$ (width × height × depth) and an effective tolerance for free head-motion of about $30\,\text{cm} \times 15\,\text{cm} \times 20\,\text{cm}$ at 60 cm distance. The tracking system has a 50 Hz sampling rate with an accuracy of 0.5°. A fixation was defined as one or more gaze points within a circular area of 30 pixels for a minimum duration of 100 ms.

3.3. Materials

3.3.1. System description document

The material used in this research was adapted from the work of Hungerford et al. (2004). The settings were modified for use in this current study and translated into Turkish.[2] The English version of the document was also prepared for the readers.[3] The system description document was included in the system requirements.

3.4. System ERD

Fig. 1 shows the prepared entity relationship diagram (ERD) of the system with seeded defects which were also adapted from the work of Hungerford et al. (2004). The circled areas in the figure show the id of each defect seeded in the ERD.

Using the open source ER-Assistant tool (ERD Tool, 2010), a total of 17 defects were seeded in the ERD. The English version of this ERD was also prepared for the readers.[4]

The defects seeded in the ERD, shown in Table 2 above, can be grouped into four as follows: (1) *Missing an entity or relationship* that is described only in the system description document but missing in the ERD; (2) *Incomplete/incorrect information* type defects that are described in the system description document differently from the ERD representation, and defects that are related to the *super class/sub class* structures; (3) The defects on the *cardinality* or *optionality* definitions of the relationships that are represented differently in the ERD compared with the system description document and; (4) The defects *containing extra information* in the ERD which is not described in the system description document.

3.5. Validity and reliability

In order to enhance the content validity of the prepared ERD system that is seeded with 17 defects, three ERD experts analyzed the system description document and the ERD without defects before the experiment. The experts were one professor and two software engineers each having 10 years experience with ERD in the field and according to their comments the seeded defects and their locations as well as the system description documents were updated. Pilot testing was implemented on the instruments to obtain feedback about the organization of the experiment and to determine if there were any problems. The pilot test was undertaken by two software engineers, one with 3 years experience with ERD and the other with 5 years experience. Further revisions were made to the experimental set-up and environment based on the testers' comments.

[2] http://www.atilim.edu.tr/~nergiz/erd_desc_TR.pdf.
[3] http://www.atilim.edu.tr/~nergiz/erd_desc.pdf.
[4] http://www.atilim.edu.tr/~nergiz/erd_eng.pdf.

3.6. Procedure

One week before the experiment, each participant was informed about the study and was provided the system description document, so that they understood the system requirements of the study material prior to the experiment. The study was conducted in the Human-Computer Interaction Laboratory at Middle East Technical University. Before the experiment, eye movements of each participant were calibrated automatically by Tobii Clear View software with five fixation points. The quality of the calibration was checked by examining the calibration plot. Recalibration took place when the calibration data was poor or missing. During the experimental study, each participant was tested individually in a single session, asked to study the system and try to find the defects in the ERD (shown through the eye tracker monitor) using the system description document. No additional instructional materials were provided. The participants were informed about the number of defects in the diagram and they were asked to think aloud during the experiment. For each participant, the Clear-View program provided a time-sampling list of fixations that included the duration and spatial location of each eye fixation in *xy* coordinates. During the experiments, video-audio recordings and behavior of the participants in the test room were collected. An interview was conducted with each participant immediately after the experiment and the conversation was recorded.

3.7. Data analysis

Four different groups of data were collected during the experiment. The collected data was then analyzed by building correlations among different behaviors of the participants. Fig. 2 summarizes how these data were used and analyzed in this study. *Observation Data* is the two observers' notes about the behaviors' of the participants. *Video-audio Recording* is the video and audio recordings of the participants during the experiment. This recording shows the verbal descriptions of each participant as well as their behavior during the experiment. During the experiment each participant was asked to note down the defects they have found and their descriptions on a paper. These *written protocols of each participant* were collected to be used for further analysis. The three data sources namely the observation data, video-audio recordings and written protocols of the participant were analyzed together to calculate D_{pij} which is the time taken by participant i (P_i) to discover defect j, and O_{pij}, which is the defect detection order of defect j by participant i (P_i). These variables were used to develop the proposed formulas for the defect difficulty level (DF) and the defect detection performance of the participants (PP).

Eye-Tracker *fixation* and *duration* data were collected by Tobii eye-tracker device which shows the eye fixation durations of the participants on the defected area of the ERD as described in Fig. 1. *The audio recordings of the interviews* that were established after each experiment were transcribed for the analyses. These two data sources were used to validate the proposed formulas for DF and PP. Finally, the Eye-Tracker eye-gaze video data was collected by the Tobii eye-tracker, which shows the video of all the eye-gazes of the participants during the experiment. These video recordings were examined to discover the search patterns (SP) that each participant used during the defect detection process.

4. Proposed formulas

In order to measure the defect detection difficulty level for each defect seeded in the ERD, the following assumptions were taken into account:

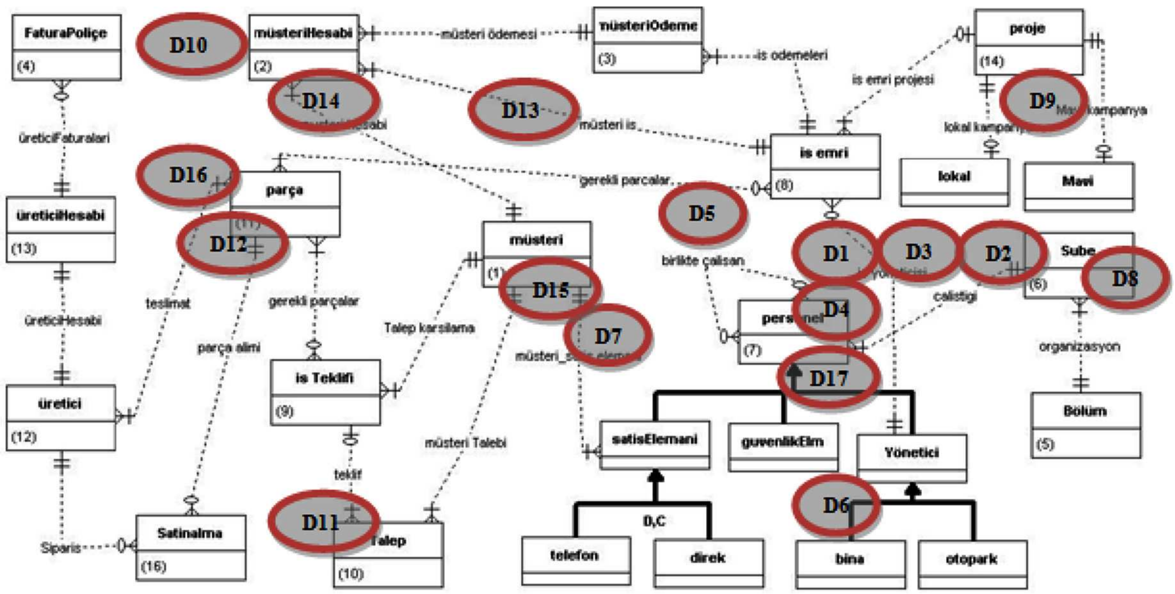

Fig. 1. ERD of the system.

Table 2
Defect details.

Defect	Description	Defect type
1	There should be an entity named “Ofis” (*Eng. Office*)	Missing Entity
2	There should be an entity named “telefonNo” (*Eng. phoneNo*)	Missing Entity
3	There should be a relationship between (Ofis × telefonNo)	Missing Relationship
4	There should be a relationship between Ofis × personel (*Eng. employee*)	Missing Relationship
5	There should be a relationship between musteri (*Eng. customer*) × isEmri (*Eng. workOrder*)	Missing Relationship
6	The entities “Bina” (*Eng. buildings)* and “otopark” (*Eng. carparks*) should be connected to the entity “guvenlikElm” (*Eng. securityGuard*) not to “Yonetici” (*Eng. manager*)	Super/Sub Class
7	The cardinality of the relationship between entities “satisElemani” (*Eng. salesman*) and “musteri” should be MxN	Cardinality
8	The entity “Sube” (*Eng. unit*) should be a weak entity	Wrong entity type
9	“Mavi” (*Eng. Mavi*) and “lokal” (*Eng. local*) should be sub-classes of an entity named “tanitim” (*Eng. advert*) which has a relationship with “proje” (*Eng. project*)	Missing Super/Sub Class
10	There should be a relationship between “musteriHesabi” (*Eng. customerAccount*) and “FaturaPolice” (*Eng. invoice*)	Missing Relationship
11	The cardinality of the relationship between “talep” and “isTeklifi” (*Eng. jobProposal*) should be 1 × 1	Cardinality
12	The cardinality of the relationship between “Parca” (*Eng. Part*) and “satinalma” (*Eng. purchase*) should be MxN	Cardinality
13	No need to show the relationship musteriIs. This information can be obtained through the relationships isEmri × musteriOdemesi (*Eng. custPayment*) and musteriOdemesi (*Eng. custPayment*) × musteriHesabi. This creates redundancy in the database.	Extra info
14	The optionality of the relationship musteriHesabi × musteri of the musteriHesabi side should be optional, not required.	Optionality
15	The optionality of the relationship musteri × satisElemani of the musteri side should be optional, not required	Optionality
16	The optionality of the relationship uretici (*Eng.supplier*) × Parca of the “parca” side should be optional, not required	Optionality
17	The sub-class connections of the “Personel” super class should be defined as disconnected (d) and complete (c).	Super/Sub Class

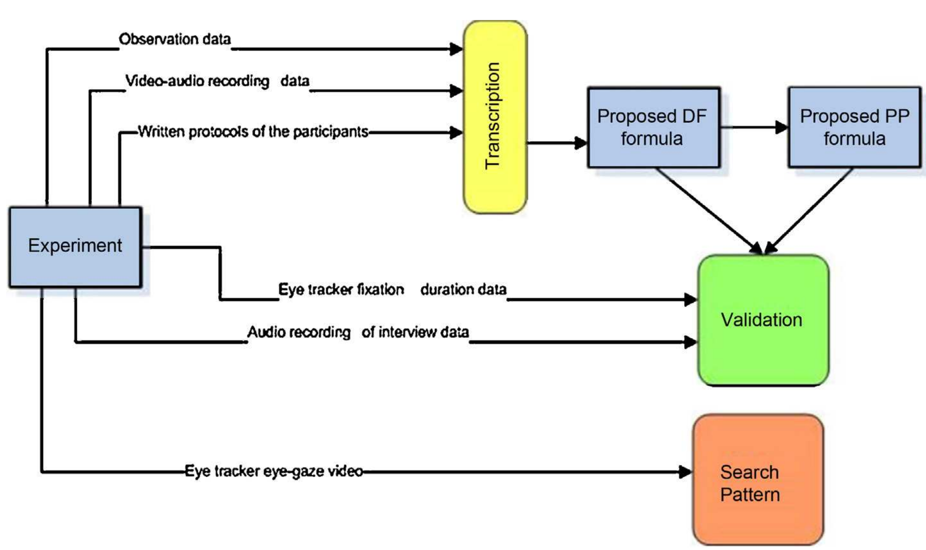

Fig. 2. Research procedure.

1. The difficulty in detecting a defect is directly proportional to the time spent identifying the defect; if someone spends more time finding a defect, then it is harder to recognize than a defect that is detected in a shorter time.
2. The difficulty in detecting a defect is directly proportional to the recognition order of the defect; the most easily detectable defects are recognized first.
3. The difficulty in detecting a defect is inversely proportional to the number of people who recognized the defect; if a defect is detected by all the participants, then it is easier than a defect that is detected by fewer people.

There are several measures used to measure information retrieval of subjects; like precision and recall which measures subjects' imprecision or incompleteness in retrieving past information (Genero et al., 2008). However, in our study, participants are not asked to retrieve any past information but they are asked to find the defects in the ERD according to the description of the case. That is why these measures in the literature are not implemented into the formulas used in this study. Moreover, in assessment studies, there are also other factors considered in measure development like experience and ability of subjects performing specific tasks. However, determining experience of a subject in a particular context is not easy and straight forward, which cannot be measured through years of familiarity because it is not possible to determine how dense the subject came across with that concept during years of study. Therefore, in this study familiarity and ability of the participants have been included through two factors: time spent for detecting a defect and identifying the defect at the first place or at a later time. From these assumptions, the defect detection difficulty level measure is formulated as (1).

$$\text{Defect Detection Difficulty Level Formula,} \quad \mathrm{DF}_j = \frac{\mathrm{D}_j \bullet \mathrm{O}_j}{\mathrm{S}_j} \qquad (1)$$

where DF_j is the defect detection difficulty level of the jth defect, D_j is the average duration spent by all participants for finding defect j, O_j is the average score of all participants for detecting jth defect and S_j is the success rate of detecting defect j (number of people who detected defect j/total number of participants).

In this formula O_j refers to the score gained from the defect detection order. For example, if O_j is 1, it means that the jth defect has been detected at the first place, meaning it is the earliest among 17 defects; hence, it is an easier defect to find and the participant gets a lower score for detecting that defect. Similarly, if O_j is 11, the defect is detected later in the detection process, therefore that defect is harder to find and the participant gains a higher score from detecting it. The defect detection order may also be related to the order in which the information about the defect is presented in the system description document. However, since this document was given to the participants one week before the experiment, they were already familiar with the concepts. For this reason, this affect was ignored in the study.

Using the defect detection difficulty level value of each defect (DF_j), a performance measure formula was developed to describe the defect detection performance of each participant i (PP_i) (see (2)).

$$\text{Defect detection performance formula,} \quad \mathrm{PP}_i = \frac{\sum_{j=1}^{n} \mathrm{DF}_j}{\sum_{k=1}^{s} \mathrm{DF}_k} \qquad (2)$$

where PP_i is the defect detection performance of the ith participant, DF_j is the difficulty level of the jth defect calculated by formula (1), n is the total number of defects detected by participant i and s is the total number of defects seeded in the ERD.

It can be seen from formula (2) that defect detection performance of participant i (P_i) is the proportion of the cumulative difficulty level of defects detected by P_i, to the cumulative difficulty level of all the defects seeded in the ERD.

5. Results

The analysis of the experiment results is provided in the following sections in terms of difficulty levels of the defects, participants' defect detection performance and their search patterns. The results of this study are also validated through the second experimental study. The results of experiment-2 are also discussed at the end of this session.

5.1. Defect detection difficulty level (DF)

Table 3 illustrates the data used to obtain DF values. The following values were taken from the observation data; durations ($\mathrm{D}_{\mathrm{p}ij}$) represent each participant's (P_i) time spent (in seconds) in detecting each defect j (D_j). Therefore, the $\mathrm{D}_{\mathrm{p}ij}$ values differ from the eye fixation duration data obtained by the eye-tracker, which only considers the eye fixation duration values on the defected areas. For instance, from the cell in the second column and first row of Table 3, it is understood that participant 1 (P_1) spent 142 s in order to detect defect 1 (D_1). D_j in Table 3 shows the average $\mathrm{D}_{\mathrm{p}ij}$ values of each defect j. For instance, average $\mathrm{D}_{\mathrm{p}i1}$ was 87 for defect 1 (D_1).

The defect detection order values ($\mathrm{O}_{\mathrm{p}ij}$) were calculated from the observation data. These values show the order in which a participant P_i detects each defect j. For example, from Table 3 it is seen that participant 1 detected defect 6 first ($\mathrm{O}_{\mathrm{p}16}$). O_j in Table 3 shows the average of $\mathrm{O}_{\mathrm{p}ij}$ values for each defect j. R_j is the number of people who detected defect j. As a result, by using these values in formula (1), defect detection difficulty level (DF_j) values were calculated for each defect j as shown in Table 3. The corresponding DF values for undetected defects were left empty.

As seen in Table 3, defects 2 and 10 were never detected. Hence, they do not have difficulty level values. During the interviews, with respect to defect 10, P_2 explained, "I think the most difficult error was related to faturaPolice (Eng. invoice). I could not understand how I should relate faturaPolice to musteriHesabi (Eng. custAccount)". From this explanation, even P_2, who performed better than the other participants, recognized that there should be a defect in that area, but he could not identify it.

The difficulty level of defects 11, 12, 15 and 16 were higher compared to the others. These defects are classified as harder to recognize. According to Table 3, the most easily detected defects are defect 06 ($\mathrm{DF}_{06} = 113$) and defect 17 ($\mathrm{DF}_{17} = 280$). The qualitative data collected during the interviews performed with each participant also supported this result. During the interviews, three participants specifically declared that they recognized those defects easily. For instance, P_1 stated, "I found the bina (*Eng. buildings*) otopark (*Eng. carParks*) subclass/super class error easily, because of the bold line." Similarly, P_2 said, "The super-class sub-class errors were much easier for me." P_2 also declared that "While studying the ERD, I first started from the bold lined areas, which were easy for me to understand". Hence, a bold effect is an effective factor in recognizing those defects easily. During the experiments, the defects located in the bold parts of the diagram (D_{06} and D_{17}) were the most easily detected ones in Fig. 1 (connections of entities such as Personel [*Eng. Employee*], satisElemani [*Eng. salesman*], guvenlikElm [*Eng. securityGuard*] and Yonetici [*Eng. manager*]). During the interviews, the participants also declared that they easily recognized those defects because of the bold representation. It is also important to report that the second group of defects that were detected easily (D_{03}, D_{07}, D_{01}, D_{04} and D_{05}) were also connected to this bold represented area. Only exception was D_{02} which is considered to be the missing information type of

Table 3
Calculated defect difficulty level values.

Defect	D_{p1j}	D_{p2j}	D_{p3j}	D_{p4j}	D_j	O_{p1j}	O_{p2j}	O_{p3j}	O_{p4j}	O_j	R_j	DF_j
01	142			31	87	2			6	4	2	692
02												–
03		50		30	40		2		7	4.5	2	360
04	198	27			113	5	3			4	2	900
05	50	163			107	3	10			6.5	2	1385
06	34	87	36	43	50	1	4	2	2	2.25	4	113
07			67	68	68			4	3	3.5	2	473
08		105		270	188		6		4	5	2	1875
09	75	300	297	448	280	4	11	3	5	5.75	4	1610
10												–
11		484	240		362		7	5		6	2	4344
12			156	245	201			6	8	7	2	2807
13		130	165		148		8	8		8	2	2360
14		46			46		9			9	1	1656
15		176			176		5			5	1	3520
16			186		186			7		7	1	5208
17		45	165	420	210		1	1	1	1	3	280

defect related to another missing entity in the ERD called Ofis [*Eng. Office*](D_{01}).

5.2. Correlation between defect difficulty and participants' eye fixations and durations

Parallel to the observational and audio-video recordings data, eye fixation durations were also recorded by the eye tracker. According to this data, average time taken for the experiments was 48 min 25 s (SD = 8 min 49 s). The average length of eye durations of participants was 31 min 10 s (SD = 7 min 20 s), and the average number of eye fixations was 1830 (SD = 1078). Table 4 summarizes the fixation duration values (FD_{pij}) of each participant i (P_i) on the defect area of the ERD for each defect j. For example, on the location of defect 02 area in the ERD, P_1's fixation duration (FD_{p12}) was 5902 s.

The studies in the literature show that the mean fixation duration values are higher for more demanding tasks (Loftus and Mackworth, 1978; Underwood et al., 2004). Based on this assumption, if the proposed defect detection difficulty level formula is accurate, then there should be a correlation between these values and the related eye fixation durations. In other words, a defect that has higher defect difficulty measure (DF) value should also have higher eye fixation duration (FD) value. In order to test this proposition, statistical analyses were conducted in which non-detected defects were excluded. The analysis was conducted for 60 different pairs of defect difficulty level values and the related eye fixation duration value of each participant (15 detected defects by 4 participants). Fig. 3 shows the scatter plot of these two measures.

The relationship between the defect difficulty level measure of defect j (measured by DF_j) and the eye fixation duration measure for the defect area (as measured by FD_j) was investigated using the Pearson product-moment correlation coefficient. Preliminary analysis was performed to ensure there were no violations of the assumptions of normality, linearity, and homoscedasticity. There was a statistically significant correlation between the two variables ($r(60) = 0.309$, $p < 0.01$). This result shows that when the mean eye fixation duration value on the area of a specific defect is longer, the difficulty level of that task is higher. Thus, the calculated DF value for that defect is larger.

Table 4
Defect difficulty level and fixation durations.

Defect(j)	DF_j	FD_{p1j}	FD_{p2j}	FD_{p3j}	FD_{p4j}
01	692	1316	2510	5201	6339
02	–	5902	4823	14,031	6020
03	360	876	1992	7475	3647
04	900	5242	12,574	11,165	28,847
05	1385	17,307	18,512	36,758	7477
06	113	3747	7273	3210	14,394
07	473	8415	19,493	24,181	4605
08	1875	2951	10,664	22,703	13,736
09	1610	3428	9904	44,457	5246
10	–	8394	12,019	18,341	3809
11	4344	5621	6183	46,706	12,199
12	2807	10,999	5023	42,325	6658
13	2360	13,539	14,690	41,414	7336
14	1656	22,826	35,315	37,860	7299
15	3520	15,093	25,928	36,478	24,821
16	5208	14,751	4882	80,915	1795
17	280	3887	17,260	7458	31,421

5.3. Defect detection performance of a participant (PP)

The defect detection performance of participant i (PP_i) was calculated using formula (2). For normalization, the resulting values were multiplied by 100. The calculated PP_i values are shown in Table 5. Additionally, eye fixation duration performance of each participant i (FD_{pi}) were calculated according to (3). For normalization, the results were multiplied by 100 as shown in Table 5.

$$\text{Fixation duration performance calculation,} \quad FDP_i = \frac{\sum_{j=1}^{n} FD_{ij}}{\sum_{k=1}^{s} FD_{ik}} \tag{3}$$

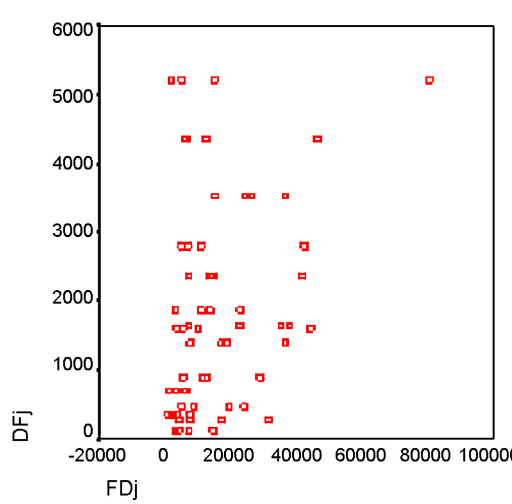

Fig. 3. Scatter plot of DF and eye fixation duration.

Table 5
Participant performance (PP_i) and fixation duration performance (FDP_i).

Participant	PP_i	FDP_i
P_1	17	21.51
P_4	30	46.35
P_3	62	60.47
P_2	67	76.68

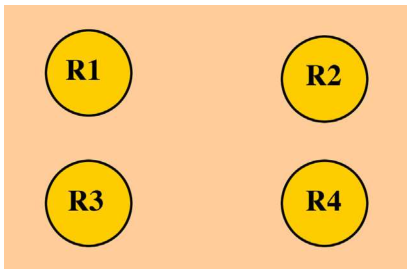

Fig. 4. ERD regions.

Table 6
Eye movements among the ERD regions.

Participant	R1xR2	R1xR3	R1xR4	R2xR3	R2xR4	R3xR4
P_1	167	92	19	16	59	24
P_2	90	132	41	22	91	24
P_3	79	21	16	2	128	9
P_4	36	72	65	20	66	58

where FDP_i is the fixation duration performance of participant i, FD_{ij} is the fixation duration value of participant i over the area of defect j, n is the total number of defects detected by participant i and s is the total number of defects seeded in the ERD

When the PP_i and FD_{Pi} values are compared, as seen in Table 5, the ranks for these two parameters are the same for all the participants (P_1, P_4, P_3 and P_2), which indicates that as the participants' performance value increases, so do their eye fixation durations.

5.4. Correlation between software practitioners' defect detection performance and their search patterns

In order to understand the participants' search patterns in detecting the defects in the ERD, the gaze data obtained from the eye-tracking device were analyzed. For this purpose, ERD diagram screen was divided into four regions as shown in Fig. 4.

The participants' eye movements between these regions were analyzed. The defects were almost evenly distributed in all regions as shown in Fig. 1. For this analysis, Table 6 shows the eye movement values of each participant between the ERD regions as defined in Fig. 4. As seen from the table, participant 1 (P_1) had the highest number of eye movements between Region 1 and Region 2 (R1xR2).

Normalized eye movements were calculated by dividing each participant's eye movement data at a specific region by his/her total eye movements; results are shown in Table 7.

The horizontal and vertical eye movements of the participants were analyzed between the ERD regions. The horizontal eye movements were calculated from the movements between regions R1xR2 and R3xR4. The vertical eye movements were calculated from the movements between regions R1xR3 and R2xR4. From the eye movement graph, it is clear that P_1 mostly performed horizontal movements (between regions R1xR2 and R3xR4), whereas P_2 performed an almost even number of horizontal and vertical movements. In order to have a more in depth view of the eye movements in the regions, the eye movement ratios between the vertical (V-summation of the eye movements between the regions R1xR3 and R2xR4) and horizontal (H-summation of the eye movements between the regions R1xR2 and R3xR4) regional eye movements were calculated as shown in Table 8. For the normalization, the resulting SP (V/H) values were multiplied by 100.

As seen from Table 8, the rank according to the calculated eye movement values of the participants; V/H (P_1, P_4, P_3 and P_2) is the same as the rank of participant performance (PP_i) and fixation durations (FD_i) values. In other words, the participants who search the ERD vertically get higher performance scores which also indicates that horizontal search in the ERD lowers their performance. Table 9 below summarizes the findings of the defect detection performance of participants (PP_i), their fixation durations performance (FDP_i) values and vertical over horizontal search (V/H) values.

From Table 9, it is concluded that the proposed defect detection performance of the software engineers (PP_i), their fixation duration performance values on the defected area (FDP_i), and their vertical over horizontal search pattern (V/H) values have the same tendency. This means that when the participant's defect detection performance is high, so are their fixation durations on the defected area and V/H values. In other words, participants who search the ERD vertically get higher performance scores which means that horizontal search in the ERD lowers their performance. In all measures, P_2 was the most successful participant, followed by P_3, then P_4 and P_1.

Table 7
Normalized eye movements among the ERD regions.

Participant	R1xR2	R1xR3	R1xR4	R2xR3	R2xR4	R3xR4
P_1	0.44297	0.24403	0.05040	0.04244	0.15650	0.06366
P_2	0.22500	0.33000	0.10250	0.05500	0.22750	0.06000
P_3	0.30980	0.08235	0.06275	0.00784	0.50196	0.03529
P_4	0.11356	0.22713	0.20505	0.06309	0.20820	0.18297

Table 8
Normalized eye movements among the ERD regions.

Participant	Horizontal (H)	Vertical (V)	SP (V/H)
P_1	0.50663	0.40053	79.058
P_4	0.29653	0.43533	146.81
P_3	0.34510	0.58431	169.32
P_2	0.28500	0.55750	195.61

Table 9
Summary of results.

Participant	PP_i	FDP_i	V/H
P_1	17	21.51	79.058
P_4	30	46.35	146.81
P_3	62	60.47	169.32
P_2	67	76.68	195.61

5.5. Experiment-2

In order to validate our results we have conducted the same experiment with five participants as shown in Table 10.

Table 10
Participant information for validation experiment.

Participant	Years in the field	Experience with ERD
P_1	3	0
P_2	2	1
P_3	3	0
P_4	2	1
P_5	2	2
Average	2.25	1

Table 11
Calculated defect difficulty level values for the validation experiment.

Defect	D_{p1j}	D_{p2j}	D_{p3j}	D_{p4j}	D_{p5j}	D_j	O_{p1j}	O_{p2j}	O_{p3j}	O_{p4j}	O_{p5j}	O_j	R_j	DF_j
01	74	196	155		130	138	1	3	1		1	1,50	4	260
02														
03														
04														
05		283		101		192		6		4		5,00	2	2400
06		248	54	190	25	129		1	2	1	2	1,50	4	242
07	155	216	477	744		398	2	4	3	2		2,75	4	1368
08														
09					302	302					3	3,00	1	4530
10														
11														
12				896		896				5		5,00	1	22,400
13	494					494	3					3,00	1	7410
14		365		572		469		5		3		4,00	2	4685
15														
16														
17		126				126		2				20	1	1260

The participants of the second experimental study have lower experience years in the field (2.25 years on average) than that of original study (9.75). Similarly their experience with ERD (in average 1 year) is lower than that of the original study (3.75). Table 11 shows the calculated Defect Difficulty Level Values for the Validation Experiment.

In the original study, when we order the defects according to their DF values, the defects 2, 8, 10, 11, 12, 13, 15 and 16 were either couldn't be detected or assigned higher DF values. As seen from Table 11, the situation is similar for the second experiment. In the second experimental study, the defects 2, 3, 4, 8, 10, 11, 15 and 16 cannot be detected and the defects 12 and 13 are assigned higher DF values which are detected by only one participant. In the validation study, additionally the defects 3 and 4 cannot be detected where as they were detected in the original study. This may be because of the higher experience level of the original study participants. Hence, the number of defects that cannot be detected in the second experiment (8 defects) is also higher than that of the original experiment (2 defects). Additionally, defect difficulty level and fixation durations values for the second experiment are also analyzed. In this experiment the participants' eye fixation duration values were as in Table 12.

To understand if there is a correlation between DF_j and FD_j values, an analysis was conducted for 45 different pairs of defect difficulty level values and the related eye fixation duration value of each participant (9 detected defects by 5 participants). The relationship between the defect difficulty level measure of defect j (measured by DF_j) and the eye fixation duration measure for the defect area (as measured by FD_j) was investigated using the Pearson product-moment correlation coefficient. Preliminary analysis was performed to ensure there were no violations of the assumptions of normality, linearity, and homoscedasticity. There was a statistically significant correlation between the two variables ($r(45) = 0.525$, $p < 0.01$). This result shows that when the mean eye fixation duration value on the area of a specific defect is longer, the difficulty level of that task is higher. Thus, the calculated DF value for that defect is larger. The results of this validation study are parallel to the original study which shows that the proposed formulas are working very well.

Table 12
Defect difficulty level and fixation durations for the validation experiment.

Defect(j)	DFj	FDp1j	FDp2j	FDp3j	FDp4j	FDp5j
01	260	1316	2510	5201	6339	1316
02		5902	4823	14,031	6020	5902
03		876	1992	7475	3647	876
04		5242	12,574	11,165	28,847	5242
05	2400	17,307	18,512	36,758	7477	17,307
06	242	3747	7273	3210	14,394	3747
07	1368	8415	19,493	24,181	4605	8415
08		2951	10,664	22,703	13,736	2951
09	4530	3428	9904	44,457	5246	3428
10		8394	12,019	18,341	3809	8394
11		5621	6183	46,706	12,199	5621
12	22,400	10,999	5023	42,325	6658	10,999
13	7410	13,539	14,690	41,414	7336	13,539
14	4685	22,826	35,315	37,860	7299	22,826
15		15,093	25,928	36,478	24,821	15,093
16		14,751	4882	80,915	1795	14,751
17	1260	3887	17,260	7458	31,421	3887

6. Discussions

Results of this study show that *use of bold effects in the ERD is an effective factor for detecting the defects.* Usually, software engineers easily detect the defects in bold defected area and they start searching the ERD from this area. This finding supports Glasgow et al.'s study which claims that the use of bold effects in possibly defected areas might enhance the diagrammatic reasoning because it affects visual perception routines (Glasgow et al., 1995). They also report that use of bold effects in the ERD is an effective factor for detecting the defects. Usually, software engineers easily detect the defects in this area and they start searching the ERD from this area (Glasgow et al., 1995).

Results of this study also show that *detecting missing information type defects in the ERD is harder than detecting other defects.* Two of the missing information type defects were not detected at all by any of the participants (D_2: a missing entity type defect and D_{10}: a missing relationship type defect). As Wohlin and Aurum (2003b) also reports, the defects found by fewer reviewers have a lower detection probability, which in return means that those defects are harder to recognize. No matter what the type of the missing information was, an entity, an attribute or a relationship, it could be difficult to detect even for the experienced software engineers. The other defects that were detected by some participants and considered to be missing information type of defects (D_1, D_3, D_4 and D_5), are all in the close proximity of the bold represented area of the ERD. This might be the reason for the detection of these defects. Even though D_2 can also be considered to be in the close proximity of this bold represented area, since it is related to another missing entity, it gets more complex and could not be detected by any of the participants.

Additionally, *defining and identifying missing information in the ERD is harder than recognizing it.* Usually, the participants declared that there should be something missing around the defected area that contains missing information, but they could not identify it. This finding is supporting Hungerford et al.'s study (2004), which reports that detecting missing information and relationships is easier than deciding the representation of the missing information in the ERD.

Another finding of this study shows that *the defects that are related with the cardinality and optionality of the relationships are harder to detect.* The defects D_{16}, D_{11}, D_{15}, and D_{12} were rarely detected and they were all related to the cardinality or optionality definitions of the relationships.

Wohlin and Aurum (2003b) have reported that strong reviewers indicate few false positives while the not-so-strong reviewers (relative to the others) are expected to indicate either few of each or many of each. In our experiment, the most successful participant P_2, who had the highest score in discovering the defects compared to the other participants, did not report any false positive defect whereas each of the other participants reported one false positive.

The findings also show that *as the participants' eye fixation durations raises so do their performance value.* As a result of the experimental work, a statistically significant correlation is found between the calculated defect detection difficulty level values and the eye fixation durations which indicate that as *the difficulty level of a defect grows, participants spend more time viewing the defected area* (higher eye fixation duration values) which supports Rayner's study (1998). Additionally, we understood that when the mean eye fixation duration for the area of a specific defect is longer, the difficulty level of that task is higher which is parallel to the previous findings. This finding is supportive to Knoblich et al.'s study (2001) which indicates that the duration between eye fixations refers to impasse, a state of mind before making a decision in which the problem solver feels that all options have been explored and he or she cannot think of what to do next. Loftus and Mackworth (1978) and Underwood et al. (2004) also reported similar results. Thus, the longer impasses indicate that participants are about to solve the problems.

Finally, the participants who searched the ERD vertically performed better in defect detection process. Similarly, the participants who searched the ERD horizontally performed worse in defect detection process. This finding is supportive to Hungerford et al.'s (2004) study which concludes that horizontal reading leads to identification of defects from ambiguity and inconsistency, while vertical reading aids the identification of defects of omission and incorrect fact. Since the number of errors seeded in the ERD mainly consists of missing entities and incorrect information, vertical readers performed better in the experiment. Defect detection performance before the implementation would accelerate the project development. This will eventually reduce the project costs. Hungerford and our study underlines that the discovery of defect types is related to the search patterns. Vertical searches are better to discover the defects in our study. Controversially, it might result from that the participants' native language is written vertically and therefore they are accustomed to vertical searches.

7. Conclusion

Main motivation of this study was to find measures to better understand the software engineers' reasoning process of ERD. It is expected that such measures may provide several benefits for the researchers, software companies and educators. Our first research question was:

Are there any measures to measure difficulty level of defects seeded in the ERD and performance of the software engineers in ERD defect detection process?

As an answer to this research question, we have proposed a new measure for defining the difficulty level of a defect (DF) in an ERD. Based on this measure, a measure for the defect detection performance of the software engineers (PP) was also proposed. By using these measures the following results were concluded from this study:

1. Use of bold effects in the ERD is an effective factor for detecting the defects. Usually, software engineers detect the defects in the bold defected area easily and they start searching the ERD from this area.
2. Detecting missing information type defects in the ERD is harder than detecting other type of defects.
3. Defining and identifying missing information in the ERD is harder than recognizing it.
4. The defects that are related with the cardinality and optionality of the relationships are harder to detect.
5. Strong reviewers indicate few false positives while the not-so-strong reviewers (relative to the others) are expected to indicate more.

Second research question of the study was:

Is there a relationship between the search patterns of software engineers and their defect detection performance?

In order to answer this question, eye movement data and search patterns of the participants were analyzed. Accordingly, following results are concluded:

6. Participants who searched the ERD vertically mostly, performed better in the defect detection process. Similarly, the participants who searched the ERD horizontally mostly, performed worse in defect detection process.
7. As the participants' eye fixation durations raises so do their performance values.

It should be also noted that the conclusions above which were developed by using the proposed measures are very supportive to the findings of other studies found in the literature. We believe that this is a very good evidence that the proposed measures are working correctly and they are also validated through the results of other studies.

Besides the answers to the research questions, the results of this study also provide several benefits for the researchers, software companies and educators:

- Researchers may use the proposed measures to understand and analyze the reasoning process of software engineers during ERD review process. The proposed measures can be used in further diagrammatic experiments to set initial difficulty values for the defects or to validate the findings and participants' performances.
- Researchers may also use these measures in order to better understand the design issues and cognitive processes in reasoning of other diagrammatic representations of information used in different domains.
- The proposed measures can be employed in defect detection experiments where eye-tracking data is not available.
- It could be possible to improve the ERD notations and their quality using these measures through analysis of several different design approaches in terms of different notations, colors, naming strategies, drawing strategies as well as strategies for locating the information in the diagrams.
- These measures can be used to develop guidelines for the ERD designers to represent information better and to minimize the defect ratios of these designs, which will in return decrease the software project costs.

- In a software development project, the quality and appropriateness of the designed ERD can be measured through these measures for research purposes.
- During the review process of an ERD representation, depending on the types of the defects, a systematic search pattern composed of both vertical and horizontal directions instead of a single concurrent one should be preferred. It is advisable to divide the diagram into subsections corresponding to the design scenario and apply this systematic search pattern for each subsection for an effective defect detection strategy in the review phase of the software project.
- By using proposed measures, the educators in higher education organizations providing occupational education in this field as well as the in-service training providers can improve the learners' understanding level of these diagrams and improve their skills for the ERD review process.

In order to further validate the findings of this study, the experiments could be applied to the groups of engineers other than the design team. As a future work, comparative study of different ERD representations could be explored to improve the quality of these representations. Data requirement specifications can be correlated with the ERD representation by using colors or frames and their affect on cognition would be examined. As a result, these measures can be used as a research tool to develop and improve information representation process for the diagrammatical representations in future studies.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jss.2013.03.106.

References

Akoka, J., Comyn-Wattiau, I., Lammari, N., 1999. Relational database reverse engineering, elicitation of generalization hierarchies, advances in conceptual modeling. In: ER'99 Workshops on Evolution and Change in Data Management, LNCS 1727, Paris, France, November 1999, pp. 173–185.

Bock, D.B., Ryan, T., 1993. Accuracy in modeling with extended entity relationship and object oriented data models. Journal of Database Management 4 (4), 30–39.

Bogdan, R.C., Biklen, S.K., 1992. Qualitative Research for Education: An Introduction to Theory and Methods, 2nd ed. Allyn and Bacon, Needham Heights, MA.

Bouzeghoub, M., Rochfeld, A., 2000. OOM, la conception objet des systems. d'information, Hermes.

Chandrasekaran, B., Glasgow, J., Narayanan, N.H., 1995. Introduction. In: Glasgow, J., Narayanan, N.H., Chandrasekaran, B. (Eds.), Diagrammatic Reasoning: Cognitive and Computational Perspectives. AAAI/MIT Press, Massachusetts, pp. 15–27.

Christensen, H.L., Haslund, M.L., Nielsen, H.N., Tryfona, M., 2001. Operations for conceptual schema manipulation definitions and semantics. In: Advances in Databases and Information Systems (ADBIS 2001), 5th East European Conference, Vilnius, Lithuania, September 2001. LNCS 2151.

De Lucia, A., Gravino, C., Oliveto, R., Tortora, G., 2009. An experimental comparison of ER and UML class diagrams for data modeling. Journal of Empirical Software Engineering 14 (1), 57–92.

Donmoyer, R., 1990. Generalizability and the single-case study. In: Eisner, E.W., Peshkin, A. (Eds.), Qualitative Inquiry in Education: The Continuing Debate. Teachers College Press, New York, pp. 175–200.

Entity Relationship Diagram Tool, 2010. Retrieved on 4th November 2009 from http://highered.mcgraw-hill.com/sites/0072942207/student_view0/e_r_assistant.html

Ericsson, K.A., Simon, H.A., 1993. Protocol Analysis: Verbal Reports as Data, revised edition. Bradford Books/MIT Press, Cambridge, MA.

Fichman, R.G., Kemerer, C.F., 1992. Object-oriented and conventional analysis and design methodologies. IEEE Computer, 22–39.

Gabay, J., 2001. Merise et UML Pour la Modelisation des Systemes d'information. Dunod, Paris.

Genero, M., Poels, G., Piattini, M., 2008. Defining and validating measures for assessing the understandability of entity-relationship diagrams. Data Knowledge Engineering, 534–557.

Glaser, B.G., Strauss, A.L., 1967. The Discovery of Grounded Theory: Strategies for Qualitative Research. Aldine Publishing Company, Chicago.

Glasgow, J., Narayanan, N.H., Chandrasekaran, B., 1995. Diagrammatic Reasoning: Cognitive and Computational Perspective. MIT Press, Cambridge, MA.

Henderson, J.M., 2007. Regarding scenes. Current Directions in Psychological Science 16, 219–222.

Hungerford, B.C., Hevner, A.R., Collins, R.W., 2004. Reviewing software diagrams: a cognitive study. IEEE Transactions on Software Engineering 30 (2), 82–96.

Just, M.A., Carpenter, P.A., 1976. Eye fixations and cognitive processes. Cognitive Psychology 8, 441–480.

Klemola, T., Rilling, J., 2002. Modeling comprehension processes in software development. In: Proceedings of the First IEEE International Conference on Cognitive Informatics, pp. 329–337.

Knoblich, G., Ohlsson, S., Raney, G.E., 2001. An eye movement study of insight problem solving. Memory and Cognition 29, 1534–1555.

Kumaresh, S., Baskaran, R., 2000. Defect analysis and prevention for software process quality improvement. International Journal of Computer Applications 8 (7), 42L 47.

Laitenberger, O., Atkinson, C., 1999. Generalizing perspective-based inspection to handle object-oriented development artifacts. In: Proceedings of the 21st International Conference on Software Engineering, Los Angeles, CA, pp. 494–503.

Laitenberger, O., Atkinson, C., Schlich, M., El Emam, K., 2000. An experimental comparison of reading techniques for defect detection in UML design documents. Journal of Systems and Software 53 (2), 183–204.

Larkin, J.H., Simon, H.A., 1987. Why a diagram is (sometimes) worth ten thousand words. Cognitive Science 11 (1), 65–99.

Liao, C., Pavlia, P.C., 2000. The impact of data models and task complexity on end-user performance. International Journal of Human–Computer Studies, 831–845.

Loftus, G.R., Mackworth, N.H., 1978. Cognitive determinants of fixation location during picture viewing. Journal of Experimental Psychology: Human Perception and Performance 4, 565–572.

Merriam, S.B., 1998. Qualitative Research and Case Study Applications in Education. Jossey-Bass Publishers, San Fransisco.

Nielsen, J., Landauer, T.K., 1993. A mathematical model of the finding of usability problems. In: Proceedings of the ACM INTERCHI'93 Conference, Amsterdam, The Netherlands, pp. 206–213.

Porter, A.A., Siy, H., Mockus, A., Votta, L.G., 1998. Understanding the sources of variation in software inspections. ACM Transactions on Software Engineering and Methodology 7, 41–79.

Rayner, K., 1998. Eye movements in reading and information processing: 20 years of research. Psychological Bulletin 124, 372–422.

Rochfeld, A., Bouzeghoub, M., 1993. From MERISE to OOM. Ingen. Syst. d'inform. 1 2, 151L 176.

Rombach, C.D., Kude, O., Aurum, A., Jeffery, R., Wohlin, C., 2003. An empirical study of an ER-model inspection meeting. In: Proceedings of the IEEE 29th EUROMICRO Conference New Waves in System Architecture (EUROMICRO'03).

Rumbaugh, J., Jacobson, I., Booch, G., 1999. The Unified Modeling Language Reference Manual. Addison-Wesley-Longman, Reading, MA, pp. 1–550.

Runeson, P., Wohlin, C., 1998. An experimental evaluation of an experience-based capture-recapture method in software code inspections. Journal of Empirical Software Engineering 3 (4).

Shoval, P., Shiran, S., 1997. Entity-relationship and object-oriented data modeling – an experimental comparison of design quality. Data and Knowledge Engineering 21 (3), 297L 315.

Shoval, P., Frumermann, I., 1994. OO and EER conceptual schemas: a comparison of user comprehension. Journal of Database Management 5 (4), 28–38.

Song, I.Y., Froehlich, K., 1995. Entity-relationship modeling. In: Potentials, IEEE 13, 5:29–34.

Stenning, K., Lemon, O., 2001. Aligning logical and psychological perspectives on diagrammatic reasoning. Artificial Intelligence Review 15 (1/2), 29–62.

Travassos, G., Shull, F., Fredericks, M., Basili, V.B., 1999. Detecting defects in object-oriented designs: using reading techniques to increase software quality. ACM SIGPLAN Notices archive 34 (10), 47–56.

Underwood, G., Jebbett, L., Roberts, K., 2004. Inspecting pictures for information to verify a sentence: eye movements in general encoding and in focused search. Quarterly Journal of Experimental Psychology 57A, 165–182.

Wohlin, C., Runeson, P., 1998. Defect content estimations from review data. In: IEEE International Conference on Software Engineering, April 19–25, 1998, pp. 400–409.

Wohlin, C., Aurum, A., 2003a. Evaluating cost-effectiveness of checklist-based reading of entity-relationship diagrams. In: Proceedings of the International Symposium on Software Measures, Sydney, Australia, pp. 286–296.

Wohlin, C., Aurum, A., 2003b. An evaluation of checklist-based reading for entity-relationship diagrams. In: IEEE Proceedings of the Ninth International Software Metrics Symposium (METRICS'03).

Zhang, J., 1997. The nature of external representation in problem solving. Cognitive Science 21 (2), 179–217.

Assist. Prof. Dr. Ercil Cagiltay received the degree in computer engineering from the Middle East Technical University, Turkey. She received the PhD degree in instructional technologies from the same university. She worked for commercial and government organizations as a project manager for more than eight years in Turkey. She also worked for the Indiana University Digital Library Program as a system analysis and programmer for four years. She has been with the informatics group of the Atilim University, Turkey, since 2003 as an assistant professor. Her main research interests are in information systems, higher education, and instructional systems technology.

Dr. Gul Tokdemir graduated from Middle East Technical University in 1995. She has completed her master's degree in Georgia Institute of Technology (1997) and her PhD in Middle East Technical University (METU), Informatics Institute, in 2009. She taught at Atilim University in computer engineering department until 2009. She worked as a post-doc researcher in Eindhoven Technical University. Currently, she is working at Cankaya University, Computer Engineering Department. Her research interests include Information Systems, Software Engineering issues related to databases, and design diagrams.

Ozkan Kilic received his BSc in 2005 from computer engineering, METU in Ankara. He had his MSc degree in 2007 from the department of cognitive science, METU. He is currently a PhD student at the Department of Cognitive Science, METU. He was a research assistant at the department of computer engineering, Atilim University in Ankara from 2004 to 2007. He was employed as an instructor at the same department from 2007 to 2008. He currently works as a treasury expert in the Undersecretariat of Treasury, Ankara. His research areas are computational morphology, diagrammatic reasoning and NLP.

Damla Topalli is a teaching assistant in computer engineering department of Atilim University. She is currently a MSc student in computer engineering department at Atilim University. She has a bachelor's degree from department of computer engineering at Baskent University. Her research interests include image processing, database management systems and she is working on a shared system about personal health records of patients.