A TOOL FRAMEWORK FOR DEVELOPING CONTEXT-SENSITIVE USER ASSISTANCE SYSTEMS USING MODEL-DRIVEN ASPECT WEAVING

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Murat Açar

August, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Bedir Tekinerdoğan(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ali Yazıcı

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Özgür Ulusoy

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

A TOOL FRAMEWORK FOR DEVELOPING CONTEXT-SENSITIVE USER ASSISTANCE SYSTEMS USING MODEL-DRIVEN ASPECT WEAVING

Murat Açar
M.S. in Computer Engineering
Supervisor: Asst. Prof. Dr. Bedir Tekinerdoğan
August, 2012

User assistance systems act as a guide for the users of software products. These systems aim to guarantee a successful user experience by helping in performing tasks. Early on, off-line user manuals were mostly the mediums of user assistance, and technically, they were independent of the systems they belong to. The upward trend in user assistance systems is that the provision of assistance is automated through some attached mechanisms to the software systems. There have been numerous proposals introducing fresh and novel methods for the purpose of automated user assistance. Specifically, embedded user assistance consists of instructional or conceptual information that appears within a software application window. It includes embedded help that appear within the application, field labels, and page overviews.

The overall objective of this thesis is to reveal the state of the art advances in user assistance systems, and to propose a tool framework for developing context-sensitive user assistance systems. Firstly, we conducted two systematic literature reviews for both automated and embedded user assistance systems. The systematic literature reviews are required for acquiring solid background on embedded user assistance systems as well as for exploring the main obstacles to automated user assistance systems. The research findings are presented in parallel with the work published in the literature, and we aim at revealing a variety of techniques used for automated and embedded user assistance. The systematic reviews are conducted by a multiphase study selection process under a lot of articles obtained by dedicated search strategies. Since there has been no study to systematically undertake the state of user assistance systems, our work has a pioneering value of contents providing a road-map of current trends for further researchers in the field of user assistance.

Having analyzed the results of systematic reviews, we conducted a survey of help authoring tools that revealed the lack of generalized context-sensitive user assistance solutions. Also, the utilization of methods, algorithms and tools differs from domain to domain, being rather scattered. We aimed at developing embedded context-sensitive user assistance systems, which is not trivial and has to meet several challenges. Unfortunately, user-assistance concerns such as help content and related weaving information cannot be easily localized in single modules and as such tend to crosscut multiple modules. The reuse of user assistance tools for different applications is required because developing custom-based user assistance for each separate application is laborious. Consequently, the obstacles related to the development of context-sensitive user assistance systems have brought out the idea of a tool framework for this purpose. To address these issues we developed an aspect-oriented tool framework *Assistant-Pro* that can be used to develop context-sensitive embedded user assistance for multiple applications. The framework provides tools for defining the process model, defining guidance related to process steps, and modularizing and weaving help concerns in the target application for which user guidance needs to be provided. The tool has been originally developed and validated in the context of Aselsan, a large Turkish defense electronics company.

Keywords: Aspect-Oriented Software Development, Context-Sensitive User Assistance, Systematic Literature Reviews.

ÖZET

MODEL-GÜDÜMLÜ İLGİ DOKUMA KULLANARAK İÇERİK-DUYARLI KULLANICI YARDIMI SİSTEMLERİ GELİŞTİRMEK İÇİN BİR YAZILIM ÇERÇEVESİ

Murat Açar
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Asst. Prof. Dr. Bedir Tekinerdoğan
Ağustos, 2012

Kullanıcı yardımı sistemleri, yazılım ürünü kullanıcılarına kılavuzluk ederler. Kullanıcılara görevleri boyunca yardım ederek, başarılı bir kullanıcı deneyimini garantilemeye çalışırlar. Eskiden, en çok kullanılan kullanıcı yardımı araçları, çevrimdışı kullanma talimatı dökümanlarıydı. Bu dökümanlar bağlı oldukları sistemden tamamen bağımsız basılı şekilde bulunuyordu. Bu konuda son zamanlarda yükseliş gösteren eğilim ise, yardımın sağlanmasını otomatize eden teknolojiler kullanmaktır. Gömülü kullanıcı yardımı, eğitici ve kavramsal bilgiler içeren ve kullanıcı arayüzlerinde görülen bir yardım tipidir. Bu tip kullanıcı yardımında, yardım başlıkları, alan etiketleri ve sayfa açıklamaları en çok görülen çözüm yöntemleridir.

Bu çalışmanın genel amacı, kullanıcı yardımı alanındaki en gelişkin teknolojileri ortaya çıkararak, bunların sonuçlarına bağlı bir yazılım çerçevesi geliştirmektir. Bu doğrultuda, otomatize kullanıcı yardımı ve gömülü kullanıcı yardımı alanlarının ikisi için de ayrı bir sistematik literatür incelemesi yapılmıştır. Araştırma bulguları, literatürde yayınlanmış detaylı çalışmalara paralel bir şekilde sunularak, bu konuda çok çeşitli çözüm yöntemlerinin ortaya çıkarılması amaçlanmıştır. Sistematik literatür incelemeleri, belirli araştırma stratejilerine dayalı yüzlerce farklı çalışmayı, çok aşamalı bir yayın seçme sürecine tabi tutmaktadır. Şu ana kadar kullanıcı yardımı sistemlerinin durumunu sistematik bir şekilde ele alan bir çalışma yapılmadığı için, bu çalışma güncel akımları göz önüne sererek, öncü değerde bir içerik sunmaktadır.

Sistematik literatür incelemelerinin ve yardım yaratma araçları üzerinde yaptığımız bir diğer araştırmanın sonuçlarını analiz ettiğimizde, genellenmiş

içerik-duyarlı kullanıcı yardımı sunan çözümlerin eksikliği ortaya çıkmıştır. Ayrıca yöntem, algoritma ve araçların kullanımı oldukça dağınıktır. Biz bu çalışmada, aslında hiç de kolay olmayan ve muhtelif zorluklar içeren, gömülü içerik-duyarlı kullanıcı yardımı sistemleri geliştirmeyi amaçladık. Ne yazık ki kullanıcı yardımı işleri, tek modüller içinde kolaylıkla lokalize edilemezler ve bu şekilde birden fazla modülü enine kesmeye eğimlidirler. Her münferit uygulamaya özel kullanıcı yardımı geliştirmek zahmetli olduğu için, kullanıcı yardımı araçlarının farklı uygulamalarda yeniden kullanılabilir olması gerekmektedir. Sonuç olarak, içerik-duyarlı kullanıcı yardımı geliştirmenin önündeki engeller, bu amaca yönelik bir araçlar çerçevesi fikrini beraberinde getirmiştir. Biz bu konulara çözüm yaratmak amacıyla, birden çok uygulamada içerik-duyarlı gömülü kullanıcı yardımı sunmak için kullanılabilen ve ilgiye-yönelik bir araçlar çerçevesi olan *Assistant-Pro* 'yu geliştirdik. Bu çerçeve, süreç modeli tanımlamak, süreç adımlarına ilişkin yardım içeriğini tanımlamak ve yardım gerektiren hedef uygulamada yardım içeriğini modularize etmeye ve dokumaya imkan veren araçlar sunmaktadır. Bu araçlar çerçevesi orijinal olarak, büyük bir savunma sanayii firması olan Aselsan'ın kapsamında geliştirilmiş ve doğrulanmıştır.

Anahtar sözcükler: İlgiye-Yönelik Yazılım Geliştirme, İçerik-Duyarlı Kullanıcı Yardımı, Sistematik Literatür İncelemeleri.

Acknowledgement

I would like to express the deepest appreciation to my supervisor, Asst. Prof. Dr. Bedir Tekinerdoğan, for the continuous support of my M.S. study and research, for his patience, motivation, enthusiasm, and immense knowledge. This study would not have been possible without his guidance and persistent help.

I wish to express my warm and sincere thanks to Prof. Dr. Ali Yazıcı for his detailed and constructive comments, and for his important support throughout research career.

I warmly thank Prof. Dr. Özgür Ulusoy for taking place in my thesis committee, and for his review, criticism and advices during the presentation of this thesis.

I am grateful to all my EA-407 office friends who have lent their hands to complete this thesis, specially to Burcu Dal for the words of encouragement and for helping me out with my studies. I thank my friends for the sleepless nights we were working together, and for all the fun we have had in the last two years.

The financial support of TÜBİTAK is gratefully acknowledged.

Last but not the least, I would like to thank my family: my mother Bilge Açar, my sisters, my newborn nephews and nieces, supporting me spiritually throughout my life. To them I dedicate this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Current approaches in user assistance have been evolving in response to both technological advances and ever-changing requirements. Since the development of software-intensive systems should be cost-effective in all manners, the provision of user assistance is to be accomplished with minimal efforts. The fundamental issue is to develop user-friendly help mechanisms to ensure better user experience. Off-line printed user manuals that are of long standing have somehow fulfilled their duties, and they have been started to put away. The problem with these manuals is the effort to read many pages and to find needed information to accomplish tasks. Besides, off-line user assistance solutions intervene the users' work-flow, which is undesirable in practice. In these solutions, users stop their current work, consult the documentation in order to find the information they are looking for and then return to the application. This separate effort discourages users about off-line user assistance, making them reluctant to using help [1].

1.1 Background

The state of user assistance has migrated to online help systems due to the mentioned problems. The corresponding assistance in online help systems is presented to the users in more effective formats such as hypertext and PDF.

Online help is mostly topic-oriented, procedural or reference information provided by means of software systems. It is a form of user assistance. Online help can also be used to present information on a broad range of subjects. There exist several categories of online help systems that have been proposed so far. For example, Apple Computer introduced *Balloon help* as a help system in their 1991 release of System 7.0. The associated help text was presented to the users in *balloons*, comprising of the words in a comic strip. After the inception of this solution, it has been adopted by several pop-up help text mechanisms. Apple drew out the problem of providing user assistance in depth. Their initiatory step was to identify a number of common questions that occur to the user, such as *where am I?* and *how do I get to...?*. Hereafter, they specifically identified two main types of questions that users asked in computer usage: *what is this thing?* and *how do I accomplish...?*.

One of the main problems is that existing help systems typically did not resolve the issues identified. Individuals mostly, and so to say simply, copy paper manuals into electronic formats. The *what is this thing?* question arises in many cases where the users have difficulties in using software systems. When the visual design of a graphical user interface is complicated due to non-standard widgets or buttons labeled with an indecipherable icon, users have to consult the related documentation. However, users are reluctant to stop what they are doing just because of the obscurity of some structures. For the issues set out in these observations, Apple introduced Balloon Help as the solution. Later on, Apple focused on *how do I accomplish...?* question which was quenched by Apple Guide [2].

WebHelp is a specific category of online help that can either be delivered via the Internet or as a stand-alone set of HTML files on a computer. This approach, which combines the Internet and local resources, is also used in Windows XP's Help and Support feature. *WebHelp* enables browser-independent, platform-independent online Help and electronic books using a combination of HTML, DHTML, JavaScript, Java, and ActiveX [3]. In order to make use of WebHelp, technical communicators consult help authoring tools. When online help is linked to the state of the application, in other words the user's work-flow,

it is called *Context-Sensitive User Assistance*.

Embedded User Assistance is a novel, cutting-edge approach in order to provide online help directly into the user interface. *Embedded User Assistance* keeps users in their task flow. However, there have been few illustrations of *Embedded User Assistance* for technical communicators to analyze. *Embedded User Assistance* appears within the application rather than in a separate window. To create it, we need to maintain a close working relationship with application developers. One of the solutions for *Embedded User Assistance* was the help system in Microsoft Money 99. When the user selects *Help Topics* from the Help menu, the Money 99 Help is attached to the right side of the application. The Microsoft Money 99 Help offers tutorials, explanations, and demonstrations to help the user use and learn Microsoft Money. *Embedded User Assistance* can also be created for web-based applications. It differs from other types of online help in the following senses [4]:

- It requires very short and focused topics
- It is not based on traditional organizational tools such as a table of contents or index.

Although *Embedded User Assistance* is quite effective and useful, it is not an all-round substitute for other types of guidance. There are several design considerations for *Embedded User Assistance* that incorporates any information within the user interface that guides the user:

- Field labels
- Inline instructional text
- Error or information messages
- Button labels
- On-screen examples

- Hover text
- Tool tips

Context-Sensitive User Assistance focuses on the state of the target application to which help is to be integrated. We define the states of the target application along with some related topics. In the literature, there are several terminologies to categorize *Context-Sensitive User Assistance*, and *Process-Sensitive User Assistance* is one of these categories in which we deal with processes, scenarios, and steps to incorporate embedded user assistance.

1.2 Problem statement

While developing *Embedded User Assistance* for a software-intensive system, there are some general-purpose concerns to be taken into account as follows:

Usability The presented user assistance mechanisms shall be user-friendly.

Reusability User assistance structures shall be reusable across multiple applications.

Modularity The target application requiring user assistance shall be modularly extended without any detriment to high-cohesion and loose-coupling.

Maintainability The structures related to user assistance shall be easily configured and enhanced after being in use.

Cost-effectiveness The costs related to developing user assistance and the effort in terms of time and budget shall be minimal.

Optimization The user experience with the target application shall be optimized in all manners through helping in accomplishing tasks.

Pro-activeness User assistance solutions shall be on the same level with user tasks without intervening their flows.

The crosscutting property of help and control flow concerns reduce the modularity of the system and as such impede maintenance. Furthermore, the maintenance activities on crosscutting concerns are so challenging that they may be required for multiple applications at the same time. Also, the time of development of an application is needlessly increased just because of integrating crosscutting help concerns that interfere the actual functionality. Considering the above concerns, there are a few solutions to provide *Embedded User Assistance* in practice that will be analyzed in detail. Besides, the results of our systematic literature reviews imply that generalized *Context-Sensitive User Assistance* solutions are greatly needed. The discovered methods, algorithms and tools for *Embedded User Assistance* seem to be the consequences of scattered thoughts. These issues have also been discovered in the example case applications of Aselsan [5] that initiated our work.

1.3 Contribution

In the context of this thesis, we performed two systematic literature reviews(SLRs) for presenting the current trends in user assistance systems. To the best of our knowledge, there has been no published literature on this topic that thoroughly reviews the best practices. Also a general-purpose application framework containing several tools to integrate context-sensitive help content in a modular way is proposed. The contribution of this thesis can be summarized as follows:

SLR to Automated User Assistance Systems We reviewed articles on *Automated User Assistance* of the best quality in respect of our quality assessment criteria. The systematic review is presented in this thesis with all details of stages to perform. A total of 575 papers were analyzed in this part, and this provides a general overview of user assistance systems.

SLR to Embedded User Assistance Systems In a similar fashion, we analyzed the *embedded* category of *Automated User Assistance* systems to

approach our case in hand which is based on *Context-Sensitive User Assistance*. In this systematic review, we reviewed 550 papers resultant of a well-defined search strategy, and presented the results of high quality papers that were selected as primary studies.

User Assistance Model We have specified the potential components of a user assistance model that would enable us to develop generalized user assistance systems. Upon the proposed tool framework, we report potential empty rooms for further studies in this field on the sound basis of our systematic literature reviews.

Metamodeling for User Assistance Systems We have employed *Aspect-Oriented Software Development* based on *Annotations* and *Models* such as user models, process models and interest models. Also, the combination of paradigms used in this thesis is aligned with *Model-Driven Software Development*, being a possible reminiscent of it. Therefore, we state *metamodeling* approaches for developing *Embedded User Assistance* by providing a road-map.

Tool Framework We had a tool framework dedicated to the industrial settings of Aselsan [5], and the framework has gone through an enrichment process in response to the results of systematic literature reviews. We provide some unsatisfied points in providing *Embedded User Assistance* along with a set of, so to say, prescribed solution approaches.

1.4 Outline of Thesis

The thesis is organized as follows:

- Chapter 2 provides a systematic literature review of *Automated User Assistance* systems.
- Chapter 3 presents the results of our systematic literature review on *Embedded User Assistance* systems.

- Chapter 4 presents the survey of *Help Authoring Tools*.
- Chapter 5 presents the proposed tool framework on the basis of systematic literature reviews.
- Chapter 6 provides the related work in the field of user assistance.
- Chapter 7 concludes the thesis with the inferences from the work done and future work.

Chapter 2

Systematic Review of Automated User Assistance Systems

This chapter is based on our systematic literature review of *Automated User Assistance* systems. Section 2.1 presents an overview of *Automated User Assistance* systems. Section 2.2 provides a background of the study. Section 2.3 describes the research method used in this study. Section 2.4 shows the results of this systematic review. Section 2.5 includes a discussion part. Finally, the chapter ends with Section 2.6 stating concluding remarks.

2.1 Overview

User assistance is of great importance in interactive software applications, and users have to be provided excellent guidance on how to accomplish tasks to lead successful experience. However, the provision of user assistance itself is a challenging concern that requires a great deal of effort. Since this is the way it is, software professionals are in need of dedicated user assistance tools for meeting guidance concept with better user experience [1].

Computer users focus their attention on the completion of tasks in hand, and

they are generally reluctant to using guides or any help instruments provided. Therefore, user assistance specialists endeavor to present effective and to the point help mechanisms. The main aim is to attract the attention of users in a pragmatic manner that they think of user assistance as a contributor rather than a disrupter. Hereby, users are supposed to follow a regular flow of work that leads them up to successful completion of missions [6]. The forms of user assistance have been evolving day-by-day as the software technologies make progress in terms of the methodologies and paradigms used. Normally, a user assistance system of good quality is not sufficient to provide an effective software system with user interaction. The users of the computing system are to obtain needed functionality, and the way to access the functionality should be easy enough to accomplish a task seamlessly. Thus, holding the user assistance on the level with employed systems is a principal task since help concern itself has a system-wide behavior. Having a ripple effect on the software decomposition, this concern has to be extracted from the principal functionalities of systems.

There has been a proliferation of techniques in this research area by the fact that it covers multiple domains. This systematic review serves as a roadmap to user assistance researchers by identifying the body of multidisciplinary research on the field. Hypothetically speaking, software developers generally see user assistance as a minor concern, and they attach help agents in a way that ruin the core functionality. Naturally, the attached structures cut across the primary decomposition of the software systems. Hence, this concern is to be modularized by means of independent help authoring tools.

Additionally, the existing solutions in the literature have their own properties, advantages and shortfalls that are observed comprehensively. This review also assesses these solutions in a general way that some major details of them are evaluated. Thus, the interested parties may specify one of the solutions or groups of solutions in order to fulfill their own cases on hand. As was previously mentioned, this research domain is an increasingly growing area, and it is of primary importance that individuals have background knowledge about the characteristics of proposed techniques before applying them. We are currently entering an era of automated user assistance systems, and this systematic review extracts the

tendencies published in the literature towards the development of these systems.

2.2 Background

2.2.1 Automated User Assistance Systems

Users are usually abandoned to themselves along with their frustrations when using a piece of software, whereas they would need intelligent and co-operative, in other words automated, help. Ideally, human computer interaction concepts have to remedy the unaccomplished cases where the fulfillment of users needs is somewhat perfunctory.

The technological progress and a great body of work performed by researchers and practitioners in the field of user assistance have not put an end to this problematic concern. Possibly, the turmoil in this domain has gone on in a complicated way from past to present due to the lack of enlightening studies that reveal the technical innovations. Thus, the individuals are in need of an awareness research in a sense.

Basically, considering the design of an essence user assistance system, we can speak to the matter of two criteria that are to be covered [7].

- the facilitation of the accomplishment of a particular task by a user having no idea about the completion
- the provision of effective mechanisms for the users to learn the use of the system with a good progressive rate of performance.

The term *automated* is considerably broad, and mainly, it refers to the existence of user assistance systems that are working together with the appertaining system. Technical communicators, engineers and visual designers have been using the forms of user assistance that are somewhat automated. The progressive

tendencies towards user assistance are some new techniques that have recently arised in order to shape the assistance mechanishms in an automated way [6].

2.2.2 Systematic Reviews

The inception of systematic reviews is based on the evidence-based concept which is devised in the field of medicine. Inspired from this point, the world of software engineering has been expanding the volume of research undertaking systematic reviews. The popularity gained by this area led the evidence-based software engineering (EBSE) [8,9]. The major thought here is that it is worthwhile considering why evidence would be beneficial to the field of software engineering.

EBSE is of vital importance because software intensive systems are starting to form the basis of many applications and to take a crucial place in daily life. Reference [7] states the means that EBSE would provide:

- A common goal for individual researchers and research groups to ensure that their research is directed to the requirements of industry and other stakeholder groups.
- A means by which industry practitioners can make rational decisions about technology adoption.
- A means to improve the dependability of software-intensive systems, as a result of better choice of development technologies.
- A means to increase the acceptability of software-intensive systems that interface with individual citizens.
- An input to certification processes.

The improvement in decision making processes by integrating current best evidence from research is the initiatory point of systematic reviews. Since Kitchenham et al. published the seminal paper of EBSE [7], systematic reviews have been increasingly growing. The references [10–15] are the reviews that was published

in the field of software engineering. These reviews look for the best evidence in specific areas of software engineering. Also, reference [16] is taking systematic literature reviews as primary studies, and conducts a review of them, in other words a tertiary study.

2.2.3 Objectives of the Review

Having discussed the growth of systematic reviews in the field of software engineering, we can discuss the objectives of the current review. Initially, automated user assistance systems have attracted great interest from the software industry. As was previously stated, there have been several approaches towards this research area. However, both the researchers and the practitioners need a roadmap to rely on prior to applying techniques and methodologies. Mostly, practitioner books are the sources in order to get an overview of automated user assistance.

There has been no systematic review of automated user assistance systems published. The previous studies like [17–19] are the ones that stand as, so to say, a survey of current trends in this area, but their scope and objectives are too narrow to be taken as a roadmap. Besides, their aim is not to undertake the whole field.

We hope that this study will be beneficial for both researchers and practitioners by enlightening the current proposals on automated user assistance systems. Also, as we consider the claims that are supported by scientific studies, the assessment and findings will be useful. Additionally, the review takes both qualitative and quantitative studies into account so that the individuals can draw benefits from diverse studies, even by integrating and creating ensemble of them.

2.3 Research Method

We conducted the review to reveal existing evidence concerning the automated user assistance systems. Kitchenham and Charters [20] published a guideline for

performing SLRs, and we entirely followed the guidelines, procedures and policies stated in this paper. This section discusses our research method that is based on an extensive review protocol.

2.3.1 Review Protocol

Before conducting the review, we thoroughly researched the methods used in performing SLRs. In accordance with the guideline, there are several techniques to compose a review protocol that is one of the most important parts of a systematic review. In our protocol, we described the strategies for performing the review.

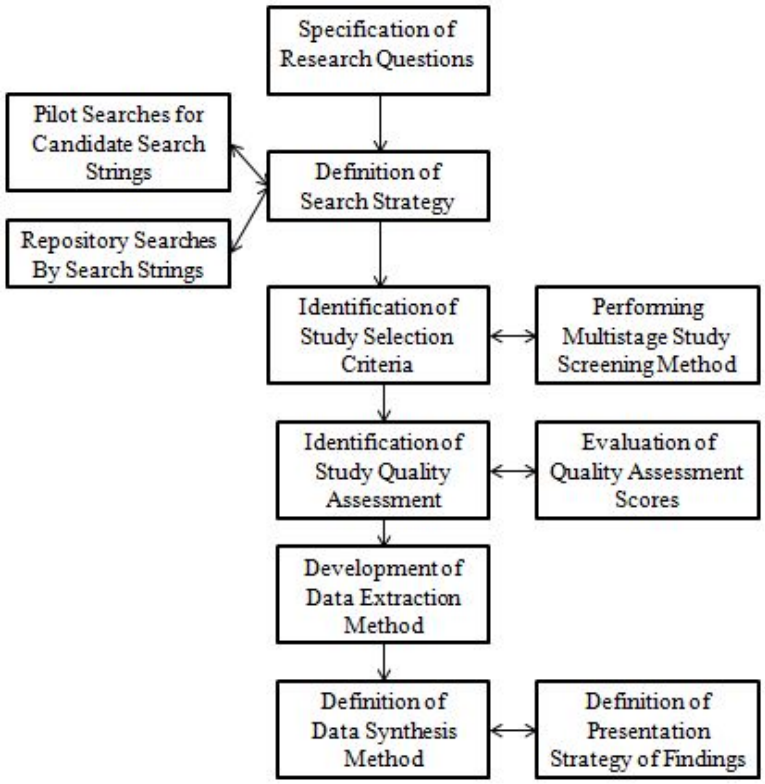

Figure 2.1: Activities under the review protocol

The whole picture is substantially in the form of Figure 2.1 where the consecutive steps take place. Firstly, we specified our research questions based on the point of origin of this systematic review. Hereafter, we defined an in-depth

search strategy that was formed after performing deductive pilot searches to reveal possible search strings. A good search string brings well search results that will come to a successful conclusion in terms of sensitivity and precision rates. At this step, we inspired from a novel approach proposed by Tell and Babar [21] in which they evaluated a published SLR by analyzing the employed search strings. Once the search strategy was defined, we specified the statements of our study selection criteria. These criteria were defined explicitly in order not to have a research bias.

We screened the primary studies at all phases on the basis of inclusion and exclusion criteria. Also, peer reviews were performed by the authors throughout the study selection process. Afterwards, we assessed the quality of selected primary studies based on a scoring instrument. Having the scores of individual studies, we developed a somewhat data extraction form that specifies the sections to be extracted from the selected papers. The decision on the selection of specific parts of studies is analogous to the specification of research questions where we targeted our points of interest in this review. Last but not least, the data synthesis process takes place in which we present the extracted data and associated results.

2.3.2 Research Questions

The most important part of any systematic review is to clearly and explicitly specify the research questions. This phase affects the subsequent parts of the systematic review accordingly [20]. This is owing to the fact that the more precise the research questions are, the more accurate the findings are. As was previously stated, there has been no systematic review on automated user assistance systems. Hence, we aimed at the current state of art by means of our research questions with a wide sphere of influence in order to get the best evidence. In order to form a sound basis for both researchers and practitioners, we need to set our sights on the state-of-the-art by examining the evidence related to automated user assistance. Thus, the first research question is formulated as follows:

- **RQ 1.In which domains of computer science have automated user assistance techniques been applied?**

  As we stated before, the term *automated* can be achieved by employing diverse models, methods and algorithms. In some cases, the integration of several approaches can be proposed to ensure the provision of automated user assistance. Thus, we aimed at, so to say, depicting the research space of these studies. To shape a stepping stone for future research, our second research question in this SLR is:

- **RQ 2. What are the existing research directions within automated user assistance?**
  - **RQ 2.1. What are the different automated user assistance solutions used?**
  - **RQ 2.2. What are the implications of automated user assistance solutions for future search and practical use?**

  The plausibility of the reported results and findings in an SLR is to be attached great importance so that the readers can be well aware of the conclusions drawn. Therefore, the overall strength of the body of evidence presented is to be brought out for the readers. Our third research question is defined for this purpose as follows:

- **RQ 3. What is the strength of evidence in support of the stated findings?**

The *population* in this review is the domain of automated user assistance. *Intervention* includes context-sensitive, process-sensitive, embedded, intelligent and adaptive user assistance techniques that are observed after performing a preliminary analysis discussed in our search strategy. The *comparison* criterion is not taken into consideration in this review since our study is not intended to compare any practice with the intervention.

Outcomes are not limited in this review as automated user assistance can improve a plenty of factors of importance to practitioners. We do not impose

any restrictions on the *context* and *experimental design*. The reason is that the paucity of primary studies would be a problem for the review. Thus, we prefer the aggregation of diverse study types that would hopefully be representative for the community of research and practice.

2.3.3 Search Strategy

The primary aim of any systematic review is that an unbiased search strategy has to be employed to explore as many primary studies as possible. As we decomposed our research questions into some distinct facets (i.e. *population* and *intervention*), the designation of search string was accomplished according to the words determined in these facets. Also, a list of synonyms, abbreviations, and alternative spellings was composed as an auxiliary instrument. Hereafter, a multifaceted search string was obtained by means of Boolean ANDs and ORs.

It is an evident fact that generating a search strategy also requires a comprehensive search on the reference lists from observed primary studies, journals (especially some prestigious company journals) and conference proceedings, grey literature, research registers, and the Internet [20]. A preliminary analysis is usually feasible before conducting an SLR, because the database searches inevitably depend on the authors background knowledge. Also, these searches are not in favor of scientific rigor.

Tell and Babar [21] stated some questions, inspired from [17], about the rigor and performance of a search strategy as follows:

- How to design a rigorous search strategy that maximizes the collection of relevant studies?
- Are there any balancing criteria that regard the trade-off between recall and precision in a search strategy?
- Can we evaluate a predefined search strategy with the underlying search strings?

Zang and Babar [22] suggest a prior survey of publications that will constitute the *quasi-gold standard*. This investigation helps us in designating search strings by revealing better keywords. The main point here is that the identification of relevant digital libraries becomes a necessity, and by this way, the need for a validated search strategy can be met.

The reason for attaching the term *quasi* is that the acquisition of an absolute gold standard is impossible for the majority of SLRs. In other words, running a perfect search that results in a set of primary studies in which every single one is desired whereas we do not miss any published one is rather doubtful. Due to this unfortunate circumstance, it would be the best of all that we can use some known primary studies based on our knowledge on the topic. Possibly, we may have a relatively comprehensive and rapidly growing collection of relevant studies by manually analyzing related sources. In this way, the elicitation of search string will be quite objective instead of some subjective decisions on the keywords.

We focused on the formation of an optimal search strategy that would retrieve as much relevant information as possible, while striving against low cost and effort. A searchs optimality depends on its recall and precision [23]. Recall and precision are the measures that the people in medicine make great use of. In order to evaluate a predefined search strategy, we can borrow and use these measures as the indicators of perfection.

It has been argued that depending on the objectives of an SLR, one of the criteria can be more favored and used by the investigators [23]. However, the one cannot pretend not to see the fact that a sensitive search strategy requires a great manual effort of dealing with irrelevant articles whereas a precise search strategy unavoidably misses a great many of relevant articles.

Figure 2.2 reveals the situation in explicit manner. That is why, the bizarre trade-off between recall and precision can possibly make researchers sacrifice either manual effort or pure relevant studies.

After having a solid background for generating a search strategy, we thought that constructing a *quasi-gold standard* (QGS) would help us in arriving at an

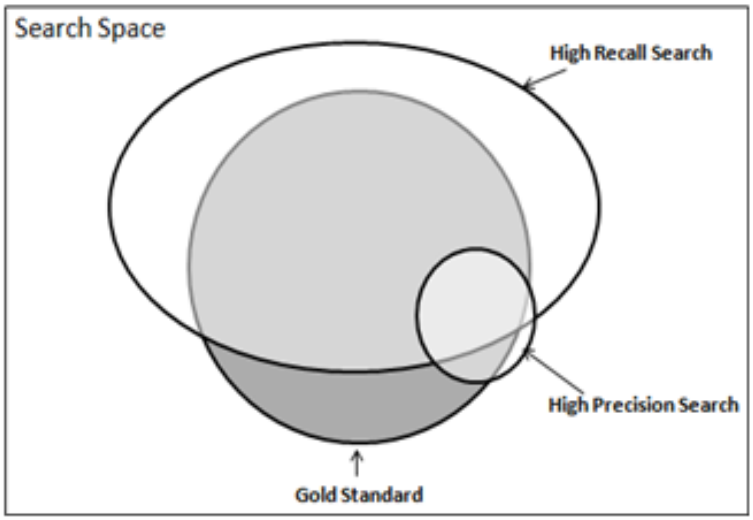

Figure 2.2: Activities under the review protocol

optimal search strategy. The primary studies, which we manually selected in reliance upon our knowledge of topic, were analyzed in order to elicit better keywords that would optimize the retrieval of relevant material. The analysis of the articles in the QGS was carried out by using word frequency and statistical analysis tools. WordStat [24], which is a content analysis & text mining software, and SimStat [25], which is a statistical analysis and bootstrapping software, were used for this purpose, introduced by Provalis Research [26].

First, the term frequency - inverse document frequency (TF*IDF) algorithm was operated on the titles and abstracts of the QGS papers. As stated by Tell and Babar [21], full text analysis would mislead us into thinking inaccurate keywords as true indicators because of the titles in the reference section. Also, the keywords of authors were manually examined to enhance the representative set of words observed. Finally, a definite set of search strings was obtained (see Figure A.1 in Appendix). Also, the execution of search strings in different sources should be performed seamlessly through analyzing the advanced search mechanisms provided by each venue.

We applied the search strings within the bounds of possibility that each venue facilitates. Although the structure of search strings seems to be different, there is no doubt that they are semantically equivalent. As was previously stated,

in addition to the database searches, we conducted manual searches both as a preliminary analysis and as a subsequent analysis after having observed the publication channels returned by the search strings. This manual searches were worth applying as we retrieved some good-quality articles that an automatic search could not reveal.

Source	**Number of Included Studies After Applying Search Query**	**Number of Included Studies After Exclusion Criterion 1**	**Number of Included Studies After Exclusion Criterion 2**
IEEE Xplore	271	23	9
ACM Digital Library	153	21	6
Wiley Interscience	18	3	1
Science Direct	41	12	4
Springer	34	16	3
ISI Web of Knowledge	30	6	4
Other Channels	28	12	3
Total	**575**	**93**	**30**

Table 2.1: Publication sources searched

2.3.4 Study Selection Criteria

Table 2.1 shows the distribution of the 30 primary studies according to the venues. At the very beginning, the search strings returned 575 papers from seven different sources. We later applied two exclusion criterions on this large-sized sample of papers. The overall exclusion criteria are as follows:

Exclusion criteria 1:

- Do not relate to a specific field of computer science
- Do not relate to user assistance
- Do not state any application of techniques, algorithms or methods to provide user assistance

- Do not report any results on the earnings of the approach proposed

Exclusion criteria 2:

- Abstracts or titles that do not mainly discuss the provision of user assistance were excluded
- Abstracts or titles that do not propose an approach to automate user assistance on the basis of the alternate terms that we have discussed were excluded

2.3.5 Study Quality Assessment

As was previously stated, we did not impose any restrictions on the context and experimental design; thus, any research methods or experimental designs were undertaken to assess the quality of primary studies. At this stage, the analysis involves both qualitative and quantitative studies, and the quality assessment was thought of being the initiator of data extraction and synthesis. The main objectives of this step can be listed as follows [20]:

- To enhance the study selection criteria
- To ascertain whether the study results are being affected by the quality bias
- To guide the interpretation of findings for data synthesis
- To reveal some open research questions and implications for future research

Study quality has no widely-accepted definition, but it is suggested that the extent to which a primary study reduces systematic errors (i.e. *bias*) and improves validity and applicability is believed to be an expression of quality. Therefore, a quality instrument is to be composed of questions used for assessing bias and validity of the selected primary studies.

We developed a quality instrument which is composed of checklists of factors that we planned to inquire for the primary studies. The derivation of the quality checklist was done by considering factors that could bias study results. As stated in [20], the types of bias are selection bias, performance bias, measurement bias and attrition bias. We refined the types of bias into our quality instrument by deliberating both generic and specific items with respect to different kinds of primary studies we selected. After having considered the bias and validity problems, we formed our quality instrument by first aiming at different stages of primary studies that fall into an empirical study. These stages are:

- Design
- Conduct
- Analysis
- Conclusions

Since our review includes qualitative studies, we merged the types of questions required to assess their quality with the ones that formed previously. While developing our quality instrument, we adopted the summary quality checklists that are proposed in [20] for both quantitative and qualitative studies. We substantially reviewed the list of questions in the context of our review and selected the ones that are aligned with our research questions.

The quality items in the instrument are deployed on a numerical scale because we did intend to rank and classify the studies with respect to an overall quality score. Therefore, we preferably employed a three point scale (i.e. yes = 1, somewhat = 0.5, no = 0) during the assessment.

The quality checklist is shown in Table 2.2. As was previously stated, we used the outcomes of quality assessment stage in order to assist data analysis and synthesis. We examined whether quality differences are correlated with the results reported in different kinds of primary studies.

No	Question
Q1	Are the aims of study clearly defined?
Q2	Are the scope, the context and the experimental design of the study clearly stated?
Q3	If the study involves assessment of a user assistance technology and draws some comparisons, is there a rationale for the evaluation?
Q4	Are the study participants or observational units adequately described?
Q5	Does the report have implications in practice and results in research area for automated user assistance?
Q6	Are the variables used in the evaluation likely to be valid and reliable?
Q7	Are the measures used in the study quite explicit and aligned with the research aims?
Q8	Is the research process documented adequately?
Q9	Are the main findings stated clearly in terms of creditability, validity and reliability?
Q10	Is there an explicit statement of the limitations?

Table 2.2: Quality Checklist

2.3.6 Data Extraction

In order to precisely extract and record the data retrieved from each of the 30 primary studies, we read the full-texts of them in a paired manner. The information needed to address our research questions and study quality criteria was collected by means of a data extraction form (see Figure A.2 in Appendix).

Actually, when the study review protocol became definite, the data extraction form was composed in order to reduce the tendency to bias. Since we considered the quality assessment stage as a part of the data analysis, the information collected for both the quality criteria and the review data was kept in the same form.

The data extraction form was piloted by both of the two researchers in consensus meetings so as to be consistent in subsequent analysis. After independent data extraction, data from both researchers were compared and disagreements were resolved by consensus. For this purpose, a sample of papers was chosen and read by the researchers, and the data extraction form was piloted on them. Thus, it is improved through a series of iterations and inter-researcher consistency was

assessed.

Basically, the data extraction form first included standard information such as name of the reviewer, date of data extraction, study ID, title, authors, journal, publication details, a brief summary and space for additional notes.

Secondly, the form covered the data directly related to answering the research questions. Some of the fields were: main theme of the study, motivation for the main theme, publication details, study aim, targeted domain, study settings, automated user assistance solution used, examples of application of solution, research method used, assessment approach, findings, constraints/limitations, implications for future research and major conclusions.

We recorded the places where the extracted information existed within the primary studies in spreadsheets. In order to enhance the process of synthesizing the extracted data, the form was developed in a progressive way so that the transition was performed seamlessly. Table 2.3 shows the data extracted for the research questions, respectively.

Research Questions	**Data Extracted**
RQ1	main theme of the study, motivation for the main theme, targeted domain, publication details
RQ2.1	study aims, automated user assistance solution used, research method used, examples of application of solution
RQ2.2	constraints/limitations, implications for future research and practical use, findings, major conclusions
RQ3	assessment approach

Table 2.3: Research Questions and Data Extracted

2.3.7 Data Synthesis

Data synthesis is the process of collating and summarizing the extracted data and the results of the selected primary studies [20]. At this stage, we performed a qualitative and quantitative analysis separately on the data extracted from the reviewed papers.

We investigated whether the qualitative results can lead us to explain quantitative results. For example, a primary study involving an assessment of an automated user assistance technology could help interpret other solutions in kind quantitatively. However, we also realized that reporting protocols differed too much in what we actually collected quantitative information. The reason behind is that the papers which are principally quantitative in nature are also heterogeneous, and the reported data is rather limited. Hence, a statistical meta-analysis was infeasible and could not be performed in our case.

On the other hand, descriptive or qualitative analysis could be performed smoothly on the reviewed papers. We made use of tabular representation of the data when feasible, and it enabled us to make comparisons across studies. Also, using the quantitative summaries of the results, we inferred the implications for future search, and consequently the existing research directions within automated user assistance.

2.4 Results

2.4.1 Overview of Selected Studies

In this section, we present the year-wise distribution of the primary studies along with the venues that they were published. This, in a sense, preliminary illustration could help see the big picture and place of research studies towards automated user assistance. Our collection of various kinds of 30 primary studies revealed the same inferences on the analogy that we argued the upward trend of automated user assistance. Figure 2.3 shows the year-wise distribution of primary studies.

Analyzing the publication channels of primary studies, there is diverse range of venues discovered. Table 2.4 shows the publication channels, the types of articles and the number of studies that fall into the channels accordingly. One of the noteworthy publication channels is the International Conference on Computer supported Cooperative Work in Design in which the topics of agents and

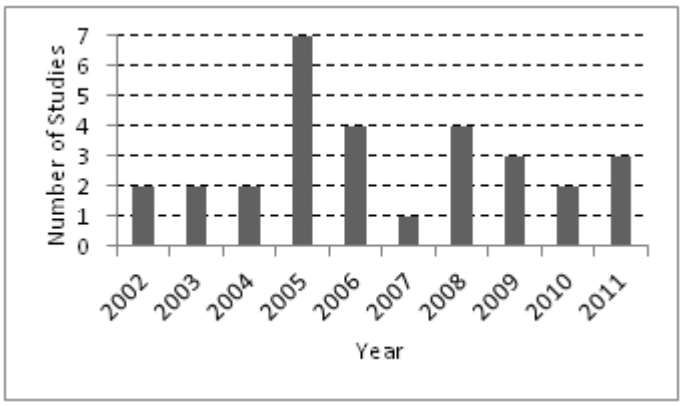

Figure 2.3: Year-wise distribution of primary studies

multi-agent systems, ontology and knowledge management, and collaborative design and manufacturing environments are contained. Also, the publication channel ACM International Conference on Design of Communication (SIGDOC) contained 3 studies in which user-centered design, methods, methodologies, and approaches are discussed.

2.4.2 Research Methods

It is very important that the primary studies explicitly define the used research methodology. By analyzing and assessing the studies reported approaches, we can draw conclusions about the strength of evidence within them. Table 2.5 provides the list of research methods used in the selected 30 primary studies.

There are five types of research methods that we looked for in the review, and the numbers reveal that most of the primary studies are based on either a single case study or an experiment. The reviewed survey-like study [27] made contributions also in the interpretation of qualitative primary studies. Also, there exists one study per the other two groups that explicitly fall into the stated research methods. The study [28] both establishes a comparison reference between three different approaches and reviews the current trends and research efforts.

Publication channel	Type	Number of studies
Computer Supported Cooperative Work in Design	Conference	3
SIGDOC	Conference	3
JASIST	Journal	2
Artificial Intelligence	Conference	1
Cooperative Information Agents	Conference	1
IEEE Transactions on Systems, Man, and Cybernetics	Journal	1
Web Intelligence	Conference	1
Cognitive Systems Research	Journal	1
Human - Centered Computing	Journal	1
Theoretical Issues in Ergonomics Science	Journal	1
Cognitive Ergonomics	Conference	1
Electronics, Robotics and Automotive Mechanics	Conference	1
Knowledge-Based Systems	Journal	1
New Generation Computing	Journal	1
Intelligent User Interfaces	Conference	1
Information Technology and Applications (ICITA)	Conference	1
Computers and Education	Journal	1
World Wide Web	Journal	1
Simulation Conference (WSC)	Conference	1
Dissertation	Thesis	1
Computers in Industry	Journal	1
IEEE Transactions on Visualization and Computer Graphics	Journal	1
Human-Computer Interaction	Journal	1
Information Processing and Management	Journal	1
Design, User Experience, and Usability	Journal	1

Table 2.4: Distribution of studies in terms of publication channel and occurrence

Research method	Studies	Number	Percent
Single-case	[29–42]	15	50.0%
Multiple-case	[43]	1	3.3%
Survey	[27]	1	3.3%
Experiment	[44–55]	12	40.0%
Benchmarking	[28]	1	3.3%

Table 2.5: Studies by research methods

2.4.3 Methodological Quality

While analyzing primary studies, we avoided the assumption that "*Unreported work means to be undone*" because it may mislead us despite being tempting. It should be noticed that a systematic literature review is a methodologically rigorous review of research results. For this purpose, using the quality checklist, we tried to address methodological quality in terms of rigor, credibility and relevance together with reporting quality. All in all, we dedicated the first four questions for the quality of reporting, the fifth question for relevance, and next three questions for rigor and the last two for assessing the credibility of evidence. Figure 2.4 shows the histogram of reporting quality results. Revealing the results of first four questions, it indicates that most of the primary studies (76,7%) are good according to reporting quality.

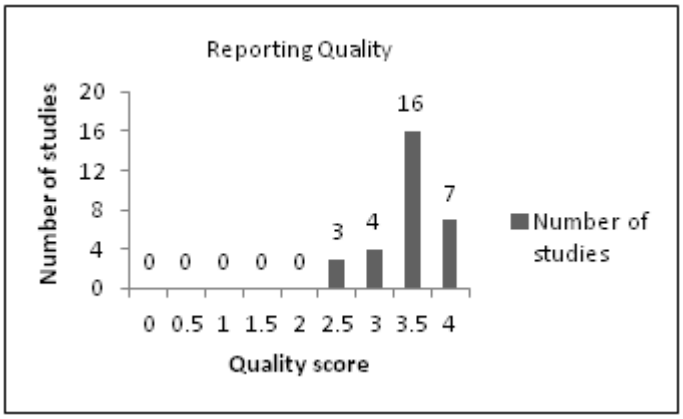

Figure 2.4: Reporting quality of the primary studies

The assessment of the relevance of our primary studies for the automated user assistance is important so that we can ensure that the results provide value for research and practice. Figure 2.5 shows the relevance quality scores that are based on the evaluation of fifth question. 60% of the studies were found to be directly relevant to the field, and 40% of them were considered as relevant to some extent.

Since the studies having implications in practice and results in research area for automated user assistance to the full extent got full score in relevance quality, this systematic literature review discovered a considerable number of relevant

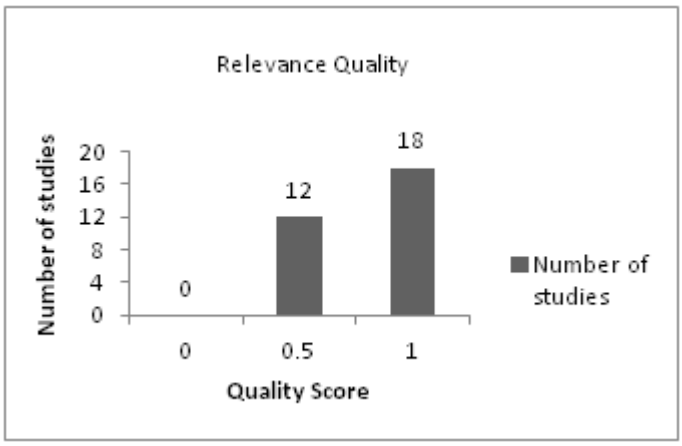

Figure 2.5: Relevance quality of the primary studies

studies despite the paucity of studies in the field. As another methodological quality measure, we should investigate whether the primary studies have thorough and appropriate proposals of approaches in the target domain. In other words, we should assess the rigor of studies, helping us notice the trustworthiness of the findings.

Figure 2.6 denotes the rigor of the research methods employed on a scale from 0 to 3. Considering the scores 2.5 and 3 as first-rates, 9 of the primary studies (30%) established the validity of their findings in a proper form. Also, the studies having scores 1.5 and 2 were treated as good mediums, and 15 of them (50%) fall into this category. As a result, 24 studies passed the assessment of rigor quality with fair scores. Two studies [39, 49] are of top quality in terms of rigor.

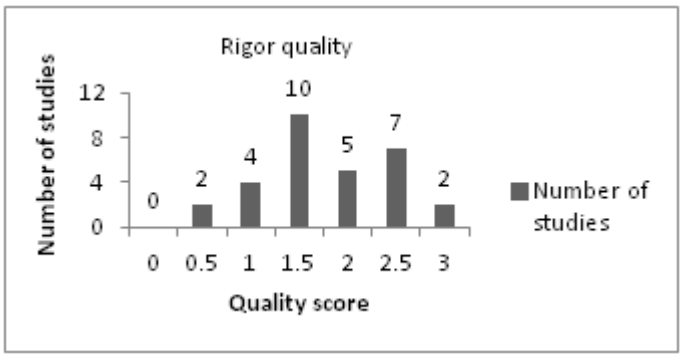

Figure 2.6: Rigor quality of the primary studies

Our last two criteria were intended for the credibility of evidence that is the

extent to which the findings and the major conclusions of the primary studies are profoundly clear, valid and suggestive. Figure 2.7 shows the histogram of quality scores based on credibility of evidence. Regrettably, we did not assess a primary study having full credibility of evidence. Besides, 14 of the studies (47%) failed in this step of quality assessment, having rather bad scores. Five studies [28,32,35,45,52] got the highest score in this rating scale, having reasonably valid and meaningful findings and corresponding conclusions.

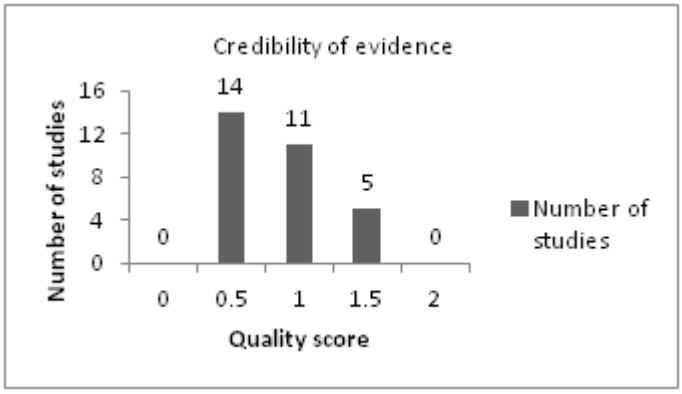

Figure 2.7: Credibility of evidence of the primary studies

Consequently, we can now finalize the overall methodological quality scores. Figure 2.8 shows the total of quality scores in terms of four criteria: reporting, relevance, rigor and credibility of evidence. 19 of the studies (63%) having scores greater than 7 are relatively good, and three studies [35,49,52] are at the head of this group being high quality. 8 studies having scores (5.5, 6.5) are of fair quality while 3 studies having scores less than 5 are considered to be of poor quality.

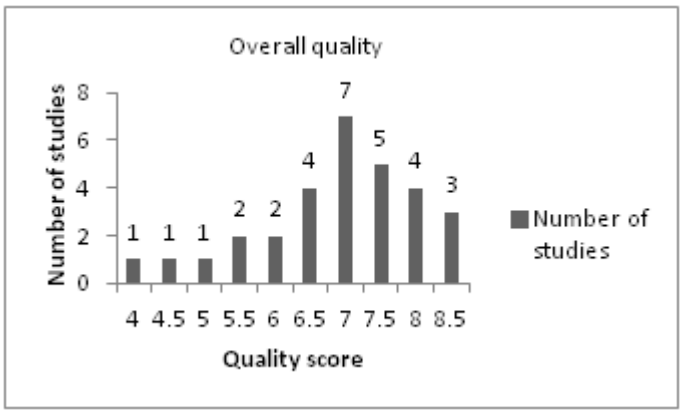

Figure 2.8: Overall quality of the primary studies

Last but not least, the above histogram shows a somewhat left-skewed distribution, meaning that the majority of the primary studies were assessed to be good to a certain extent.

2.4.4 Systems Investigated

This section outlines the results we extracted related to three main research questions. Here, we present the data extracted from the primary studies in the form of findings, separately for each research question.

RQ1. In which domains of computer science have automated user assistance techniques been applied?

Under this research question, we provide the data extracted from the main themes and motivation for the main themes of the primary studies and target domains along with their publication details in order to present the specific fields that automated user assistance techniques have engaged some attention. We categorized the studies into eight main groups. In order to answer this question, we first analyzed the targeted domains of the individual primary studies to hold a general view. Table 2.6 shows the categories of the target domains that we discovered.

Domain	Studies
Task-specific environments	[29, 45]
Adaptive, multi-layer and multi-dimensional user interfaces	[27, 28, 30, 54]
Collaborative Environments	[38, 39, 43, 51, 56]
Interactive systems	[33, 44, 47, 55]
Information Retrieval Systems	[48, 50]
Agent-based environments	[32, 34, 35, 37, 41, 46, 49, 52]
Mobile devices	[31, 36, 53]
Learning Environments for Remote Experimentation	[40, 42]

Table 2.6: Target domains of proposals

The above categorization was done by descriptive and qualitative synthesis as proposed in [20]. On one hand, the descriptive synthesis is in a way that we tabulated the extracted information about this research question, and tried to highlight the similarities and differences between primary studies.

Concurrently, the information related to this question is of qualitative nature; hence, we analyzed these parts comprising natural language outcomes. This approach is an ensemble of reciprocal translation and line of argument analysis [57]. Technically speaking, in a clustering fashion, after having identified eight cluster seeds, we assigned the studies with respect to their nearest categories.

In the first category, the primary studies raise the automated user assistance in task-specific environments. The main themes are based on providing adequate help in the cases where the concepts of tasks can be handled as granules that bring fine-grained consideration of task experience. In order words, task experience is grounded on a modeling approach, specifically ontology-based models. Interestingly, the unification of stress recognition with user assistance is proposed in [45]. Here, automated user assistance is achieved by balancing the benefits of improving user performance and the costs of performing user assistance. In study [29], two issues are discussed: the problem of finding help and the question what appropriateness of help for a specific object really means.

The primary studies that fall into the second category mainly discuss the

automation of user assistance in the user interface level. The authors of [27] consider the user assistance in three main levels: task, application and user experience. They emphasize the global complaint of the users of help systems that help content is placed in the wrong level in front, but the actual level of provision should depend on the users characteristics. From this perspective, they discuss how the location of automated user assistance in different levels of systems can be achieved. Also, in study [30], semantic transparency concept is introduced as a user interface property.

In study [54], semi-autonomous manipulation of the software systems by adaptive user interfaces is proposed. The main motivation behind this is that the more intuitively a user interface is designed, the more effectively users operate a software system. As was previously stated, the study [28] is a survey-like publication in the field of automated user assistance in which the authors basically set their sights on the recent advances in user interface level of provision of user assistance. In this paper, the benefits and advantages of intelligent user interfaces in handling the differences between users preferences, skills, experience and a lack of personalization are explained along with the challenges and approaches user for this purpose, which are: *Artificial Intelligence*, *User Modeling* and *Human-Computer Interaction*.

The third category is one of the most popular domains in which automated user assistance techniques are employed. These studies focus on either collaborative design or collaborative learning environments, and they deal with the appropriateness of automation of user assistance in these environments. Collectively discussing the main themes stated under this category, intelligent user assistance and mediums like software agents have been recognized as a promising approach to implement collaborative systems. In collaborative environments, using cognitive user models, especially user interest and user behavior models, is proposed along with the utilization of inference, knowledge update and collaboration components. The models and components are the basis of personal assistant agents from which we can derive flexibility and adaptability to effectively work with the corresponding users to achieve their goals in goal-directed collaborative tasks. In other words, the main idea here is to create user-adaptive environments

within collaborative systems.

The study [51] somehow differs from the other studies in this category. Here, the main theme is to assist teachers by a check mechanism of student participations within a collaborative distance learning environment. The idea is to detect conflicting cases in which a teacher may want to intervene. The collaboration and communication problems of students are thought to be solved by sticking with teachers and providing automated assistance to them.

Under the category of interactive systems, the proposals are somewhat theoretical and conceptual. The reason behind is, the papers here argue a claim that a domain-independent concept of user assistance is needed. There is a problem of ambiguous and uncertain user characteristics that make us have difficulties in decision making.

In study [44], the authors main concern is to keep users in productive state by means of automated user assistance. Also, a reasoning method is employed in a graphical user interface in [47]. A noteworthy statement is that users are reluctant to use help even if they encountered problematic situations. To ensure a successful user experience, the user assistance is to be automated and integrated as functionality instead of a separate presence. Going further into the matter, in the study [55], manual control of an interactive system is achieved through a predictive haptic user assistance method. It is related to offering real-time guidance for the situations such as animation control and driving. Another study [33] in this category treats user assistance as a somewhat *fuzzy* concept, and this concept is said to be requiring derivations from cognitive ergonomics. This study is a framework level of proposal leading to a comprehensive taxonomy.

User assistance is also in general use in the field of information retrieval, and it is further in a state of transition to automated techniques. This review takes these kinds of primary studies into consideration under a specific category. Automated assistance is defined as a temporal, goal-driven dialogue of expressions, actions or responses in the context of information retrieval systems.

The studies [48, 50] are based on determining whether automated user assistance improves searching performance. They are founded upon the general opinion that there is a lack of empirical evidence about the instrumentality of automated assistance during the search process. The time users need assistance and the type of assistance they look for are the factors under consideration. For example, it is stated that users seldom make use of advanced search features without even knowing how to exploit them. The extent and the circumstances to which automated assistance is of actual benefit in information searching process is indeterminate and, so to say, rather obscure.

Agent technology is one of the most encountered areas with which automated user assistance is associated. Eight of the primary studies (27%) aim at agent-based environments, and discuss the automation of user assistance in this context. The authors of [49] deal with the recommendation agents in browsing based on the integration of user profiles, navigational patterns and contextual elements. The main theme here is that pro-active and context-aware retrieval in which relevant documents are automatically presented to users according to their activities is based on the knowledge about active user goals.

The study [34] is grounded on the hypothesis that there are a considerable number of users willing to use agents provided that they know what an agent is all about. The idea here is to develop user assistance software that is highly customizable and adaptable to the user configurations and preferences. They tried to simplify the instruction process of an agent as close as possible to natural language specifications.

In the study [35], hypothesized intentions of users are the indicators of the accuracy of their workflows. Their graphical user interface mechanism is supposed to intervene undesirable situations and provide automated assistance. As was in the study [47] from another category, the authors emphasize the users reluctance to use help even in problematic cases.

An agent-based framework is proposed in [46] as a recommendation agent that offers related documents with respect to users responses in an ad-hoc fashion. The authors of study [32] propose a reflective architecture comprised of a set of

cooperative agents for modular design of application assistance software. Using several autonomous agents each dedicated to a single task is employed to provide automated user assistance.

The article [52] is based on the adoption of agents in web authoring tasks. Two tools intended to capture the users preferences and assist him/her throughout the interaction are designed, and as such minimizing the effort in webpage authoring. Unobtrusive and pro-active user assistance is pronounced also in the study BB which is based on a structured pipeline of perception, sensor interpretation, intention analysis, strategy synthesis, and actuation.

The study [37] is of somewhat unlike nature that it is a novel concept for cognitive assistance and training in manual industrial assembly aimed at designing a mobile, personal system, for the purposes of task solving and tool handling. Handling the increased complexity in industrial processes is tried to be solved by, in a sense, an agent-based fashion.

Going into the environment of mobile devices, the study [53] proposes an inventive mobile phone design tracking mobile usage pattern in which the users are synchronously aware of the costs, deviations in usage and means of cost-effective usage. Also, the study [36] presents an approach in which the mobile application shows performable tasks, modalities and commands, and provides interactive exercises for the users familiarity. Naturally, the complexity of the interactive exercises and the selection of most suitable modalities stand upon the user characteristics. The study [31] is related to the declarative description of actions, tasks, and solution methods, by which hybrid planning allows for the generation of knowledge-rich plans of action.

It is noted that there is a need for automated user assistance in learning environments for remote experimentation. The studies [40] and [42], where the former repeats the latter by the same authors, focus on intelligent user help in the context of remote experimentation. The matter of remote learning from a students perspective is that context specific help is necessary at some point of experimentation.

RQ 2. What are the existing research directions within automated user assistance?

This section explains how user assistance e techniques have been shaped up to be automated. We analyzed and tried to clarify the research space of the published literature in the field.

The primary studies come up with a wide variety of proposals of models, methods and algorithms for the attainment of the provision of automated user assistance. In order to derive comprehensive answers for this question, we further subdivided it into two questions. These sub-questions will jointly elicit sufficient amount of evidence to answer the main research question.

RQ 2.1. What are the different automated user assistance solutions used?

Under this question, we have categorized the proposed automated user assistance solutions into five main headings. Also, within each category, we looked for the different methods specifically. We observed that each primary study intersects mostly more than one specific category, meaning that the provision of user assistance is achieved through the integration of several approaches.

Modeling category comprises a wide range of concepts, and most of the primary studies include and employ at least one of the concepts under this category. We have grouped cognitive solutions together and placed under the category reasoning because in many cases these solutions are of human plausible reasoning. The other remaining solutions are more about application level of proposals, and we subdivided them into three main headings as: *Architectural*, *Framework*, and *Tool*. Among those, from the first category to the last, the definition and the design of solutions are becoming more concrete as their names imply.

Eight of the studies fall into the group of agent-based solutions in which modeling principles are followed. Two of these studies also use ontology-based methods in accordance with their agent-based technologies. Additionally, we can

observe that goal-based solutions are partially merged with agent-based proposals as we have two studies in this fashion. The study [37] is related to manual tasks in industry, and somehow the solution here is both agent-based and cognitive in terms of ergonomics. User model, user behavior models, user interest models, inference components and collaboration components are mostly used in the primary studies that are in the group of collaborative environments.

Table 2.7 shows the categories of discovered solutions with detailed specific categories. The studies [38, 43, 56] are the most active participants under these specific categories. The study [40] uses an ensemble of specific solution methods within a pipelining mechanism such as agent-based, goal-based and user models which will further lead the study to a component-based framework structure. The study O is a somewhat single performer in the fields of domain-specific models, model-driven frameworks and reference architectures which are the concepts of software engineering paradigm. The study [35] aims at goal-recognition and corresponding human-plausible reasoning.

Considering the main category of framework, it should be noted that we have discovered six different types of studies that are the single representatives of each group. Also, fourteen of all specific categories contain only one representative, possibly implying that the solution varieties among the primary studies. From the perspective of computer science, there is a good distribution among studies with respect to theoretical and applied sciences. Seven of the primary studies conclude at the design and development of automated tools for the purpose of user assistance. Additionally, almost all of these studies are inspired from modeling approaches.

Table 2.7: Categories of Automated User Assistance Solutions

Main Category	**Specific Solution**	**Studies**
Modeling	Agent-based	[32, 34, 35, 37, 41, 46, 49, 52]
	Ontology-based	[29, 30, 46, 52]
	Goal-based	[31, 35, 39, 41]
	Cognitive ergonomics	[33, 37]
	Interaction modalities	[36, 45]
	Domain-specific models	[34]
	User models	[28, 38, 39, 41, 56]
	User behavior models	[38, 39, 43, 45, 49, 54, 56]
	User interest models	[38, 39, 43, 49, 54, 56]
	Inference components	[38, 43]
	Collaboration component	[38, 39, 43, 56]
	Pipelining	[41]
Reasoning	Case-based reasoning	[42]
	Goal recognition	[35]
	Human-plausible reasoning	[35, 47]
Framework	Dynamic	[45, 55]
	Agent-based	[46]
	Model-driven	[34]
	Probabilistic	[44]
	Component-based	[41]
	Conceptual	[33]
Architectural	Multi-agent	[32, 34]
	Reference	[34]
Tool	Taxonomy-based	[33]
	Client-side	[50]
	Mobile	[31, 36, 53]
	Framework-based	[34, 38]
	Add-on	[48]

Table 2.8: Example application areas of solutions

MainCategory	**Specific Category**	**Studies**
Critical	Human stress monitoring	[45]
	Decision aiding systems	[44]
	Air traffic control	[44]
	Process control in nuclear power plants	[44]
	Real-time assistance systems	[45]
	Chemical plants	[44]
	Emergency vehicle dispatchers	[44]
	Assistance to people with disability	[44]
Web-based	E-commerce assistants	[32]
	Web Browsing	[32, 49, 51]
	Distance learning web platforms	[51]
	Distance Internet-based laboratories	[42]
	IR systems including automated searching	[48–50]
Portable appliances	Mobile devices	[31, 36]
	Lightweight devices	[36]
Non-functional concerns	Interoperation of heterogeneous software agents	[56]
	Multidisciplinary design optimization software	[39]
	Electronic multimodal operational and service assistance	[33]
Real-world	Manufacturing infrastructure	[37]
	Steered path-following scenario	[55]
Computer-aided	File manipulators	[35, 47]
	Spreadsheet-based controlling systems	[29]
	Spreadsheet and design applications	[54]

The categories of solutions would act as a roadmap for the researchers and practitioners in response to their automated user assistance concerns. There is a diverse range of solution methods for broad target domains, and employing the

discovered methods, even in a multidisciplinary fashion, would introduce fine-granular automated user assistance proposals. After having analyzed different proposals, we investigated in which environments the corresponding proposals are used. Table 2.8 presents the example application areas in which the proposed solutions are tested or validated. In this table, we have six main categories including as a result of a qualitative synthesis approach as we performed in the previous sections. The determination of main categories is based on translating each case into each of the other cases, and as such arriving at a general category through one-by-one analysis.

The category of critical applications contains only two primary studies. As was previously stated, the study [44] integrates an active sensing mechanism into dynamic Bayesian networks, and the solution method is, in a way, outlying considering the categories we identified. The study [45] which uses interaction modalities, user behavior models and a dynamic framework is rather applicable to critical systems.

Web-based application areas cover the studies from all groups of solution categories. The solutions are practicable in the cases where we are in need of web-browsing assistants or performance-enhancing information retrieval tools. The proposals that are trained in mobile devices like PDAs and mobile phones seem to integrate goal-based modeling concepts with interaction modalities. Also, the individuals having non-functional requirements, especially in collaborative environments, focus on modeling approaches to fulfill the concerns.

RQ 2.2. What are the implications of automated user assistance solutions for future search and practical use?

This question is aimed at revealing the implications behind the primary studies for further advancements. We used the categories identified in the RQ1, and presented the implications for each group accordingly in order to further the data synthesis.

Two studies in the group of task-specific environments are concerned mostly with open research challenges towards modeling practices. In the study [45],

the authors focus on the multi-modality evidences for automated user assistance such as physical appearance features, physiological measures, user performance and observed behaviors in order to compose a dynamic probabilistic inference model for stress recognition. They imply that a dynamic influence diagram model can successfully recognize human stress and provide automated user assistance seamlessly. Thanks to the just-in time assistance provided, they keep the users in a positive state by holding stress levels down.

On one hand, the study [29] puts it forward that ontology-based approach should not be bounded to task-experience dimensions in task-specific environments. They consider the possibility of sharing ontologies for help systems across applications. Also, the automated identification of right ontologies associated with users task experience is another future research direction where user models exploit task experience models. Table 2.9 presents the research methods used in these studies, and the major aims stated.

Article	**Research Method**	**Motivation**
[45]	Experiment	Balancing the benefits of improving user performance and the costs of performing user assistance.
[29]	Case study	Focusing on the problem of finding help and the question what appropriateness of help for a specific object really means.

Table 2.9: Primary studies in task-specific environments

Four studies that fall into the group of adaptive, multi-layer and multi-dimensional user interfaces arrive at the use of multidisciplinary approaches. The study [27] is more like a survey of modularity dimensions of online help, the authors discuss the variation across multiple levels of user experience. They

specified why-what-how dimensions, and the use of these dimensions in the task-application-user dimension is an open research challenge. Also, the authors suggest a rating mechanism within an automated user assistance system in which the users evaluate help content, identifying correctness of help with some reasons of errors. Also, dynamic and user-settable levels could enhance user experience in which users customize the level of explanation through slider bars in a multi-layer, multidimensional interface. This mechanism is also an implication for future research. The study [54] brings out episode-based learning compared to other approaches for user assistance. Without a definite conclusion based on a large sample of subjects, their method presents automated learning and personalized adaptation according to some empirical results. Their implication for future research is the extension of this approach as a generic tool in an application-independent context for reusability.

In the study [28], the research scope is limited to Artificial Intelligence, User Modeling and Human-Computer Interaction. The authors suggest multi-disciplinary approaches in the development of intelligent user interfaces, and they emphasize an open research challenge that is the limited amount of empirical evaluation of adaptive systems. In order to manifest the advantages of adaptive interfaces compared to non-intelligent interfaces, more research should be carried out together with strong empirical evidences. The study [30] focuses on the concept of semantically transparent interfaces to provide automated user assistance based on ontology-structured text collections. They imply the possibility of integrating semantic transparency into multi-layered interfaces. The future advancements will certainly fall into the intersection of Human-Computer Interaction and Knowledge Management.

The next category that has created a great deal of research interest for automated user assistance is collaborative environments such as design and learning. Table 2.10 shows the motivations behind these studies in order to recall the study settings before going into the implications for research and practice. The authors of study [56] state that the design of personal assistant agents in collaborative environments is a new research area. The major issues to be taken into account are user modeling, reasoning and design making, and collaboration

Article	Research Method	Motivation
X	Case study	How the intelligent assistant agents reason on user interest model and user behavior model by utilizing the collaboration component
S	Multi-case study	Effective and appropriate user model representations, inference approaches and collaboration mechanisms
P	Case study	Benefiting from agents flexibility and adaptability to design and develop intelligent user assistance systems
L	Experiment	Solving the communication problems between students by incorporating an intervention mechanism
Y	Case study	Developing a personal assistant that helps the user participate in goal-directed collaborative tasks by advising the user, monitoring the routine task procedures and creating user-adaptive environment

Table 2.10: Primary studies in collaborative environments

mechanisms. They specifically focused on encapsulating the specific collaboration mechanisms into a generic collaboration component and as such providing automated user assistance in collaborative design environments. The proposed approach is said to be effective for the provision of automated user assistance. Also in the study [38], the same authors propose an agent-architecture in which the collaboration between the agents helps the engineers to finish the collaborative design task successfully. The study [39] is based on exploiting the user model to capture the user's interests and behaviors and as such providing automated user assistance. According to the authors, in real world engineering design environments, the proposed approach, which is based on a collaborative personal assistant agent framework, would be used. They also suggest an approach in

which the users are able to customize their user models in a collaborative design environment.

The study [51] presents an alert-based approach with behavioral and context analysis for teachers in collaborative work environments. For this purpose, an intelligent agent integrated with a web-based distance learning platform is designated. The use of automated user assistance increased the number of collaboration conflicts detected. The means of evidence behind here is that the conflict detection accuracy was validated both with artificial data and with a controlled group of users in a real course. They leave a concrete future work that is the experimentation with larger number of courses and groups of students in order to capture more conflicting cases and tune the model accordingly. The study [43] states a long-term goal and a possible implication for research that is to develop a general collaborative personal assistant agent framework applicable to various collaborative engineering environments.

The primary studies in the domain of interactive systems have also some notable implications. The authors of study [44] advanced the theory and application of efficient information fusion in humancomputer interaction and arrived at a conclusion that an affective state-detection system itself is not enough to provide automated user assistance. Their procedures and paradigms are said to be insufficient to effectively recognize the states of subjects considering the strictness of the target domain. The proposed information-theoretic mechanism is intended to systematically model the concepts of dynamic Bayesian networks in which users affective state is captured to determine the time and the type of assistance. Also, in this domain, they integrated an active sensing mechanism with a dynamic Bayesian network in order to provide automated user assistance. In the study H, the intelligent file manipulator produced human-like reasoning according to evaluation results. Their proposed human plausible reasoning theory, which is attached to an interface, is expected to establish an upper limit that we can derive benefits from an intelligent interface.

The study [55] presents a haptic look-ahead guidance method in which the

cues need to be compared with visual guidance. The authors state the advantages of predictive guidance, and possible application of this approach to highly dynamic tasks including intelligent-system supports. The authors of study [33] state that assistance design is a communication field handled by fuzzy terms and different implicit perspectives. The proposed taxonomy is intended for the design of automated user assistance systems.

The articles in information retrieval systems got very high scores in terms of the quality assessment. The study [50] in the field of information retrieval proposes the design of a general-purpose automated assistance application using implicit feedback. The noteworthy finding here is that automated assistance systems may improve the Web searching performance by offering more number of relevant documents. For future studies, predictability is brought forward as a concern to be considered in these systems since the users utilize these systems in a predictable manner. Also, instead of offering automated user assistance at the query level, the session level assistance along with a more personalized and targeted approach is practicable being more advantageous. The main implication of the study [48] is that detecting the patterns of usersystem interaction, we can customize automated user assistance systems providing just-in-time guidance.

In the domain of agent-based environments, there are eight studies (27%) that are collectively assessed in terms of the implications. In the study [32], the authors aim at ensuring coordination among various assistants working for the same application by offering a great degree of flexibility and modularity in the design and implementation of assistants. They show the appropriateness of an assistant set based on reflective software architecture for a web browser which will support e-commerce activities. They also discussed the modularity and reusability issues of their proposed assistants. The run-time integration of highly-cohesive assistants and the customization of them for a variety of applications are the major findings.

The authors of study [37] highlight the lack of methods helping operators in executing complex, manual assembly tasks, and an overview about personal

cognitive assistance in production environments. To the best of their knowledge, in this field, the only available instruments are training courses, text-based documentations and learning-by-doing, which are time-consuming, extensive and inadequate. The main implication is that further advancements are to be in favor of humans instead of substituting them. The study [52] provide a particular solution which tries to find out the boundaries of *WYSIWYG (What You See Is What You Get)* approach, rather than a universal one to the issue of end-user authoring, but. The approach is multidisciplinary in a way that it is based on Programming by Example and Model-Based User Interfaces paradigms. The study [41] deals with the core question of providing automated user assistance in smart environments. The authors suggest the integration of modeling and simulation efforts for the different types of models at different levels of abstractions, whereas they exploited them separately. Also, a component-based modeling and simulation framework for smart environments would be the major implication of this study for future research.

The study [34] reveals a software engineering issue related to agent-based environments which is the user customizations issue. The findings indicate that the choice of the dimension in which the software architecture will be modularized is of utmost importance. From this perspective, the authors bring out the need for better software architectures, where security and privacy issues are also considered, to build personalized user agents. The study [35] states that the incorporation of intelligence to a graphical user interface will bring out much more productive users. In order to adapt this approach to a different domain, the kinds of errors that users usually make should be revealed by an empirical study by which the reasoning mechanisms are tuned accordingly. It is empirically assessed that the users of an interface may not be aware of the situations in which they actually need help.

It is stated in the study [49] that the use of browsing assistants enables us to capture interests from an ongoing user activity and to detect out-of-context interests. The proposed approach is to specify and associate browsing activities in user profiles. In this way, consistent, comprehensive and meaningful contexts

can be identified. Also, the empirical results in this study indicate that the extraction of association rules describing browsing patterns at a conceptual level assists in estimating user interests in a browsing session. The study [46] proposes the approach of recommending related documents according to the user feedback inferred from similar-page searching mode. They highlight an implication that capturing semantic relations through some semantic-based improvements in the architecture and defining relative ontology among concepts during Web navigation are the open research directions.

In the category of mobile devices, we have three of the primary studies. First, the authors of study [31] showed a goal-based approach, which is based on automated reasoning techniques, on user plans, just like in plan recognition, and a possible speech dialog. Their major conclusion is that assistance in this method is provided only in case it is needed, and as a future work, a tutoring system such as a proactive electronic instruction manual will be useful for the purpose of automated user assistance.

The study [53] shows an approach which exploits patterns in mobile usage. The authors applied this mechanism as an automated user assistance solution, and they propose it as a highly-used and interesting component to the mobile phones. The study [36] is based on the provision of automated user assistance for mobile devices in ubiquitous computing environments. The proposed approach handles the group of inexperienced users since their acquaintance improves slowly. A possible advancement would be the evaluation of this approach based on some user studies along with a broader range of user groups.

One of the interesting categories of primary studies is the learning environments for remote experimentation. In this category, we have two participants having their implications to the best of their findings and conclusions. The study [40] presented an intelligent context specific help system in terms of its embryonic stages of architecture and design. Their implication is that the upward tendency of the use of automated user assistance solutions in remote experimentation environments will certainly increase. In the study [55], the proposed method offers considerable benefits in the field of remote experimentation. They initiated the

development of an intelligent context-sensitive help system that shows promising results. The facilitation of collaborative experimentation between students is achieved through the addition of remote cooperative working functionality to the remote laboratory.

RQ 3. What is the strength of evidence in support of the stated findings?

As was previously stated, we have to reveal the plausibility of the reported results and findings for the readers of an SLR. Therefore, this research question was formed to identify the extent to which the readers of the review can rely upon the implications. In the literature, there are some definitions that address the strength of evidence in systematic reviews. We used the definitions from *GRADE (Grading of Recommendations Assessment, Development and Evaluation)* working group [58] in order to avoid doubts imposed by the weaknesses of evidence hierarchies. Table 2.11 shows the definitions that we used and adopted from [58].

Table 2.11: Definitions used for grading the strength of evidence

Grade	**Definition**
High	Further research is very unlikely to change our confidence in the estimate of effect
Moderate	Further research is likely to have an important impact on our confidence in the estimate of effect and may change the estimate
Low	Further research is very likely to have an important impact on our confidence in the estimate of effect and is likely to change the estimate
Very low	Any estimate of effect is very uncertain.

The above definitions are used to determine the strength of evidence considering four elements that are: study design, study quality, consistency and directness. Concerning study design, the *GRADE* system assigns higher grade to experiments than to observational studies. Twelve (40%) primary studies are of experimental type in the review. Also, Table 2.12 shows the average quality scores corresponding to this group of studies. As a result, our first categorization

about the strength of evidence based on study design is moderate.

Table 2.12: Average quality scores of experimental studies

Experimental studies	[44–55]
Number of studies	12
Mean quality score	7,375
Standard deviation of quality score	1,00284

Considering the quality of the studies, the issues of bias, validity and reliability are not discussed explicitly in most of the primary studies. Besides, none of the studies got full score in our last quality assessment criterion that is the statement of limitations where seventeen of them discussed to some extent and the remaining did not state any limitations at all. Only seven studies (23%) stated the findings clearly in terms of creditability, validity and reliability. Hence, there is an inevitable risk of bias due to the low quality scores in terms of the study design, requiring a great deal of circumspection.

With respect to consistency which refers to the similarity of estimates of effects across studies, we realized slight differences among articles. Since we categorized the primary studies into eight main groups, we can only talk about specific groups evidences. We did not perform a sensitivity analysis by excluding poor quality studies or studies of a particular type because of the varying presentation of both objective and empirical results. Besides, the synthesis of quantitative results was not feasible because the outcomes of primary studies were not presented in a comparable way. In other words, the reporting protocols differ from study to study, causing us to perform the data synthesis in a more qualitative or descriptive way which was not solely desired. Consequently, we arrived at a conclusion that the results are barely consistent owing to the large extent of imprecise evidence.

With respect to directness which is defined in [58] as the extent to which the people, interventions, and outcome measures are similar to those of interest, a narrow range of the studies reported comparisons of interventions. This is most probably because of the wide range of solution domains in the field of automated user assistance. At the very beginning, the *intervention* in this review was composed of *context-sensitive*, *process-sensitive*, *embedded*, *intelligent*

and *adaptive* user assistance techniques, and we did not impose any restrictions on *outcomes* by foreseeing the plenty of factors of importance to both researchers and practitioners. Regarding the directness of evidence for each category of primary studies, most of the studies were performed by, so to say, the pundits of automated user assistance. The subjects were mostly academics and representative user assistance specialists. A considerable amount (65%) of primary studies was performed on the basis of industry-as-laboratory. Thus, the total evidence based on directness of the primary studies is *moderate*.

Collectively assessing the four elements about the strength of evidence of the reviewed literature, the analysis of the impact of automated user assistance solutions resulted in a ***moderate*** grade. Also, considering our research questions, qualitative studies were more appropriate, in a sense, than quantitative ones for revealing more types of solutions and deciding on the technologies to employ. In other words, the likely paucity of empirical studies in the specific fields of automated user assistance results in encompassing as many types of studies as possible, which we settled as a strategy in this review.

2.5 Discussion

2.5.1 Benefits and Limitations of Automated User Assistance Systems

This systematic review is originally based on the findings of individual primary studies that address the introduction and adoption of automated user assistance techniques. The majority of the papers present positive statements in support of their findings towards automated user assistance. The qualitative data reported is mostly in parallel with the quantitative results.

The benefits of automated user assistance are strongly emphasized in software intensive systems in which the interaction with the users is of vital importance. The main argument here is that we can ensure better user experience through

better user assistance. The development and need of autonomous agents for user assistance is increasingly growing, especially in the industry. Context awareness and intelligent user interfaces provide the users just-in-time assistance by which they stick with the concerned work-flows.

2.5.2 Limitations of the Review

To the best of our well-prepared search strategy, we tried to avoid some possible bias in the selection of publications. We used a *quasi-gold standard* (QGS) that hopefully helped us in forming an optimal search strategy. We applied word frequency and statistical analysis tools on a well-controlled and piloted set of studies in order to capture better keywords for the review. However, it should be noted that there were some other keywords being discipline, or category-specific in our case.

We performed the inclusion/exclusion procedures on a well-established screening of primary studies. We included both qualitative and quantitative studies in almost all respects, but a rather unwell yield of this decision is that we could not present the empirical results in a uniform way. Alternatively, we translated the results of primary studies in each group as much as possible in order to attain uniformity at least in specific categories. Anyhow, we aimed at presenting the variety of automated user assistance solutions, and the *comparison* criterion was not considered because we did not intend to compare any practice with the interventions specified.

2.6 Concluding Remarks

There has been a relatively good interest in the research area of automated user assistance from a diverse range of domains. According to a thorough preliminary analysis and to the best of our knowledge, there has been no study to systematically undertake the state of automated user assistance systems. We tried to reveal

the body of multidisciplinary research on this field by systematically analyzing the published literature since 2002. We reviewed 575 papers that are discovered using a well-planned review protocol, and 30 of them were assessed as primary studies related to our research questions.

Considering the diversity of primary studies with respect to main themes, they were categorized into groups according to the nature of each research question. We reported a lot of benefits of automated user assistance solutions in several domains along with the implications for the future research and practice. Also, the strength of evidence is relatively good in spite of the diversity of both target domains and solution methods.

A significant finding of this review is that there are so many empty rooms for further research and practice in this field in response to the implications identified. To sum up, our work has a pioneering value of contents providing a roadmap of current trends in automated user assistance systems for both researchers and practitioners.

Chapter 3

Systematic Review of Embedded User Assistance Systems

This chapter is based on our systematic literature review of *Embedded User Assistance* systems. Section 3.1 presents an overview of *Embedded User Assistance* systems. Section 3.2 describes the research method used in this study. Section 3.3 shows the results of this systematic review. Finally, the chapter ends with Section 3.4 stating some discussion and concluding remarks.

3.1 Overview

To recall the preceding chapters, user assistance can be considered as a guided assistance to a user of a software product. It helps accomplishing tasks and ensures a successful user experience [1]. The most conventional form of user assistance is an off-line printed user manual which is rather separate from the system. Up to now, most technical communicators are focused on traditional documentation deliverables, such as [6]:

- User guides
- Online help

- Reference manuals
- Tutorials (print and online)

These documents are inherently reactive, and accessed only when the user fails to perform a task correctly or when the user does not know how to proceed. Hence, these kinds of user assistance typically disrupt the users flow of work to provide the information he or she seeks to perform the task correctly. Research on user assistance has indicated that it is due to this separate effort and disruption of the users flow of work, that users are reluctant to using help. Due to these issues, *online help systems* have emerged with providing information to the user in an electronic format which can be accessed directly in the application. Until recently, *online help systems* have mostly followed a topic-oriented approach which refers to the guidance that can be requested based on keywords. This type of user assistance is presented in different formats such as HTML, text, or PDF [1].

Embedded user assistance systems have come into play as a special form of *online help* in which the documentation of the application resides within the application. The term *Embedded user assistance* usually refers to help that is part of the real estate and behavior of the software application to which it relates, instead of a reactive documentation in a separate place. Users do not have to intervene the work-flows to access the user assistance since it is presented to them as a built-in aspect of the user interface itself. *Embedded user assistance* can take many forms which are stated in [59] as:

- Simple static text positioned beside or beneath a field or control
- A dedicated user-assistance pane that pro-actively displays useful information about the screen that the user is viewing
- Text hyper-links, often worded as questions, that display useful information or explanations

By designing graphical user interfaces to answer our users'questions in advance, we can develop more intuitive software systems. We pro-actively prevent

task failure and keep the user within the task. Technical communicators, engineers and visual designers have long been using techniques of embedded user assistance. Nowadays, new techniques have been employed to provide even proactive assistance to users [6].

We can list some traditional forms of embedded assistance as follows:

- Text in the user interface
- Wizards
- Do-it-for-you functionality, such as coaches and show-mes

Today, most of the user interfaces are comprised entirely of text. The techniques of *Embedded user assistance* have been employed by many companies. Also, researchers and practitioners are looking for novel technologies to adopt *embedded* and *non-obstructive* assistance, and we will analyze the efforts done recently for this purpose.

3.2 Method

3.2.1 Review Protocol

Before conducting the review, we need to plan the complete set of review activities and deploy them on a proper schedule. There have been several approaches undertaking a comprehensive review protocol. We used the guidelines proposed in Kitchenham [20], and arranged the activities in accordance with these somewhat rigid set of stages. Basically, the systematic review can be divided into three main activities: planning the review, conducting the review and reporting activities. Under these main categories, we have several steps to perform.

The overview of planning the review is to perform a feasibility study in advance in order to form a initiator of further review activities. We need to ensure

where the proposed research fits into the current body of knowledge and identify the existing basis of the target field in hand. In order to obtain a sound basis, we have to reveal any existing studies that relate to our phenomenon. After, we can formulate our research questions by taking several viewpoints (e.g. *population*, *intervention*) into account. The contextual quality of research questions will determine the effectiveness of the review. After the determination of review objectives, we should develop a detailed review protocol that will be the anchor of research activities. Just as in the case of most systematic review, this protocol determined the formulation of research questions, search strategy, inclusion, exclusion and quality criteria, data extraction and constitution of data synthesis.

The stage of conducting the review reflected the steps specified in the review protocol in detail. In other words, it is the process of realization of a review. We decided the ways to perform these steps which will be explained in the following sections. Afterwards, the reporting activities finalized the review process.

3.2.2 Research Questions

This review is intended to shed some light on the undisclosed state-of-the-art of embedded user assistance systems. Although embedded user assistance is not a novel technology, there is a proliferation of proposals, possibly oblivious of recent advances in the field. To the best of our knowledge, there have been no reviews or comprehensive surveys to analyze the proposed models, methods and algorithms. In this case, the major objective of this systematic review is to reveal and assess the evolving art and science that focus on the embedded provision of user assistance. Also, we aimed at both identifying gaps and commonalities in the published literature and evaluating current best practices. The evaluation is to be based on an unbiased quality assessment procedure.

The specification of research questions for the mentioned purposes is of utmost importance. The concreteness of research questions affects all stages of systematic reviews, leading to proceed with a well-defined search strategy. In order to provide a baseline for state-of-the-art, we need to first characterize the research studies

carried out so far. The actual research space of embedded user assistance can be discovered from different points of view. We followed a somewhat top-down approach while specifying the research questions and formulated our first research question as:

- **RQ 1. In which areas of software-intensive systems have embedded user assistance techniques been applied?**

  Having seen the research directions and target areas in the field, we should, in a way, categorize the motivations behind the adoption of embedded user assistance. Since it covers a diverse range of approaches, embedded technologies are applicable to various software-intensive systems. To associate the motivations with target domains, we formed our second research question as follows:

- **RQ 2. What are the motivations behind embedded user assistance?**

  The third research question is aimed at displaying the extent to which the scientific studies have reached so far. In order to make this case clear, we need to both analyze the techniques and corresponding research directions. Hence, this question is formed as a combination of two sub-questions to well distinguish between different claims.

- **RQ 3. What is the state-of-the-art on embedded user assistance?**
  - **RQ3.1. What are the techniques applied?**
  - **RQ3.2. What are the research directions followed?**

  Finally, we can observe how embedded user assistance is being experienced and what impacts it has in terms of benefits derived and limitations imposed. For this purpose, the last research question is:

- **RQ 4. What is the strength of evidence in support of the stated findings?**

Considering the question structures, the whole domain of embedded user assistance is the *population* in this study. *Intervention* addresses general, functional, procedural and conceptual types of embedded user assistance. The *comparison* viewpoint is not applicable in our case since we do not want to state some comparative evidence. As our fourth research question aims to discover existing *outcomes*, this criterion is also not limited to any specific factors of importance. Some systematic reviews might choose to impose several restrictions on the *context* and *experimental design*, but in our case, the study designs appropriate to answering the review questions would, most probably, be of a limited number. Hence, these two concerns are also not taken into account in this study.

3.2.3 Search Strategy

The rigor of search process is of utmost importance which distinguishes systematic reviews from traditional reviews. We have to follow an unbiased search strategy to retrieve as many related material as possible. We set the boundaries of our study in the last decade to maintain an opinion on the evolution of embedded user assistance. Thus, the starting point of our review is January 2002, propped up by the discovery that we retrieved good-quality articles within this period. The end time of the review is January 2012 since we employed our search strategy at this time. In addition to automatic venue searches, we manually search some prestigious publication channels that the researchers and practitioners have as their targets.

After having determined the publication channels to be searched, we formulated our main search string after some pilot searches. We adopted the approach to formulate a search string from our previous systematic review in Chapter 2. The motivation behind this approach is to improve the scientific rigor in a systematic review through analyzing well-known publications and reveal better search keywords. The publications available beforehand are based on our background knowledge on embedded user assistance. Their methodological quality in terms of reporting, relevance, rigor and credibility of evidence were validated to ensure their possible improvements on the search strings.

We decomposed our research questions into keywords along with a list of synonyms, abbreviations and alternative spellings. Some preliminary searches were performed to assess the research space of potentially relevant studies. Also, we searched some prestigious company journals and community channels. In order not to allow a likely publication bias to lead a systematic bias, we have to go beyond standard search strategies by paying great attention on channels that cannot be revealed by automatic searches. The search was applied on: journal articles, workshop papers, conference papers, some prestigious company journals and conference proceedings, grey literature, research registers, and the Internet. The search strategy limited our search to the papers written in English.

Finally, our search string used in this systematic review is a Boolean expression like:

(X1 AND (X2 AND (X3 OR X4 OR X5))) AND (Y1 OR Y2 OR Y3 OR Y4 OR Y5)

where,

X1 $\Rightarrow$ Embedded

X2 $\Rightarrow$ User

X3 $\Rightarrow$ Assistance

X4 $\Rightarrow$ Guidance

X5 $\Rightarrow$ Help

Y1 $\Rightarrow$ Interface

Y2 $\Rightarrow$ Tool

Y3 $\Rightarrow$ Software

Y4 $\Rightarrow$ System

Y5 $\Rightarrow$ Context* (-sensitive,-aware,-based,-dependent)

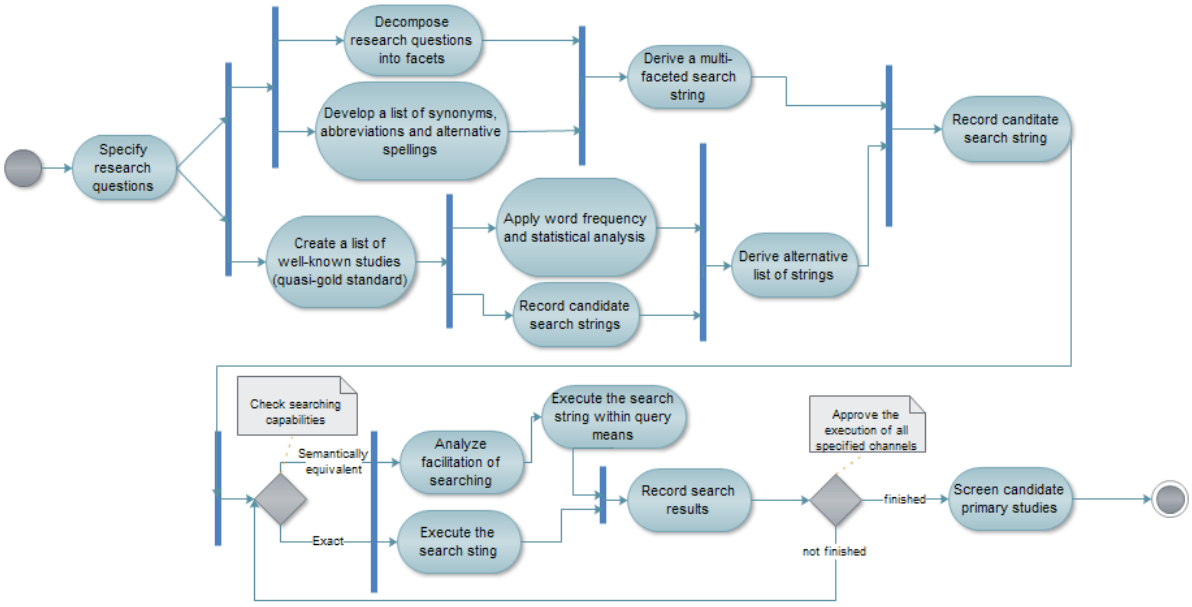

Figure 3.1: Activities under search strategy

The big picture of our search strategy is shown in Figure 3.1. Here, we state only the major activities to reveal the most important parts of our well-organized review protocol.

3.2.4 Study Selection Criteria

As it is shown in Table 3.1, we identified 550 papers in total (after removing duplicates from different publication channels) by a thorough literature search. Search methods include automatic and manual searches where the former was performed by direct search strings, and the latter is performed by manually browsing journals and conference proceedings. Since the main objective of this systematic review is to reveal the state of art advances in embedded user assistance systems, it is necessary to determine proper inclusion/exclusion criteria on the diverse set of identified papers that undertake the topic differently.

We included the articles that include embedded user assistance either as only one element or as the main concern. Also, the studies from a diverse set of

Source	Initial results of applying the search string	Papers left after Exclusion Criterion 1	Papers left after Exclusion Criterion 2	Papers left after Quality Assessment Criteria	Search Method
IEEE Xplore	217	34	7	7	Automatic
ACM Digital Library	128	22	8	7	Automatic
Wiley Interscience	18	5	1	1	Both
Science Direct	18	6	1	0	Automatic
Springer	54	19	8	5	Automatic
ISI Web of Knowledge	30	6	0	0	Both
IngentaConnect	26	7	1	1	Both
Taylor and Francis	16	4	1	1	Both
Other Channels	43	24	13	2	Manual
Total	**550**	**127**	**40**	**24**	

Table 3.1: Publication Channels

subjects of investigation (i.e. *students*, *professional developers*) are taken into account. Moreover, we did not look for any specific type of *outcomes*, and we did not impose any restrictions on the *context* and *experimental design*. The *intervention* criterion was strongly taken into account in the terminal stages of inclusion.

We divided the identified papers into two equal sets, and we kept the record of the number of observed agreements and also computed the Kappa coefficient of agreement, which is a statistical measure of inter-rater agreement for qualitative (categorical) items. According to the inclusion/exclusion criteria, the number of observed agreements was really high (97%). Kappa coefficient was calculated as 0.901, which is almost prefect according to [60].

Peer-reviewed articles were analyzed according to the following exclusion criteria:

- ***Exclusion criterion 1***:
  - Do not relate to software-intensive systems
  - Do not relate to user assistance
  - Do not state any application of techniques, algorithms or methods to provide user assistance
- ***Exclusion criterion 2***:
  - Abstracts or titles that do not mainly discuss the provision of embedded user assistance were excluded
  - Abstracts or titles that do not propose an approach to automate user assistance on the basis of the alternate terms that we have discussed were excluded

Studies were eligible for inclusion if they were elected as quite valued for embedded user assistance, which is assessed by means of the quality checklist. In other words, we used our quality assessment checklist to enhance the study selection criteria as proposed in [20].

3.2.5 Study Quality Assessment

The quality assessment was substantially performed at the earlier stages of study selection process since we meant the quality values to assist in study inclusion process. For this purpose, we recorded initial rater agreements for all of the studies left after the second exclusion criterion. This is done by critically appraising the studies, and assessing their quality according to a well-defined checklist.

Each of the 40 studies that remained after exclusion criteria was assessed according to the checklist in Table 3.2. This checklist includes four main quality grades that are: reporting, relevance, rigor and credibility. The quality checklist was formed according to the factors that could bias study results.

The quality items are deployed on a numerical scale to observe the overall strength of evidence in primary studies. The scoring procedure was Yes=1, Partly=0.5 and No=0 during the assessment. These values were used to devise detailed inclusion/exclusion criteria. Since we did not impose any restrictions on the *context* and *experimental design*, the assessment covered both qualitative and quantitative studies. Therefore, we constructed the checklist in a way that it addressed both qualitative and quantitative studies. We collectively applied the procedure, and in case of disagreements, we discussed the conflicting issues.

We first analyzed the quality of *reporting* in terms of rationale, aims, scope and context. Also, we looked for comprehensive research published and eliminated lessons-learned papers in advance. Then, we discussed the issues of *relevance* and judged the studies according to their implications for software industry and research community. This criterion is so important that in this way, we can ensure that the results provide value for research and practice.

No	**Question**
Q1	Is the paper based on a comprehensive research (Is it more than an experience report that is merely based on expertise?)
Q2	Are the aims and objectives were clearly reported (including a rationale for why the study was undertaken)
Q3	Are the scope and context in which the research was carried out clearly stated?
Q4	Is the reporting of study clear and coherent?
Q5	Is the research useful for software industry and research community?
Q6	Are the subjects of investigation adequately described? For example, SE experience, type (student, practitioner, consultant), nationality, task experience and other relevant variables.
Q7	How clear is the basis of evaluative appraisal?
Q8	How well has the approach to, and formulation of, analysis been conveyed?
Q9	How clear are the assumptions/theoretical perspectives/values that have shaped the form and output of the evaluation?
Q10	Does the study clearly provide stated findings that relate to the aims of research with credible results and justified conclusions?
Q11	Is there an explicit statement of the limitations?

Table 3.2: Quality Assessment Checklist

In order to hold a view about the trustworthiness of the findings, we should pay great attention on the rigor of studies. Thus, we investigated the basis of analysis

and evaluative appraisal along with the assumptions, theoretical perspectives and values. Finally, we discussed the issues of *credibility of evidence* that is the extent to which the findings and the major conclusions of the primary studies are clear and meaningful.

After employing the quality assessment checklist, we excluded 16 of the 40 studies because of their poor quality values, and 24 studies remained as our primary studies.

3.2.6 Data Extraction

The full-texts of 24 primary studies were read, and relevant data required fulfilling our research questions and quality assessment procedure was to be extracted. We separately recorded the quality values and review data since quality assessment was one of the initiators of study selection. Once we developed the review protocol and our research question, for the sake of an unbiased review, we specified the elements to be extracted and record them in our data extraction forms in advance.

After independent data extraction, data from both researchers were compared and disagreements were resolved by consensus. For this purpose, a sample of papers was chosen and read by the researchers, and the data extraction form was piloted on them. Thus, it is improved through a series of iterations and inter-researcher consistency was assessed as was in the quality agreements.

The first part of data extraction form was composed of standard information about this step such as name of the reviewer, date of data extraction, study ID, title, authors, journal, publication details, a brief summary and space for additional notes.

Secondly, the data addressing our research questions in individual studies was extracted. At this step, we looked for a series of elements that are: definition of embedded user assistance given in study, motivation, objectives/research directions, background of study (industrial vs. laboratory), subjects of investigation, target area, technique for embedded user assistance, study design, main factors

of importance, assessment approach, findings, benefits/limitations of embedded user assistance discussed, implications for future research and major conclusions. Also, individual extraction activities brought some personal notes and publication details.

The important parts of primary studies were highlighted by means of an auxiliary editor in order to ease recurrent extraction activities. In other words, we noted the sections of papers where the corresponding information resides. Also, for the sake of easy data synthesis, we classified the data fields and tried to group resembling data beforehand.

The extracted data was tabulated to show:

- The types of definition and adoption of embedded user assistance (addressing RQ1)
- The domains that studies aim at and the specific target areas (addressing RQ1)
- The actual backgrounds of studies (i.e. the cases they conducted) (addressing RQ1)
- The motivations behind the use of embedded user assistance (addressing RQ2)
- The concerns leading the adoption of embedded user assistance (addressing RQ2)
- The subjects of investigation (addressing RQ3)
- The techniques used for embedded user assistance and the objectives along with research directions (addressing RQ3)
- The benefits and limitations, and implications for future research(addressing RQ3)
- The study designs observed (addressing RQ3)

- The findings and major conclusions (addressing RQ3)
- The assessment approaches used by the authors (addressing RQ4)

3.2.7 Data Synthesis

In this step, we collate and summarize the extracted data from the primary studies both qualitatively and quantitatively. Since we used a classification scheme in data extraction, the data categorization was performed comfortably. The key concepts were organized in tabular forms in order to make comparisons across studies. We synthesized the quantitative and qualitative studies separately, and then attempted to integrate them. Qualitative synthesis was based on natural language results, findings and conclusions. At this stage, we aimed at identifying higher-order interpretations by means of the three approaches proposed in [20]:

Reciprocal translation is about translating each case into each of the other cases.

Refutational synthesis is about translating studies that are implicitly or explicitly refutations of each other.

Line of argument synthesis is the process of analyzing the individual studies at first and then undertaking the set of studies as a whole. We made great use of this technique while synthesizing qualitative data.

The protocols presenting quantitative information seemed to be too much varying as it can be expected in the field of software-intensive systems. Therefore, we transformed the results of quantitative information into descriptive or qualitative analysis.

3.3 Results

3.3.1 Overview of Selected Studies

In this section, we analyze the primary studies according to their publication years and corresponding publication channels. As was previously stated, we set our sights on the past decade. Although the publication year could be a decision criterion to be used as an inclusion/exclusion factor we did not take it into consideration while assessing primary studies. For this situation, we examined whether quality scores were correlated with the results reported in distinct time periods. However, we did not observe such a relation between studies from different years, meaning that uniformity was preserved among studies in terms of publication years. Considered closely in Figure 3.2, 16 primary studies (67%) were published in the last five years .

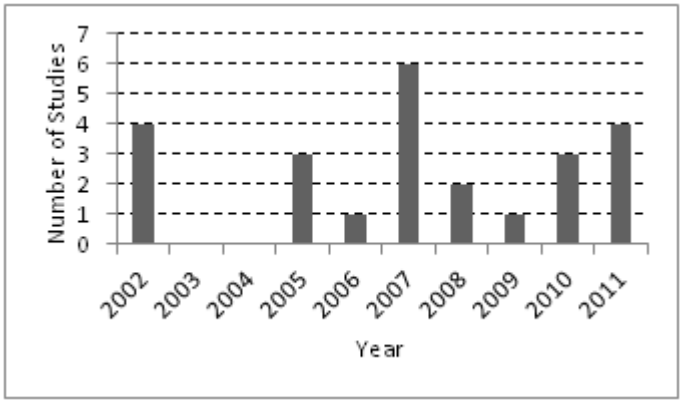

Figure 3.2: Year-wise distribution of primary studies

We analyzed the distribution of the studies considering their publication channels. Table 3.3 shows the occurrences per publication channels observed. *ACM interactions magazine* features the studies on human-computer interaction and interaction design communities as we have two studies in this group. Also, two of the studies were published on *ACM International Conference on Design of Communication (SIGDOC)* in which the topics of design principles, usability/user studies, user interface design and human-computer interaction are discussed. The

study [61] was found as a grey literature that is difficult to find via conventional channels.

Publication channel	**Type**	**Number of studies**
interactions - Help! User assistance and HCI	Magazine	2
SIGDOC	Conference	2
Technical Communication	Journal	1
Transactions on Aspect-Oriented Software Development	Book	1
Human-Computer Interaction	Book	1
Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing	Conference	1
Military Communications	Conference	1
Professional Communication	Conference	1
Systems, Man and Cybernetics	Conference	1
Mobile HCI	Conference	1
Knowledge-based intelligent information and engineering systems	Conference	1
Human Factors in Computing Systems	Conference	1
Intelligent Environments	Conference	1
Online Documentation and Content	Conference	1
Cybernetics and Intelligent Systems	Conference	1
Integration of Knowledge Intensive Multi-Agent Systems	Conference	1
Ambient Intelligence	Conference	1
Information School Repository	Technical report	1
Simulation Conference (WSC)	Conference	1
Transactions on Education	Journal	1
Advances in Artificial Intelligence	Book	1
New Generation Computing	Journal	1
Human-Computer Interaction	Journal	1

Table 3.3: Distribution of studies according to publication channels and occurrence

3.3.2 Research Methods

Table 3.4 provides the list of research methods used in the selected 24 primary studies.

Research method	Studies	Number	Percent
Single-case	[62–75]	14	58.3%
Multiple-case	[76]	1	4.2%
Survey	[59, 77, 78]	3	12.5%
Experiment	[61, 79–82]	5	20.8%
Benchmarking	[83]	1	4.2%

Table 3.4: Distribution of studies according to research methods reported

3.3.3 Methodological Quality

In order to conduct a methodologically rigorous review, we need to assess the methodological quality of primary studies. We dedicated our quality assessment elements for the evaluation of methodological quality. The issues pertaining to quality to be taken into account are: rigor, credibility, relevance and reporting quality. Out of 11 questions, we allocated corresponding screening criteria for all these issues.

Considering the reporting quality, most of the studies presented well-established aims, scope and context. Figure 3.3 shows the histogram of appraised values for reporting quality.

The assessment of relevance is a very important criterion in which we assess whether the primary studies are useful for software industry and research community. Satisfactorily, all of the studies revealed their relevance to embedded user assistance to the full extent. As it is shown in Figure 3.4, we had a good set of primary studies considering embedded user assistance as a major concern.

A likely paucity of relevant studies in the field was avoided. Rigor criterion is based on how clear and reliable are the reported methods, proposals and corresponding findings. Figure 3.5 shows the distribution of values for rigor quality.

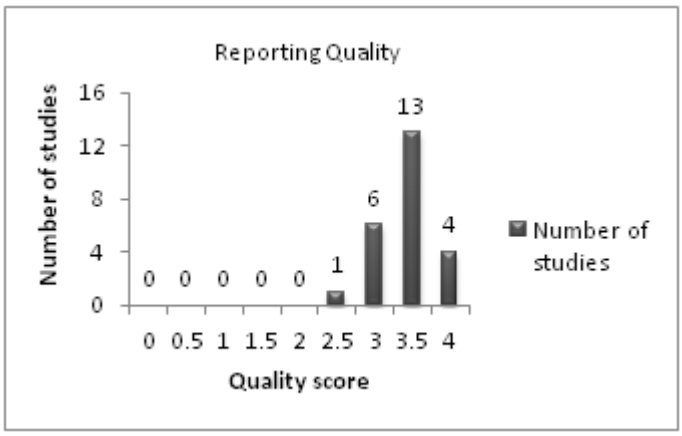

Figure 3.3: Reporting quality of the primary studies

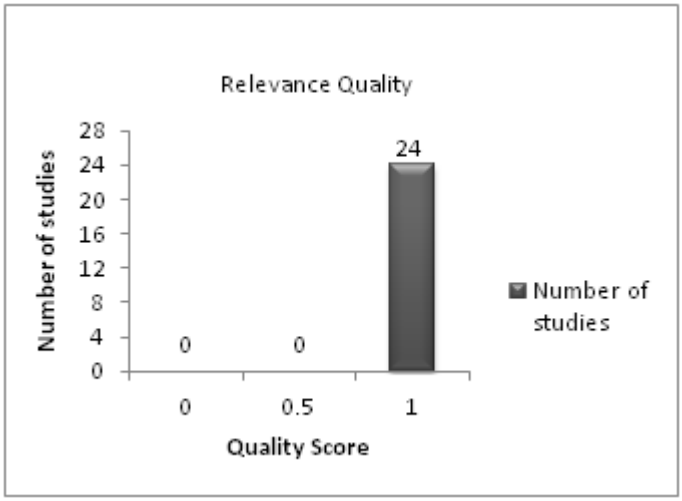

Figure 3.4: Relevance quality of the primary studies

We appraised no studies to get full scores in terms of rigor. Nine (38%) of the primary studies that got scores between 3 and 3.5 can be considered of good-quality according to rigor of stated findings. Among these studies, [75, 76, 79, 82] are of best quality in terms of rigor.

We should also assess the credibility of evidence that is the extent to which the findings and the major conclusions of the primary studies are profoundly clear, valid and suggestive. Figure 3.6 shows the histogram of quality scores based on credibility of evidence. Regrettably, we did not assess a primary study having full credibility of evidence. Seven of the studies having 1.5 score can be considered

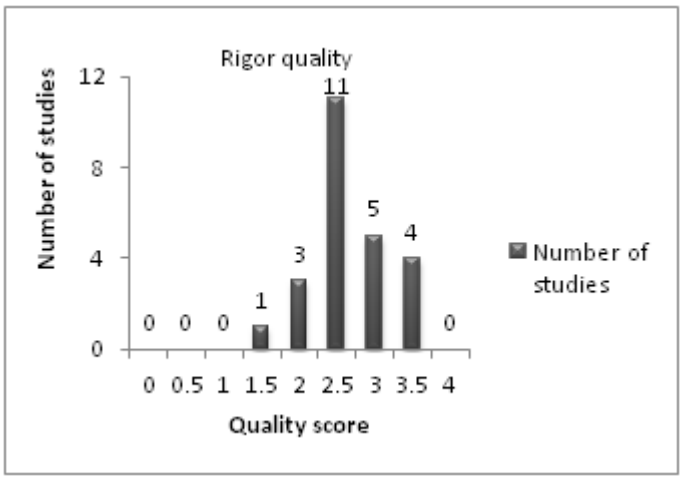

Figure 3.5: Rigor quality of the primary studies

as credible research, and twelve of the studies are of medium quality in terms of credibility of evidence.

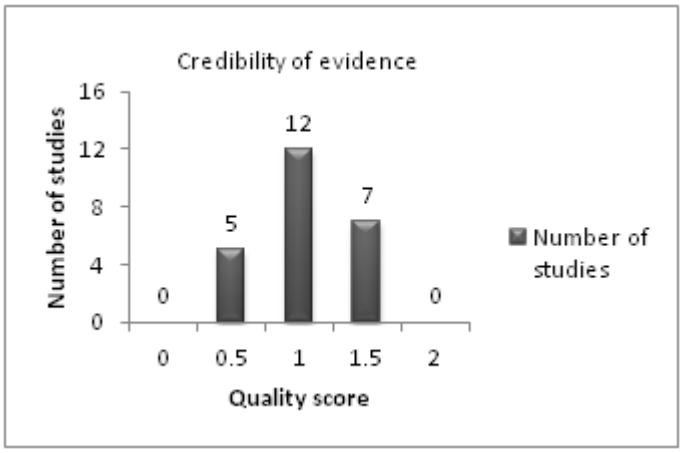

Figure 3.6: Credibility of evidence of the primary studies

Finally, Figure 3.7 shows the overall methodological quality scores out of 11. All of the criteria (i.e. reporting, relevance, rigor and credibility of evidence) were taken into account, and 17 of the studies (71%) having scores greater than 8 are relatively good. Two studies, [75, 76], are the top quality studies among the set of our primary studies. The studies having scores less than 8 are considered to be good-mediums or poor quality studies, and we, unfortunately, have 7 participants in this set. We can now derive a conclusion related to methodological quality that the set of primary studies are relatively good in terms of the four criteria specified

for this purpose.

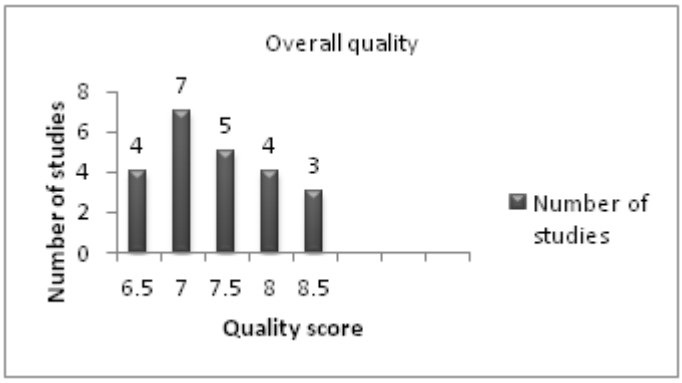

Figure 3.7: Overall quality of the primary studies

3.3.4 Systems Investigated

This section presents the results of our data extraction and corresponding data synthesis procedures applied to 24 primary studies. As was previously stated, we have four research questions, some having sub-sections, and the related data is presented under sections pertaining to each research question. If we recall our formulation of research questions, we can say that we followed a, so to say, top-down approach, meaning that the results will be presented from general to specific findings. This approach is a good way of revealing the current body of knowledge provided that the data extraction process is carried out with full respect to research questions, and by well-tabulating the corresponding information from primary studies.

RQ 1. In which areas of software-intensive systems have embedded user assistance techniques been applied?

Under this research question, we tried to classify the primary studies. The definitions of embedded user assistance, the target areas and the background of individual studies are the factors that we took into consideration. Table 3.5 shows the definitions and characterizations of embedded user assistance observed in primary studies. Actually, the entries in this table present the provision mechanism of embedded user assistance by slightly touching on the methods proposed.

Table 3.5: Characterization of embedded user assistance in primary studies

Study	Characterization of embedded user assistance
[80]	Context-awareness is the factor to build an assistive environment under which we can design good applications.
[67]	Monitoring a user's input actions and providing input-assistance actions by defining agents under assistance rules.
[64]	Helping users retrieve data from disparate sources in response to their needs and a defined criterion.
[73]	Providing a check mechanism for users to decide on the access rights under Internet pages or domains they create.
[77]	The concepts are embedded help appearing within the application, field labels, and page overviews.
[59]	Embedded user assistance refers to a built-in aspect of the user interface itself.
[78]	Direct access to help on the user needs, based on the current location, focus, and state within a software system.
[82]	Exploiting context for pro-active information recommendation based on the knowledge of user interests.
[79]	A solution that is based on improving user interfaces and providing built-in help.
[65]	An online documentation mechanism and context-sensitive help, implemented by technical solutions focusing on users' goals (user-centered design)
[71]	Agents for instruction assistance in Learning environments with rich interactivity and friendly communication interface
[62]	Being a fully recognized component of the user experience, which goes beyond the tools that manage static documentation
[68]	Offering efficient data display and ways to interact with the data by anticipating user needs or inferring from user behaviors at the interface level
[66]	Full text search supported by natural language query, the ability to support a tutorial or wizard style interface on task-oriented material and tutorials.
[74]	Mechanism to cope with semantic complexity: Semantic transparency yields embedded user assistance since semantic objects are acted on by users
[63]	Offering information needed by users for the completion of tasks right on the web page they are editing (i.e. task specific guidance on related concepts)
[72]	Assisting the user in the workflow in general and especially for applications that cannot be easily replaced, e.g. legacy applications
[61]	Enabling users, especially novice users, to use software tools effectively and assisting these users in accomplishing their tasks
[70]	Providing necessary suggestions by learning user profiles and actions where the profiles are designed to represent each user as well as the subject.
[69]	Adapting application behavior to changes in the configuration of the environment by means of using knowledge models and ad-hoc context models.
[76]	A special form of online help in which the documentation of the application resides within the application in a context-sensitive manner.
[75]	Monitoring users while they work on a graphical user interface (i.e. hypothesized intentions to prevent mistakes and to offer related advice).
[83]	Error messages appearing in pop-up windows with links to specific areas
[81]	Leveraging on-line information and user context by providing navigation assistance (i.e. intelligent human-machine interaction by assisting users

Six of the primary studies undertake the context of software-intensive systems in user interface level to provide embedded user assistance. Here, the idea is to incorporate some built-in mechanisms like avatars, additional interface components or some offering mechanisms within the graphical user interfaces of software applications. This approach is not limited to any specific type of software applications (e.g. desktop, web-based, navigation etc.) which is going to be demonstrated in detail in the following sections. Additionally, five of the studies propose that monitoring users actions in a graphical user interface is worth applying if we have well-defined set of tasks and related actions in hand. We can deploy our user assistance mechanisms on well-established and fine-grained models of user intentions that are inferred by knowledge and user models. The treatments in this set of proposals converge in a way that users goals and corresponding needs are processed by the instrumentality of related concepts in the field of software engineering.

Interestingly, the interface or window-manager levels of software applications are not enhanced with only contextual elements. There are some proposals including input-assistance functions like the studies [67,75] in which the authors discuss better user experience by reducing possible input mistakes in the workflows. This approach is enhanced by [71] where the authors go beyond the boundaries of check mechanisms by introducing agent technology. Again, it is stated in this primary study that embedded user assistance is all about rich interactivity and friendly communication interface which can be ensured by means of additional components (i.e. *agents*) that are on the same level of applications they belong to.

All in all, the proposals towards embedded user assistance first discuss the adaptation to user activities on behalf of application context. The documentation pertaining to software applications are to be resided inside the actual context as proposed in [76]. In other words, user assistance is not a minor concern to be attached to concerned parts of applications through mechanisms that usually turn out to be incapacitated in some specific parts of user work-flows. The studies [61,62] are based on the usage of software tools, and they try to exploit dedicated components rather than the tools that manage static documentation.

Having analyzed the definitions used for embedded user assistance, we can now identify the domains that the primary studies aim at and associated specific target areas. For this purpose, we categorized the studies into six main groups. In order to shed some light on this question, we first investigated the targeted domains of the individual primary studies to hold a general view. The six main groups have their sub-categories under which various types of proposals reside. Table 3.6 shows the categories of the target domains that we discovered. The categorization is to be done in an objective manner so as to reveal the actual state of current body of knowledge. Therefore, we made use of descriptive and qualitative synthesis as proposed in [20]. The extracted information (i.e. target domains) was tabulated in a systematic way to observe the similarities and differences between primary studies. It is obvious that the information pertaining to this specific question is of qualitative nature, meaning that natural language outcomes are the inputs for analysis. As was previously stated, we employed reciprocal translation and line of argument analysis.

Web-based systems have been targeted by several studies in different fashions. Together with some proposals presenting agent technology, pro-active information recommendation, anticipating users needs and task specific guidance are the major concerns taken into account within these types of studies. The connotative meaning here is that we can exploit the current state or any kind of location in software applications to integrate and embed related user assistance documentation just-in-time.

The category of enterprise systems has also received great attention by academics and industry. In this category, we can observe that the provision of user assistance is dedicated to some critical concerns such as security and real-time execution. Additionally, the study [72] discusses how we can provide embedded user assistance for the applications that have long been used (i.e. *legacy systems*). This case is very important because the technical structure of a legacy system is not always easy to handle and maintain; and thus, embedded user assistance is a challenging requirement for these kinds of systems.

Educational environments fall into a distinct category among the primary

Table 3.6: Target domains and specific areas stated in the primary studies

Target domains	**Specific areas**	**Studies**
Web-based systems	Education systems	[73]
	Applications	[68, 77]
	Forms	[67, 78]
	Browsers	[82]
	Portals	[63]
Enterprise systems	Internet security	[62]
	Military applications	[64, 76]
	Professional applications	[76, 84]
	Legacy systems	[72]
Learning environments	Digital libraries	[70]
	E-learning systems	[71, 73]
Information-retrieval systems	Network centric operations	[64]
	Military data systems	[64]
	Information-driven design	[79]
	Document management	[74]
Special User Assistance Systems	Open-source systems	[65]
	General scope	[59, 61, 75]
Real-time systems	Mobile devices	[81]
	Assistive environments	[80]
	Dynamic environments	[69]
	Speech-enabled systems	[66]

studies where Web is mostly the medium of communication. Specifically, the studies [71,73] discuss instruction assistance and supervision issues under learning environments. Again, high interactivity of user interfaces by learning user profiles and actions performed is one of the major concerns in these kinds of systems.

The category of information retrieval systems consists of a wide variety of specific areas for which we have one representative per group. Semantic complexity is one of the most important issues that we should deprive its negative effects imposed to information retrieval. In these systems, users perform conventional tasks such as searching and browsing documents that could be enhanced by the mediation of some built-in help mechanisms.

Some primary studies set their sights on the development of stand-alone user

assistance systems that interact with the applications they belong to in an embedded manner. We categorized the studies that explore the provision of embedded user assistance in general terms under the specific category. These proposals discuss the issues and related problems to be solved in the field of user assistance which we will demonstrate in the following research questions. Also, we have somewhat outlying categories that have very valuable contents and impacts on embedded user assistance, and we will analyze these proposals individually.

The studies [69, 80] focuses on the issue that is adapting application behavior to changes in assistive and dynamic environments. Moreover, the use of speech content to form an embedded user assistance system is proposed in study [66].

We also analyzed the actual backgrounds of primary studies, in other words the types of cases that the authors conducted. Figure 3.8 shows the distribution of primary studies according to research backgrounds. Here, we appraised a study as empirically-based if the authors actually evaluate a model, method or algorithm for embedded user assistance. These studies mostly propose a method, framework or tool for the provision of embedded user assistance. Otherwise, the cases, where the conclusions were drawn from empirical data despite no actual empirical evaluation, are categorized as empirically based. Also, we have two studies in total having unclear or poorly-reported evaluation approaches. Actually, these studies are the survey-like studies since we did not specifically aim at the papers including empirical evidence. Rather, we included both qualitative and quantitative studies without imposing any restrictions on the study designs.

RQ 2. What are the motivations behind embedded user assistance?

The purpose of this research question is to thoroughly examine the motivations behind the use of embedded user assistance and the concerns leading the adoption of embedded user assistance. We categorized the motivations of studies including a diverse range of approaches. We have seen that embedded user assistance technologies are applicable to various software-intensive systems.

In the category of web-based systems, the motivations are of somewhat similar nature. The study [73] prompts the motivation that new tools have been required

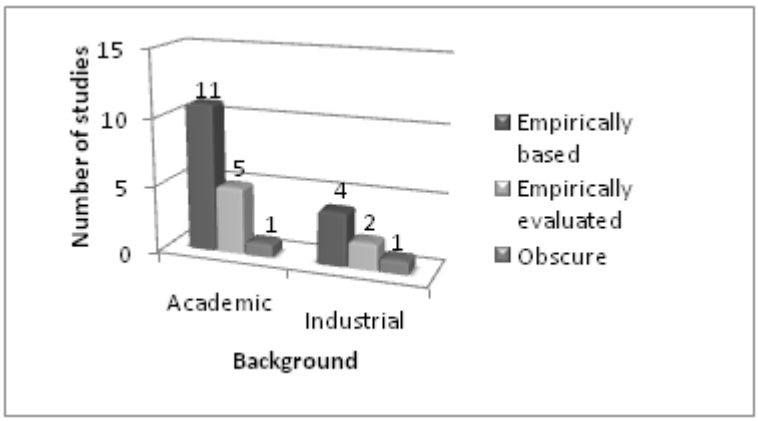

Figure 3.8: Research backgrounds of the primary studies

to provide a well-structured access to the Internet. Providing intelligent advice on potential sites, considering user activity, or providing content-specific filtering of Web pages are the issues to be taken into consideration. For example, the users of an educational system can lose concentration and navigate to unrelated websites. This problem could be ironed out by utilizing embedded technologies that monitor such situations. The best of possible solutions is to employ a client-based system which is easier to use, to implement and to reuse.

The author of [77] states that an external help system can be efficiently used to provide in-depth conceptual information and procedures. However, he also mentions that external help systems do not suffice to meet users needs because of being passive and external resource. From the users point of view, embedded user assistance is worth applying, which brings practically no effort. The most important argument here is that the users of an application cannot differentiate embedded user assistance from the actual application it belongs to.

We can move our potential users to use help mechanisms by embedding the documentation into the application level. The authors of [68] discuss semantic Web applications in which uniform data display, irrelevant display of data infrastructure and uneasiness of use are the cons to be handled. They underlined the requirement which is to clarify user tasks by knowledge transmission based on information communication. This is the reminiscent of an embedded user assistance system to present related documentation according to users expectations.

The authors of [67] bring out an evident fact that users have difficulties in filling out complicated Web forms while applying for some Web-services. This situation interrupts the users workflows; consequently, they look for other providers having more user-friendly services. In this case, enabling users to complete their inputs by themselves is to be accomplished by embedded assistance mechanisms. The resultant mechanism is a type of Web-based self-service that have recently received considerable attention. It is also stated in [78] that guidance with context is more effective than guidance without context, which has been substantially proved by the research so far.

One obstacle to contextual guidance is presented in [82] as the lacking of knowledge about active user goals. This shortage would prevent profiles from supporting pro-active, context-aware retrieval in which relevant documents are automatically presented to users according to their activities. Again, embedded mechanisms are in demand to get hold of the user goals. The study [21] explains the downsides of an HTML-based help system in which a users focus is disturbed by dedicated help pages, causing the user to get lost in the application they are currently working on.

The world of enterprise systems mostly focus on the fulfillment of non-functional concerns such as security, reusability and cost-effectiveness by the mediation of embedded guidance mechanisms. The study [17] proposes a design language, which has gained popularity with user-interface designers, that deals with user context and system interactions more directly than the user-interface designers. The studies [76, 83], where the latter states 34% of all IT projects fail just because of user adoption, discuss the difficult localization of user assistance concerns which reduces the modularity and maintainability of software systems. Also, reuse of help mechanisms is encouraged among software applications, which reduces the time of development. The study [72] proposes the development of smart environments which will be designed to offer transparent user assistance by decoupling users from computing devices. This special feature is considerably desired in case of legacy applications.

Learning environments, which are discussed in [70, 71, 73] as mostly of

computer-aided instruction, are motivated on the development of new tools to provide structured, focused, and controlled access to the Internet facilities. Considering various user communities and their personal requirements, a more enriched, blended interactivity without the limitation of space and time is to be established through embedded user assistance.

Both academic and industrial world have paid great attention on the improvements on information-retrieval systems. It is stated in study [64] that the time, location and past history of information access can also shape possible user requirements and user queries, which indirectly leads to a basis of embedded user assistance. The study [74] focuses on the semantic complexity of documents which can be solved by proper guidance.

Grayling [79] explains that users behavior appears to be far from rational due to two paradoxes: production paradox and assimilation paradox. The former is based on the high motivation of users to do their task, and paradoxically, the unwillingness to learn provided assistance mechanism that would enable them to do their task effectively and efficiently. The latter paradox is about the ad-hoc behavior of users that is bringing experience of other similar software to using software in front, where this behavior hinders more than it helps, paradoxically.

30% of the primary studies are motivated on the development of somewhat stand-alone systems of embedded user assistance, which have rarely been implemented so far. For this purpose, the study [65] proposes the translation of users goals into genuine requirements and related technical implementations as far as possible. The study [61] sheds some light on the persistent problem that why existing help systems are not enough to guide users in accomplishing their tasks. In a similar fashion, the weaknesses of user manuals, hypertext facilities and tutorials are revealed in [75].

Our last category is of outlying nature in a sense that the studies do not fall into the other categories specifically. The study [80] is intended for predicting or anticipating a future situation by learning techniques in order to be applied to assistive environment. The study [66] affirms the decision of incorporating an embedded window to display context-sensitive assistance within an application,

but due to limited spaces of display, speech technology is proposed in user assistance scenarios involving mainstream desktop/laptop software applications. It is also stated in [69] that an instrumented environment and appertaining assistance services should be offered to the users, depending on the current context of software applications. Interestingly, the authors of [85] offer the incorporation of embedded assistance contents into cell phones for the sake of personal safety in real world situations.

We can reveal the concerns leading the adoption of embedded user assistance in a big-picture view as shown in Table 3.7.

Concern	**Studies**
Usability	[59, 61, 65, 66, 68, 70, 73, 76, 81, 83]
Context-awareness	[62, 64, 76, 78, 82]
Non-obstructiveness	[59, 63, 79]
Reuse	[63, 73, 76]
Accessability	[70, 73]
Costs & development time	[67, 83]
Customization	[65, 77]
Extensibility	[65, 76]
Localization	[76, 77]
Personalization	[68, 77]
Reducing training time	[76, 83]
Auditory assistance	[66]
Easier implementation	[69]
Intelligence	[75]
Interactivity	[71]
Loose-coupling	[72]
Manageability	[68]
Modularity	[76]
Optimization	[81]
Prediction	[80]
Pro-activeness	[82]
Safety	[81]
Semantic complexity	[74]

Table 3.7: Major concerns for the adoption of embedded user assistance

It is obvious that usability concern, in general, has attracted great interest from various kinds of primary studies. These studies are in favor of user-centered

design in software-intensive systems. The second concern is the context-awareness within software applications. These studies focus on exploiting context of target applications in order to provide more effective embedded user assistance. Non-obstructiveness and reuse of user assistance across applications have also been considered from a relatively wide range of primary studies.

We tried to identify each and every concern lying behind the primary studies to express the motivations in a more detailed way. It is better to perform a, so to say, concern-domain mapping among the primary studies in order to expose a more explicit view of current evidence. The previously specified six categories have been interested in several types of requirements, leading them to fulfill different concerns at the same time. Taking Table 3.7 into account, unsurprisingly, the most popular concerns are revealed from a different point of view. Considering the six categories of domains and twenty three concerns identified after a comprehensive review, we got a distinctly distributed set of combinations.

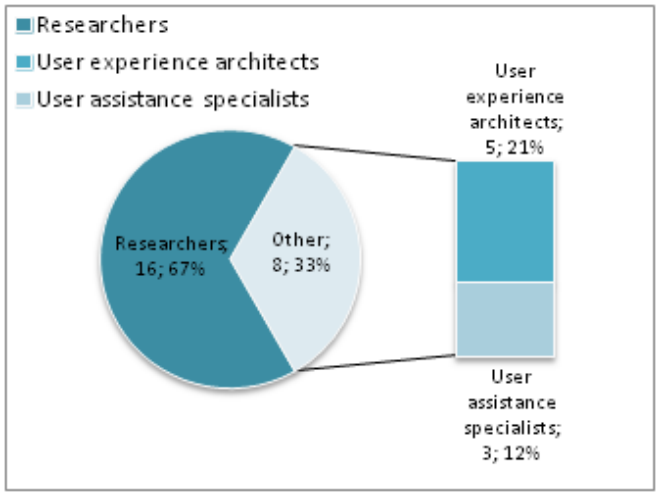

Figure 3.9: Subject of investigation in the primary studies

RQ3. What is the state-of-the-art on embedded user assistance?

Under this research question, we tried to figure out the extent to which the published literatures on embedded user assistance have reached so far. For this purpose, we first analyzed the subjects of investigation conducting the primary studies. After, we can demonstrate the employed techniques to provide user

assistance along with the objectives and research directions. The types of subjects are also important to arrive at an ultimate decision on the empirical evidence reported so far. Figure 3.9 shows the distribution of individuals who performed the primary studies.

We have a considerable amount of studies performed by user assistance professionals. 67% of the primary studies were performed by researchers, generally dealing with human-computer interaction, interface design and user-centered design of software-intensive systems. Having seen the subjects, we identified the specific techniques used for embedded user assistance. We also analyzed the research directions behind these proposed techniques which can either contain theoretical or applied set of methods.

RQ3.1. What are the techniques applied?

While answering this research question, we again considered the categories of domain under RQ1. Since the domain acquire different characters at some point of analysis, it is better to reveal the set of techniques independently for the sake of a more effective data synthesis.

Firstly, the primary studies towards web-based systems have applied specific techniques according to the nature of existing phenomenon. Being a web-based education system, the study [73] features an intelligent document classification system. Specifically, the users, teachers in this case, are allowed to train the system in order to form a basis for further classification. This is done by ranking a set of Web pages with a relevancy rating for a subject. In case of any requests, the contents that student want to access were extracted and compared with the rating pattern, which provided a rating of the similarity between the requested page and the relevant pages. This approach provides a good way of embedded user assistance for the users of browser-based systems.

The study [67] proposes a mechanism for developing an assistant agent which is a software module for monitoring a user's input operations and providing input-assistance functions. The main idea is to model the interaction between a user and a web form, and to design assistance rules, which can be interpreted at

runtime or at form generation time, according to the model by events, conditions and actions. The major advantage of this approach is that the developers can implement assistant agents without any programming by defining assistance rules. The actual behavior of assistant agents is totally defined by assistance rules.

The author of study [77] mentions about the use of overviews, field labels, popup windows and dedicated user assistance panels. The study [78] raises the matter that context in a software system can be considered in different ways. For example, this can be previous history, user roles, current fields or controls, and other settings associated with the use of a software system. In this study, options for displaying context-sensitive help are listed as: inline help, expanding/drop-down text, dedicated embedded windows and pop-up windows. Going further, the study [82] proposes a novel approach to derive semantically enhanced contexts that reveal the information related to user profiles and the current browsing activities. This method can be abstracted in a way that agents act as browsing assistants who monitor web browsing to learn user preferences and provide real-time recommendations. Exploiting hierarchical representations of user interests obtained by conceptual clustering is the key of this technique used for detecting regularities and patterns of behavior regarding such interests. In a similar fashion, the study [68] proposes a user model that represents the characteristics of the user which will be further used for creating content and presentations adapted to different users. This method of information selection filtered for the user facilitates navigation and increases the performance of web-based systems.

One of the most inventive approaches is the use of an avatar on web pages. The study [63] proposes such a technique in which users can drag the avatar, in other words the user guide, to proper locations, preventing it to interfere other elements on user interfaces. The avatar is said to be showing no idle animations within the interfaces for the sake of performance. This mechanism can provide embedded user assistance such as element explanation, input verification, step-by-step guidance and continuative information.

In the field of enterprise systems, the study [62] features a pattern language as a guide to design. This approach is defined as a collection of patterns, each of

which describes a relationship between contexts, recurring forces within contexts and spatial configurations to resolve these forces. By this way, it enables an application or family of applications to reveal a uniform feeling of user interactions. Moreover, the study [72] defines a generic architecture, which is theoretically applicable to a wide range of smart environments and applications. Here, the authors also define a work-flow, which ensures a tunable generic architecture for specific use cases. The proposed mechanism stands in between the smart environment and the legacy application to which an extension of the user interface is provided to integrate additional functionalities. Also, for controlling the environment, some commands are provided. A loose coupling of the smart environment and the application is achieved by adaptive interfaces, rules engine and context storage with importers.

The study [64] focuses on user profiles, process workflows, source availability monitoring, and real-time semantic integration. The authors propose the combination of approaches to provide embedded user assistance such as: workflow, context, profile, ontology, and information integration and aggregation. The study [76] undertakes the topic of embedded user assistance as a completely separate concern in which a model for process-sensitive user assistance is defined along with process definition and help definition tools. According to the authors of [83], error messages appearing in pop-up windows with links to highlighted specific areas to correct errors are the kinds of embedded user assistance. When the users are in need of help, they browse relevant help topics available at different levels based on the current state of application.

In learning environments, the study [71] proposes an architecture comprising of knowledge retrieval module, log recording module, learning module, and user interface. Similarly, the study [70] features a structure of components such as phrase-extraction agent, domain-specific agents, concept dictionary, user-interface agent, blackboard and dispatcher.

Since the information-retrieval systems are generally based on an interactive structure, the study [79] describes several types of embedded help like wizards, audio cues and video. The idea again is to make the embedded help an integral

part of the user interfaces. The study [74] proposes the use of framing, specifically content frames, that enable content variant as a form of semantic interaction.

If we want to develop a specific embedded user assistance system, as study [65] proposes at every step of the design, common Web technologies like DHTML, XML and JSPs are to be used in order to present user assistance documentation as a normal Web application. Further, a cognitive theory, which contains methods of human plausible reasoning and goal-recognition based on the user interactions, can be exploited just like in [75] for a stand-alone user assistance system working in an embedded manner.

In the domain of real-time systems, the individuals feature some interesting technologies to provide embedded user assistance. The study [80] focuses on context prediction for embedded user assistance which is based on predicting user activity, Markov decision process and uncertainty and fuzzy-state Q-learning. The authors of [66] developed a user assistance system using speech technology to provide context-sensitive help. The study [69] features an infrastructure to link factual context knowledge and procedural context knowledge in the form of rule sets to all relevant items. In mobile devices, the authors of [81] propose to rank users demand, prioritize information and applications, advise against the use of some applications. In this way, the presentation of non-essential information is delayed for the sake of safety.

RQ3.2. What are the research directions followed?

This question is intended for discovering the research directions stated in the primary studies. We observed both the objectives of individual proposals in a long-term fashion to determine the most popular intentions towards embedded user assistance. Having discussed the previous questions that are well-rounded, now we should identify the actual state-of-the-art in this field by analyzing the statements that are possibly reminiscent of further research and practice.

We identified 16 distinct research directions among the primary studies, and it should be stated that some of them possibly converge in a sense. The distribution of research directions according to primary studies and related concerns is shown

in Table 3.8. The corresponding percentages with respect to study coverage and concern coverage are also given.

Firstly, agent-based technologies have attracted a considerable interest by resolving several issues pertaining to the concerns reported. 17% of the primary studies underlay agents to provide embedded user assistance. Also, the number of corresponding concerns that are to be fulfilled by these technologies is quite high, which are: accessibility, context-awareness, intelligence, interactivity, proactiveness and usability.

The remaining set of primary studies state various kinds of research directions. Firstly, the use of an application-independent framework is a noteworthy research direction covering 35% of individual concerns. Moreover, browser-based embedding, semantic technologies and user-centered design are the other possible initiators of embedded user assistance. We should also note the importance of process-sensitive help which covers numerous kinds of concerns like: usability, context-awareness, reuse, extensibility, localization, and easier training.

Embedded authoring tools are mentioned, which will possibly iron out the difficulties of embedded user assistance, especially in design and development phases. It should be noted that existing authoring tools do not suffice for the essential issues that are to be considered by user assistance professionals. Thus, some dedicated tools are required to ease the development of embedded user assistance.

It is also discovered that we should follow some interaction adaptation strategies within real-time systems to provide necessary guidance required.

Table 3.8: Research directions and related concerns

Research direction	**Primary studies**	**Related Concerns**	**Concern percent**	**Study percent**
Agent-based technologies	[8] [11] [19] [22]	Accessibility,Context-awareness,Intelligence,Interactivity, Pro-activeness, Usability	**26%**	**17%**
Browser-based embedding	[5] [9]	Customization,Localization,Non-obstructiveness,Personalization	17%	8%
Application-independent frameworks	[20] [21]	Context-awareness,Easier implementation,Easier training,Extensibility, Localization, Modularity, Reuse, Usability	**35%**	8%
Semantic technologies	[13] [15]	Usability, Personalization, Manageability, Semantic complexity	17%	8%
User-centered design	[7], [10]	Usability, Context-awareness, Customization, Extensibility	17%	8%
Avatar-based technologies	[16]	Non-obstructiveness,Reuse	9%	4%
Dissolution of discrepancies	[18]	Usability	4%	4%
Embedded authoring tools	[4]	Usability,Reuse,Accessibility	13%	4%
Input-assistance functions	[2]	Costs & development time	4%	4%
Interaction adaptation strategies	[24]	Usability,Optimization,Safety	13%	4%
Middleware technologies	[17]	Loose-coupling	4%	4%
Ontology-driven technologies	[3]	Context-awareness	4%	4%
Pattern language	[12]	Context-awareness	4%	4%
Process-sensitive help	[21]	Usability, Context-awareness, Reuse, Extensibility, Localization, Easier training	**26%**	4%
Task-based technologies	[6]	Usability,Non-obstructiveness,	9%	4%
Speech technologies	[14]	Usability,Auditory assistance	9%	4%

RQ 3. What is the strength of evidence in support of the stated findings?

The plausibility of the reported results and findings for the readers of this review is very important. The overall strength of the body of evidence based on the reviewed studies is to be revealed under this research question. In order to grade the strength of evidence, there have been many proposals and systems so far. We used the definitions from *GRADE (Grading of Recommendations Assessment, Development and Evaluation)* [58] working group as in Table 3.9 in order to avoid doubts imposed by the weaknesses of evidence hierarchies.

Table 3.9: Definitions used for grading the strength of evidence

Grade	**Definition**
High	Further research is very unlikely to change our confidence in the estimate of effect
Moderate	Further research is likely to have an important impact on our confidence in the estimate of effect and may change the estimate
Low	Further research is very likely to have an important impact on our confidence in the estimate of effect and is likely to change the estimate
Very low	Any estimate of effect is very uncertain.

In order to determine the strength of evidence, we should consider four elements that are: study design, study quality, consistency and directness. Concerning study design, the GRADE system assigns higher grade to experiments than to observational studies. Five (21%) primary studies are of experimental type in the review. Also, Table 3.10 shows the descriptive statistics for the quality scores corresponding to this group of studies. As a result, our first categorization about the strength of evidence based on study design is *low* due to having fewer amounts of experimental studies despite having high quality scores.

Considering the quality of the studies, the issues of bias, validity and reliability are not discussed explicitly in most of the primary studies. Besides, only two of the studies got full score in our last quality assessment criterion that is the

Table 3.10: Descriptive statistics for the quality scores of experimental studies

Criterion	Values
Experimental studies	[1], [8], [9], [18], [24]
Number of studies	5
Mean quality score	8,2
Standard error	0,538516481
Standard deviation	1,204159458
Minimum	6,5
Maximum	9,5

statement of limitations where fourteen of them discussed to some extent and the remaining did not state any limitations at all. Only eight studies (33%) stated the findings that relate to the aims of research with credible results and justified conclusions. Hence, there is an inevitable risk of bias due to the low quality scores in terms of the study quality, requiring a great deal of circumspection.

With respect to consistency which refers to the similarity of estimates of effects across studies, we realized slight differences among articles. Since we categorized the primary studies into six main groups, we can only talk about specific groups evidences. We did not perform a sensitivity analysis by excluding poor quality studies or studies of a particular type because of the varying presentation of both objectives and empirical results. Besides, the synthesis of quantitative results was not feasible because the outcomes of primary studies were not presented in a comparable way. In other words, the reporting protocols differ from study to study, causing us to perform the data synthesis in a more qualitative or descriptive way which was not solely desired. Consequently, we arrived at a conclusion that the results are barely consistent owing to the large extent of imprecise evidence.

With respect to directness which is defined in as the extent to which the people, interventions, and outcome measures are similar to those of interest, a narrow range of the studies reported comparisons of interventions. This is most probably because of the wide range of solution domains in the field of embedded user assistance. At the very beginning, the intervention in this review was composed of general, functional, procedural and conceptual types of embedded user assistance techniques, and we did not impose any restrictions on outcomes by foreseeing the

plenty of factors of importance to both researchers and practitioners. Regarding the directness of evidence for each category of primary studies, most of the studies were performed by, so to say, the pundits of embedded user assistance. The subjects were mostly academics, user assistance professionals and user experience architects. Thus, the total evidence based on directness of the primary studies is *moderate*.

3.4 Discussion and Concluding Remarks

This systematic review is based on the results 550 papers that address the introduction and adoption of embedded user assistance. We have analyzed 24 papers as our primary studies and presented the results of them thoroughly. The benefits of embedded user assistance and the major concerns for the adoption of it are stated within these papers, and we systematically presented them. We have grouped the papers to better reveal the state of these systems in a categorical way and as such the readers can make inferences on these results easily according to their field of interest. We have employed a *quasi-gold standard* (QGS) that enhanced our search strategy. Also the inclusion/exclusion criteria have been applied on a systematic screening of discovered papers. Collectively assessing the four elements about the strength of evidence, the analysis of the impact of embedded user assistance solutions resulted in a *low* grade. Also, considering our research questions, qualitative studies were more appropriate, in a sense, than quantitative ones for revealing more types of solutions and deciding on the technologies to employ. In other words, the likely paucity of empirical studies in the specific fields of embedded user assistance results in encompassing as many types of studies as possible, which we settled as a strategy in this review.

Finally, we can now proceed with our tool framework *Assistant-Pro* by evaluating and enhancing its features based on the results of systematic reviews. In this way, the framework will be built on well grounded theories and methods that have been discovered comprehensively.

Chapter 4

Survey of Help Authoring Tools

This chapter focuses on our survey of *Help Authoring Tools*. Section 4.1 presents an introduction and background of *Help Authoring Tools*. Section 4.2 details the background of *Help Authoring Tools*, focusing on the considerations towards them. Section 4.3 summarizes the results of our inclusive survey with some well-justified decisions.

4.1 Overview

From past to present, the provision of user assistance in software-intensive systems has been evolved. Formerly, printed documentation was commonly the means of user assistance. Later on, the creation of help files became automated using *Help Authoring Tools*. *A Help Authoring Tool(HAT)* is used by technical writers and software authors to keep track of help files, manuals and documentations [86]. In other words, a *Help Authoring Tool* or *HAT* is a software program which is used for creating sophisticated help systems.

Help Authoring Tools are generally of single self-contained applications to ease the process of designing, creating and maintaining help files and documentation

materials by means of dedicated authoring features. By using them, the individuals are taken out of the complexity (i.e. coding, scripting, conditional tags) of actual systems by focusing only on the most complete and accurate help and documentation manuals. However, reusability and content management facilities are limited in *Help Authoring Tools* compared to advanced content management systems. Besides, the standardization of the use of specific *Help Authoring Tools* has not been established so far, which is eventually a detriment to flexibility [87].

Making use of *Help Authoring Tools*, the corresponding outputs like help documentation and manuals are often used to provide the following as stated in [86]:

- User guidance for a software product
- Training the use of a software product
- Educational course support
- Flourishing a a software product for advertisement purposes

As explained in the article [88], there are some common features of *Help Authoring Tools* like:

Single sourcing is a publishing principle that refers to writing help content once. We create one shared text source of all subsequent help authoring activities since we will have different output formats such as WinHelp, DOC and PDF.

Multiple output formats are supported by almost all Help Authoring Tools. There are some well-known formats as the following:

- CHM (Microsoft HTML Help) is the most widespread help file format nowadays. It is mainly based on HTML, and we can seamlessly format the help content into different presentation styles.
- PDF is another popular format, but any help content of this format cannot be weaved into applications in an embedded way. It is more

suitable for offline documentation purposes which contradicts context-sensitive user assistance.

- WebHelp is commonly used for search engine optimization in which the maintenance of help content can be carried out easily. Similar to Microsoft HTML Help, this format can be well-adapted for authoring online help content. One of the distinguishing characteristics of WebHelp is that the organization of HTML files is of a more automated fashion compared to Microsoft HTML Help.
- HLP (WinHelp) is used to be employed in older versions of Windows, but it is still being put into practice by the systems that are advanced in years.
- RTF (Rich Text Format) is a quite portable help file format in which all the formatting information about the help content is stored. Almost all word processors can recognize this format in order to unfold the containing help documentation.
- DOC format is less widely compatible than RTF, but it can include more text formating information.

WYSIWYG(What You See Is What You Get) Editor is provided by a considerable amount of *Help Authoring Tools* in order to ensure a comfortable authoring experience. The associated user interfaces mostly include well-integrated and smooth constructs that support full control over the help content.

4.2 Key Considerations of Help Authoring Tools

Before specifying our selected *Help Authoring Tools* and the evaluation criteria for this survey, analyzing the existing suggestions on this topic will be beneficial. There have been several evaluations of *Help Authoring Tools* on different contexts since they have started to attract great interest from the community. The basic

idea is to classify the tools with respect to their built-in features and to propose the measures of valuation accordingly.

4.2.1 A Sample Classification of Help Authoring Tools

We have analyzed some evaluation mechanisms, like checklists, that have already been proposed in order to select *Help Authoring Tools*. In the article [89], which was published by *indoition* (an exemplary partner for technical documentation know-how and tools), the major characteristics of *Help Authoring Tools* are nicely summarized. Here, the tools are first categorized according to their *editor* constituent being either external or internal. The categorization resulted in three major groups that are summarized as follows:

HATs with a built-in editor This group of *Help Authoring Tools* provide sophisticated *WYSIWYG Editors* in which the help creation and the integration with *meta-information* (i.e. the actual way of provision of help) are on the same level. On the contrary to dedicated text processors, these tools handle help texts and corresponding *meta-information* in a fully integrated manner.

HATs using an external editor These tools make use of external text processors such as Microsoft Word, HTML editors and XML editors. Here, the flexibility offered is much higher because the user can utilize the editor with his or her own preference. On the other hand, the margin of error is regrettably higher due to some possible variances in the translation process.

Mere converters These tools usually set up their operation on the basis of a formatting template. The formation of help texts and the associated specification on how they will be attached are two separated activities. Offline user manuals promote *single source publishing*; however, the usage of same source is rather detrimental to reusability across different product versions [89].

Having discussed the process of text translation, a good example of HATs should provide extensive translation features to secure a faultless text integration with the corresponding meta-information. Also, there are some other issues that are to be taken into account while selecting an appropriate HAT, and we can list them, based on [89], as follows:

- Version control on documents
- Simultaneous editing or team support
- Review process
- Formation of help content
- Automated generation of help documentation
- Source code documentation
- Maintenance on the help format
- Navigation features
- Context-sensitivity
- Working with legacy data
- Extra utilities
- Provision of extra software utilities

There is an inclusive survey [90] about the market overview of *Help Authoring Tools*. Here, almost all of the existing tools are evaluated and based upon some categorical headings. Table 4.2.1 shows the distribution of tools surveyed in the article [90].

Table 4.1: Categories of Help Authoring Tools

Main Category	**Help Authoring Tool**	**Proprietary**
Tools with built-in WYSIWYG editors	Flare	[91]
	Help & Manual	[92]
	RoboHelp	[93]
	HelpStudio	[94]
	Fast-Help	[95]
	HelpSmith	[96]
	Author-IT	[97]
Tools that integrate with Microsoft Word	Doc-To-Help	[98]
	Help Producer	[99]
	OfficeHelp	[100]
Tools focusing on source code	West Wind HTML Help Builder	[101]
	Doc-O-Matic	[102]
	Document! X	[103]
	TeeGofer Help Author	[104]
	VSdocman, VBdocman	[105]
	GenHelp	[106]
Tools for Linux and Mac OS	QuickHelp	[107]
	HelpLogic	[108]
	HelpBlocks	[109]
	Helen	[110]
Rapid help development tools	ScreenSteps Desktop	[111]
	HelpBurner	[112]
	Help Generator	[113]
	Dr. Explain	[114]
	TechWriter	[115]
Web-based help authoring tools	HelpConsole	[116]
	HelpIQ	[117]
	HelpServer	[118]
	exxDoc	[119]
	tomeCMS	[120]
Low-cost help authoring tools	HelpMaker	[121]
	HelpSetMaker	[122]
	HelpNDoc	[123]

We can observe, by means of the above table, how the specific approaches used in *Help Authoring Tools* are scattered. Several categories have been discovered, which can lead us to choose some representatives among them for our survey.

4.2.2 Factors Affecting the Choice of a Help Authoring Tool

As we have revealed the actual categories of existing *Help Authoring Tools*, we should now envisage some factors to be taken into account while choosing a *Help Authoring Tool*. Char James-Tanny, who was first named a *Microsoft Help MVP* in 2002 and is now providing comprehensive training and consulting services as a HAT expert, proposed several factors for this purpose in her seminal presentation [124]. We can list the factors with some explanatory statements as follows [125, 126]:

Users' environment is to be considered in a way that the users' work environment is well-established (i.e. Windows, Linux, MacOS, Unix etc.). Thereby, we can arrive at a justified decision for the choice. Since there can be, for example, Windows-based and Web-based applications requiring user assistance, we should generate proper help outputs, correspondingly.

Development environment The choice of text editor to maintain help content is also important, the help authors generally side with the editors in which they have extensive knowledge and proficiency. Microsoft Word and Adobe FrameMaker are some examples of widely-used editors.

Output The choice of output basically depends on users' environment. WinHelp, HTML Help, Java Help and Oracle Help formats are commonly used among HATs.

Other considerations The combination of different tools might be required, and this should be considered important while utilizing several HATs.

Output features These features are offered by HATs in which the help content is to follow some standard format such as Valid HTML and Natural Language Query.

Other features Some of the features that definitely affect the choice are: context-sensitive help, built-in dynamic help, content reuse, multi-authoring support and translation support. Technical writers or other concerned authors should pay great attention on the capabilities of HATs they might choose in terms of these valuable features.

Cost The power or functionality of a HAT may not be indicated by its price. Some inexpensive tools may provide extensive features compared to, so to say, high-ticket ones. We should first consider the features we would like to integrate into our help solution, and decide on one of the HATs that support them.

Support The kind of support is mostly *after-sales services*, and naturally the actual price of a HAT and its support costs are directly proportional.

Learning curve This factor is somewhat relative as it totally depends on the flexibility and user-friendliness provided by HATs.

Training The easier to learn a HAT, the less it will require training. We will discuss these factors also in our own tool framework in the following chapter.

4.3 Survey of Selected Help Authoring Tools

After a solid background information about *Help Authoring Tools*, in this section we select and survey some of the tools previously mentioned. Firstly, we analyzed Table 4.2.1 for different categories of tools and the HAT Comparison Matrix [14] that has already been proposed to decide on the tools for analysis. In the comparison matrix [14], only the vendors who would like to introduce their tools in the database by paying a fee are included. Vendors provide the capabilities of their tools in detail for comparison. Therefore, this is a credible source of evaluation in our survey. Table 4.2 shows the tools we have selected for evaluation.

HAT	Creator	Software License
Author-IT	Author-it Software Corporation	Proprietary
Doc-To-Help	ComponentOne	Proprietary
Dr. Explain	Indigo Byte Systems	Proprietary
Flare	MadCap Software	Proprietary
Help & Manual	EC Software	Proprietary
HelpServer	4ST	Proprietary or SaaS
RoboHelp	Adobe	Proprietary

Table 4.2: General overview of selected Help Authoring Tools

The capabilities of these tools are well-defined in [14] in which their vendors provide detailed information about them. To give the descriptive summaries of the tools, we can list them as follows:

Author-IT maintains its content through database operations. It provides different output formats for the help content. The editor and associated user interfaces are somehow complex that the learning curve can be long-lasting. On the one hand, it supports content localization and some other valuable features by means of its add-ons [97].

Doc-To-Help stores the meta-information about the help content in database. Several output formats such as CHM, PDF, RTF and Web Help are supported by this tool. Topic-based associations are also possible here, and it supports text creation in MS Word to the full extent [98].

Dr. Explain enables its users to capture application windows automatically through which we can add some call-outs for elements by interactive navigation [114].

Flare is an advanced tool for professional technical writers, having an efficient integrated editor. The translation work-flow is sophisticated by means of some specific built-in features [91].

Help & Manual is said to be one of the most cost-effective HATs. Single source publishing is extensively addressed for several product versions [92].

HelpServer is actually a Web-based tool for help authoring. It can be utilized both as a content management system and as a help system [118].

RoboHelp has been a leadership contender of HATs with its complicated and advanced features. It includes version control, multi-author support and close integration with Microsoft Word and Adobe FrameMaker [93].

4.3.1 Evaluation Criteria

In the comparison matrix [127], there are 26 distinct categories used for evaluating *Help Authoring Tools*. In our survey, we focus only on *Context-sensitive Help* criteria that can lead us to form a basis of comparison for our own tool framework. We have analyzed seven previously stated HATs in terms of context-sensitivity thanks to the effective comparison mechanism provided in [127]. Table 4.3 shows the features we have looked for in the selected tools, and reveals whether they are supported or not.

4.3.2 Results of the Assessment

Help authors, engineers and developers work together to create the links between the target application and the associated help content. Analyzing Table 4.3, Doc-to-Help [98] and Flare [91] support majority of target applications according to CSH(Context-sensitive help)-prefixed features. Context-sensitive help APIs for Android and iOS systems are only provided by Flare and RoboHelp [93]. It is a noteworthy result that all of the tools provide aliasing and context-sensitive help for Windows applications. RoboHelp is the only tool, in our inclusive survey, that automates the creation of context-sensitive help for C++ applications.

All of the tools allow the user to edit map numbers by topic, which brings easier integration. Also, they have built-in context-ID editors enhancing the effectiveness of context-sensitivity. The ease of use is utmost in Help&Manual [92] to the best of our analysis. According to Matthew Ellison, one of the leading

Feature/Tools	Author-IT	Doc-to-Help	Dr. Explain	Flare	Help & Manual	HelpServer	RoboHelp
Aliasing	✓	✓	✓	✓	✓	✓	✓
API	✓	✓	✗	✓	✓	✓	✓
Automatic mapping of topics	✓	✓	✓	✗	✓	✓	✗
Automatic workflow for creating context-sensitive Help for C++ applications	✗	✗	✗	✗	✗	✗	✓
CSH APIs for Android and iOS	✗	✗	✗	✓	✗	✗	✓
CSH: .NET	✓	✓	✓	✓	✗	✓	✓.NET API
CSH: existing (other) applications	✗	✓	✓	✗	✗	✓	✗
CSH: Test Dot Net output links internally	✗	✓	✗	✓	✗	✗	✗
CSH: Web/-Mac/*nix applications	✓	✓	✗	✓	✓	✓	✓
CSH: Windows applications	✓	✓	✓	✓	✓	✓	✓
Edit map numbers by topic	✓	✓	✓	✓	✓	✓	✓
Embedded Help	✗	✓	✗	✓	✗	✓	✗
Map Help topics to interface visually	✗	✓	✓	✗	✗	✗	✓
Map number/context ID/map ID editor	editing map numbers by topic	✓	✓	✓	✓	✓	✓

Table 4.3: Context-sensitive characteristics of selected Help Authoring Tools

professionals in the field, Flare is the most advanced help authoring tool, stating in his presentation [87].

Collectively assessing the results, context-sensitivity is obtained through similar approaches in all of the tools, and there should be novel and cutting-edge technologies used for this purpose. At some time, the tools tend to behave like *Content Management Systems* since the major objective of help authoring is not the presentation issue of help content, but the successful integration of it somehow.

The prices of tools range within a considerable interval, meaning that the cost-effectiveness should be further analyzed for all *Help Authoring Tools*. For the sake of an unbiased academic-purpose survey, we have not take the prices of selected HATs into account, but instead we assessed them as incurring equal costs.

Chapter 5

Tool Framework: Assistant-Pro

In this chapter, we discuss the tool framework for developing context-sensitive user assistance systems. Section 5.1 provides the overview of this chapter. Section 5.2 describes the Aselsan case in which *Assistant-Pro* was originally developed. Section 5.3 presents the process modeling approaches. Section 5.4 provides the tool architecture. Section 5.5 demonstrates the implementation of *Assistant-Pro*. Finally, Section 5.6 explicates the cost model that we have used to evaluate the tool framework.

5.1 Overview

As we have discussed in previous chapters, *embedded user assistance* is a subcategory of *automated user assistance*. We propose a tool framework specifically for *context-sensitive user assistance* which is an important category of *embedded user assistance*. In context-sensitive user assistance, the provision of help is related to the states of the software. When a specific situation arises at run-time, help content is presented to the user with respect to the associated state. Going further, we define a target system as a set of states in a topic-oriented manner. The relation between topics and states describe the state, situation or feature of the software. There are some ways to provide context-sensitive user assistance

such as:

- Automatic tooltips over controls
- Notifications in the status bar
- New panes associated with button actions

An important yield of context-sensitive user assistance is that users do not necessarily stop what they are doing in order to open and find the immediate help. This advantage is of utmost importance since the users are usually unwilling to consult an external help mechanism in the flow of work.

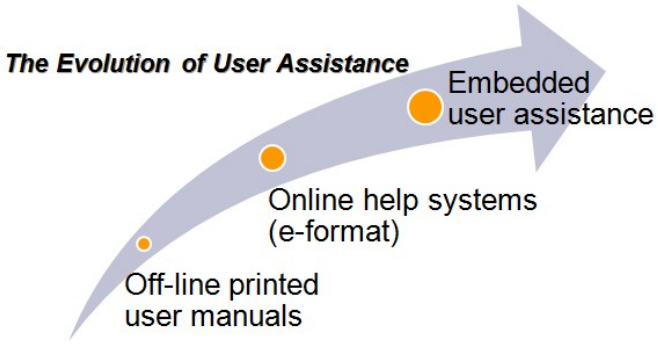

Figure 5.1: The evolution of user assistance

In our case, we deal with defining help based on the process state, which is known as *Process-Sensitive User Assistance*. The idea behind this approach is that users need to follow some certain steps while using a particular application. We can encapsulate the help content within the steps of a specific process. To give an example of this situation, safety-critical systems are composed of strict processes that have to be followed for a faultless behavior. The states and transitions in these kinds of systems are to be handled smoothly with an appropriate user assistance mechanism.

Developing context-sensitive user assistance systems is somewhat problematic, and we have to meet several challenges. We can list some of these challenges as follows:

- User-assistance concerns cannot be easily localized in single modules and as such tend to crosscut multiple modules.
  - modularity $\Downarrow$
  - maintainability $\Downarrow$
- The reuse of user assistance tools for different applications is required, where modularizing crosscutting concerns is required for multiple applications.
  - change in time vs. change in space
- Developing custom-based help assistance for each separate application is laborious.
  - development time $\Uparrow$

Considering the above challenges, the lack of generalized context-sensitive user assistance solutions has emerged.Besides,the utilization of methods, algorithms and tools is rather scattered. In order to resolve these issues, we developed an application framework consisting of several tools to integrate context-sensitive help content in a modular way. Prior to conducting the systematic literature reviews in this field, an initial version was developed in the context of Aselsan, Turkey, which is a leading high technology, multi-product defense electronics company introducing state-of-the-art equipment and software intensive systems solutions for both sophisticated military and professional applications [1, 5].

5.2 Case Description: Aselsan

The interest in context-sensitive user assistance solutions is increasingly growing since they enhance the support for guidance of systems. Previously mentioned challenges have not been explicitly undertaken so far according to the results of our systematic literature reviews. We specifically consider the state of the process to develop embedded user assistance systems based on two pilot project applications developed at Aselsan [5]. The goals of user assistance and the need

for embedded user assistance have also been encountered for the commercial applications that are developed by Aselsan.

The problem with this approach was that the users very often demand further training to operate the system effectively. Consequently, the company incurred some undesired costs just because of user assistance concerns.

The goals of user assistance and the need for embedded user assistance have also been encountered for the commercial applications that are developed by Aselsan [5]. The earlier conventional approach used in Aselsan [1]:

- A senior system engineering team develops a scenario.
- The development team implements the scenarios in code.
- UA specialists write the user manuals in both electronic and hard-copy format.
- A link is defined in the application to the electronic user manuals as PDF documents or wiki pages.

The users are supposed to know the scenario as defined in these documents at run-time. In case the user requires help they need to open the PDF documents and search for the information that solves their problem. To support the definition of help, also third party tools, such as Fast-Help [128] is used. Fast-Help is a Windows Help File Generator that produces online and offline documentation in electronic formats such as HTML, PDF or .HLP. Although these third party products help to better organize the help files and provide better presentation and query mechanisms, the provided help is still not embedded in the application. As such, the earlier mentioned problems of external help still remain.

5.2.1 Example Case Applications

Example Case Application 1 - *Message Management System* MMS is an asynchronous e-mail application for sending and receiving messages

among connected peers. A user can view his/her inbox, and can reply, forward, or delete messages. Also, a new message can be composed to be sent to the other peer by optionally saving it as a template. Incoming and outgoing messages are displayed in the graphical user interface of the application as shown in Figure 5.2. The persons concerned with this system make use of the features of database storage about the information about messages.

Example Case Application 2 - *Listing and Listening of Voice Records*

LLVR helps to deal with voice records that are of different audio formats. Database queries are used for accessing voice records as a list. Some operations like play, pause, stop and replay can be used. Also, partial play is possible by specifying an interval. The operator can also attach some textual content about the record to the database. The records can be imported from and exported to a database in a select/insert fashion. Again, this application has no guidance to users. The user needs explicit guidance to operate the application effectively and to reduce the potential faults due to misuse [1].

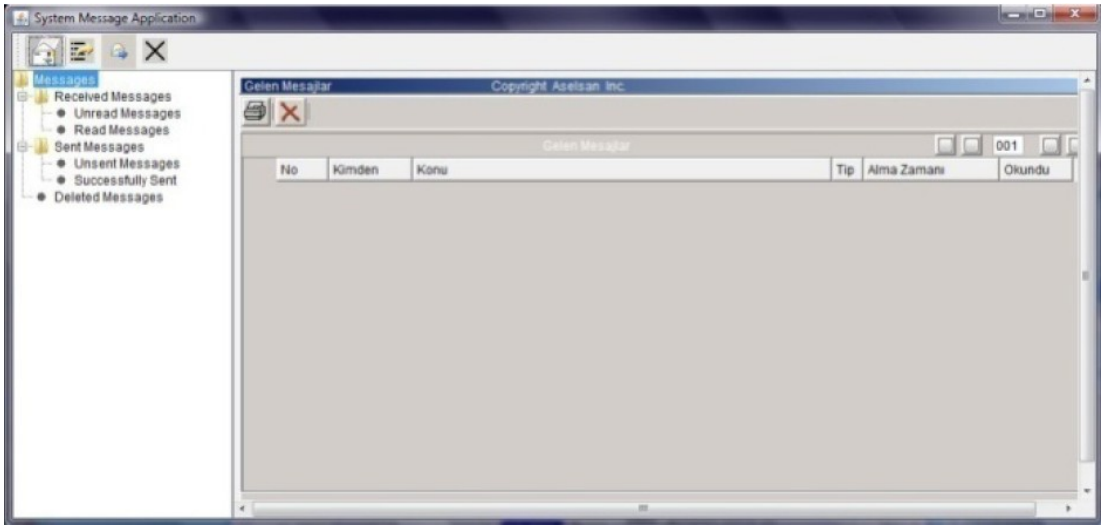

Figure 5.2: General view of Message Management System (MMS)

5.2.2 Problem Statement

Consequently, the obstacles related to the development of context-sensitive user assistance systems are:

Crosscutting behavior User assistance concerns generally result in either scattering (code duplication), tangling (significant dependencies between systems), or both. Sometimes being considered as minor requirements, these concerns are not decomposed from the remaining of the systems in hand at the prior stages of development like design and implementation. In other words, user assistance concerns are the aspects of a program affecting several parts of the system design. We will discuss why we cannot smoothly fulfill these concerns via object-oriented or procedural solutions. In practice, the solutions for these concerns should not bring some system interdependencies because the main functionality of an application is not always all about providing guidance for users. Going further, we should somehow place the help concerns on the same level of other concerns that require help for the users. In the example case applications, we encountered some code segments of button,window or frame actions that require the provision of help for users. These related code segments can be frequently changed as needed, causing the maintenance to be problematic. Considering these issues, we employed *aspect-oriented programming* techniques to encapsulate user assistance concerns into *aspects* in order to secure modularity.

Reusability Any solution for software user assistance should be reused just through very little or no modifications. By this way, the development time of a software system, in which the configurations on the code are localized, can be reduced. The main argument here is that user assistance concerns are pretty much alike in most cases. To illustrate with examples, the previously mentioned case applications require almost the same guidance functionalities. AOSD may not suffice for reuse across multiple applications. Thus, we should adopt at least one of these solutions:

→ Abstraction by a super-categorical aspect

→ Model-driven transformations

In both cases, some manual interference would be needed to ensure the completeness of guidance aspect.

Process modeling Providing just-in-time user assistance relies on the explicitness of the process followed. Although some strict processes that have to be followed for a faultless behavior are defined in external documents, the definition is not traceable at code-level specifications. Thus, the associated processes are to be modeled beforehand, and presented to the applications in proper forms.

Help Authoring Tools The framework or tool kind of solutions for embedded user assistance do not support modularity for several reasons. Besides, we can have a diverse range of applications to be taken into account which may have different layouts of user interfaces. We analyzed the complete list of help authoring tools in WritersUA [129], which specializes in providing quality training and publications for the community of user assistance professionals and is the premier source for world-class, world-wide user assistance conferences and seminars. The solutions in this list involve several kinds of user assistance such as online help, wizards, web sites, printed documentation, and improvements to the application user interface. However, there is no standard technique for embedded user assistance reported. We did not come upon any aspect-oriented solution modularly extending software systems with user assistance.

5.3 Process Model vs. User Assistance

Process-sensitive embedded user assistance requires fine-grained modeling of processes. We analyzed *OMG Software and Systems Process Engineering Meta-Model Specification* (SPEM 2.0) [130] that enables implementers to choose the generic behavior modeling approach that best fits their needs. It does not stand for

a generic process modeling language, nor does it even provide its own behavior modeling concepts. It provides some particular mechanisms to improve such generic behavior models that are characteristic for describing development processes. The main objective of this approach is to provide the choice and flexibility to SPEM 2.0 specification implementers to select the appropriate behavior model formalism themselves [130]. The associated meta-model is specifically dedicated to software development processes. Thus, user assistance concerns are not met in this specification to the full extent. Figure 5.3 shows the overall structure of SPEM 2.0 meta-model.

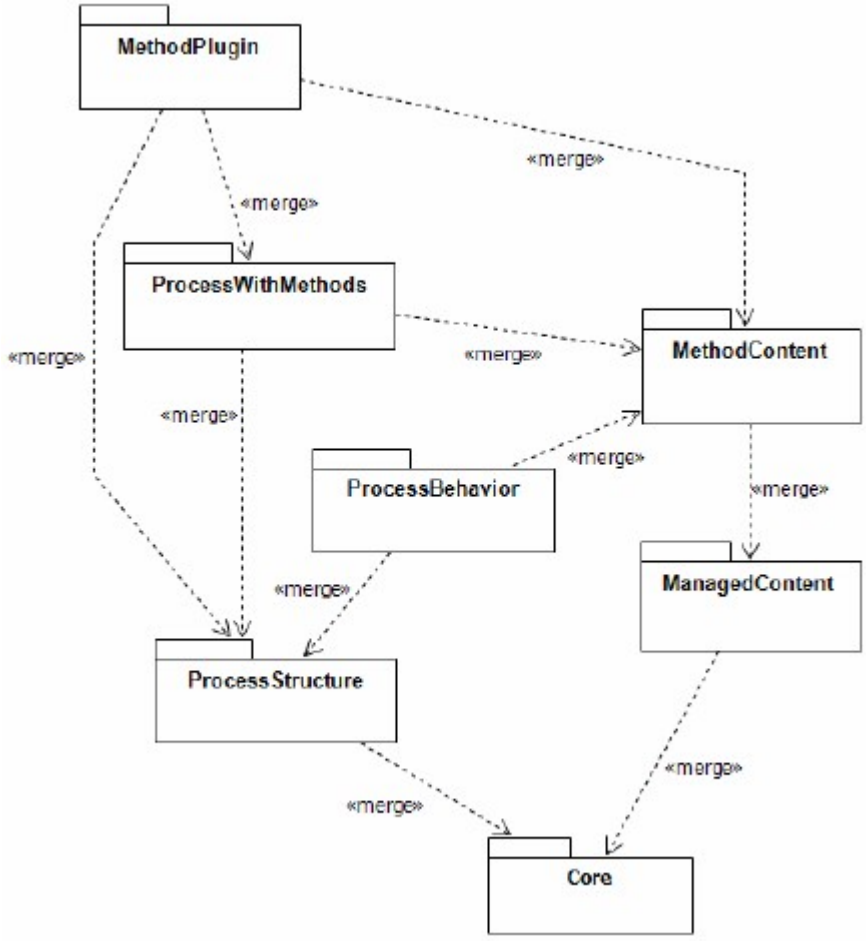

Figure 5.3: General overview of SPEM 2.0 Meta-Model

Also, SPEM 2.0 defines some key terminologies related to process modeling.

Guidance is one of this concepts taking place in meta-model specification, and it is specified as a *Describable Element*. The *Guidance* is said to be classified with specific kinds for which specific structures and contents are assumed to exist. The kinds of *Guidance* are well described in SPEM 2.0, and also related terminologies are stated. Figure 5.4 shows the Venn diagram of the key terminology.

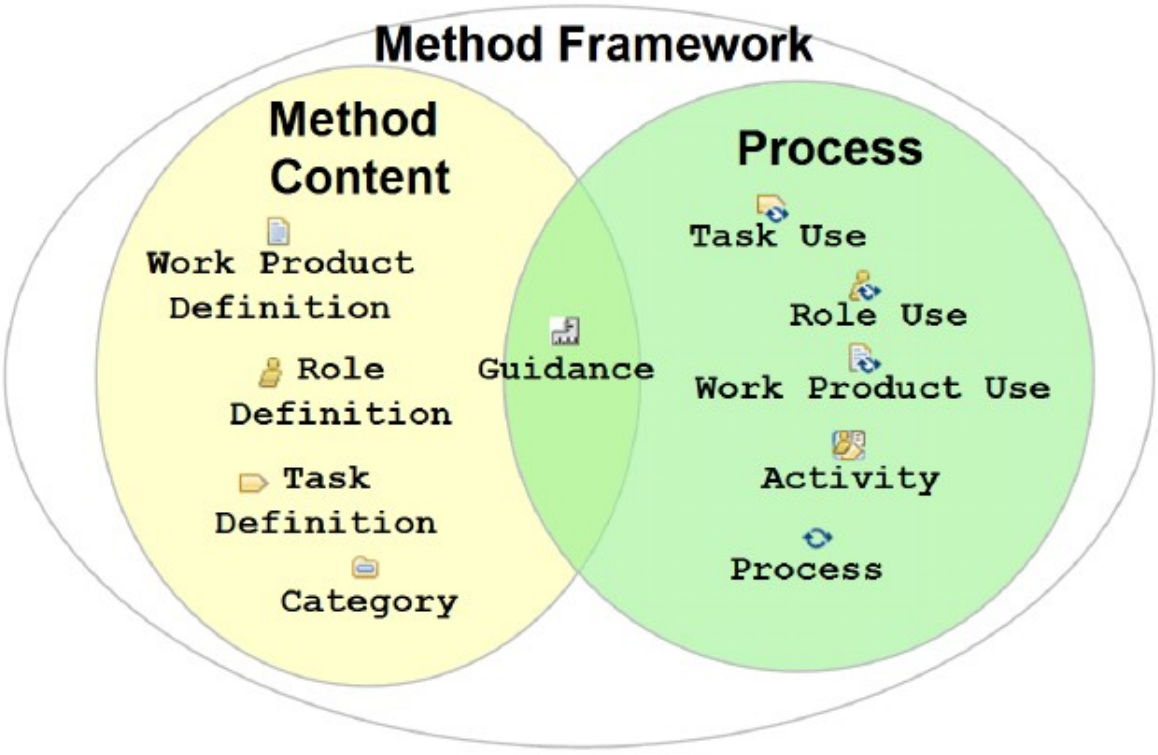

Figure 5.4: Key concepts defined in SPEM 2.0

We can now analyze the kinds of *Guidance* in detail. There are 16 different sub-categories defined in SPEM 2.0, and only some of them are related to our case that is providing embedded context-sensitive user assistance in software systems. We can list the sub-categories as follows [130]:

Checklist is a specific type of guidance that identifies a series of items that need to be completed or verified.

Concept is a specific type of guidance that outlines key ideas associated with basic principles underlying the referenced item.

Estimate (metric kind) is a specific type of Guidance that provides sizing measures, or standards for sizing the work effort associated with performing a particular piece of work and instructions for their successful use.

Estimation Considerations (metric kind) qualify the usage and application of estimation metrics in the development of an actual estimate.

Estimating Metric (metric kind) describes a metric or measure that is associated with an element and which is used to calculate the size of the work effort as well as a range of potential labor.

Example is a specific type of Guidance that represents a typical, partially completed, sample instance of one or more work products or scenario-like description of how Task may be performed.

Guideline is a specific type of guidance that provides additional detail on how to perform a particular task or grouping of tasks (e.g., grouped together as activities), or that provides additional detail, rules, and recommendations on work products and their properties.

Practice represents a proven way or strategy of doing work to achieve a goal that has a positive impact on work product or process quality. Practices are defined orthogonal to methods and processes.

Report is a predefined template of a result that is generated on the basis of other work products as an output from some form of tool automation.

Reusable Asset provides a solution to a problem for a given context. The asset may have a variability point, which is a location in the asset that may have a value provided or customized by the asset consumer.

Roadmap is a special Guidance Kind that is only related to Activities. A Roadmap represents a linear walkthrough of an Activity, typically a Process.

Supporting Material is a catch-all for other types of guidance not specifically defined elsewhere. It can be related to all kinds of Content Elements, i.e., including other guidance elements.

Template is a specific type of guidance that provides for a work product a predefined table of contents, sections, packages, and/or headings, a standardized format, as well as descriptions how the sections and packages are supposed to be used and completed.

Term Definition define concepts and are used to build up the Glossary. They are not directly related to Content Elements, but their relationship is derived when the Term is used in the Content Elements description text.

Tool Mentor is a specific type of guidance that shows how to use a specific tool to accomplish some piece of work either in the context of or independent from a Task or Activity.

Whitepaper is a special Concept guidance that have been externally reviewed or published and can be read and understood in isolation from other content elements and guidance.

The class diagram of *Guidance* kinds is shown in Figure 5.5. As it can be noted, only some of the previously mentioned items exist in this diagram as several of the kinds are related to estimation purposes.

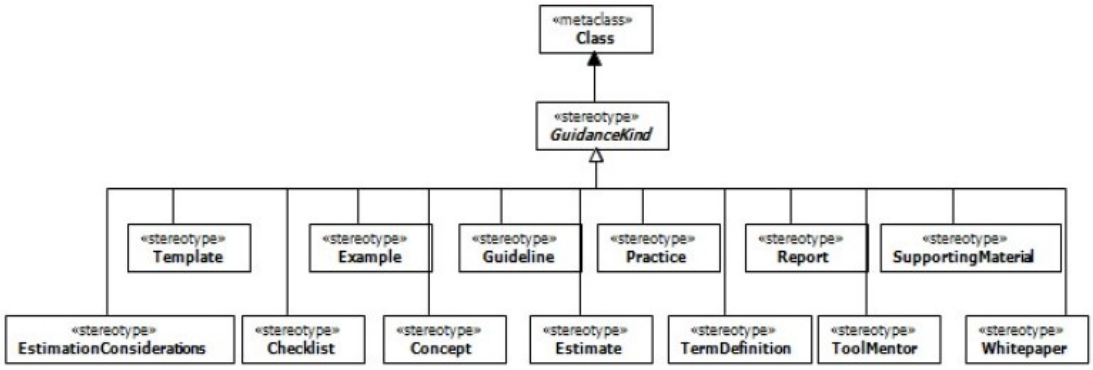

Figure 5.5: The Stereotypes related to Guidance Kinds

Adopted from the SPEM 2.0 specifications, we formed our process-sensitive user assistance model using some of the guidance kinds that can be employed in the provision of user assistance. There are two distinct parts in this model: one of them is related to process and the other is for user assistance. The items

in user assistance part are defined previously in *Guidance* kinds. The model is shown in Figure 5.6.

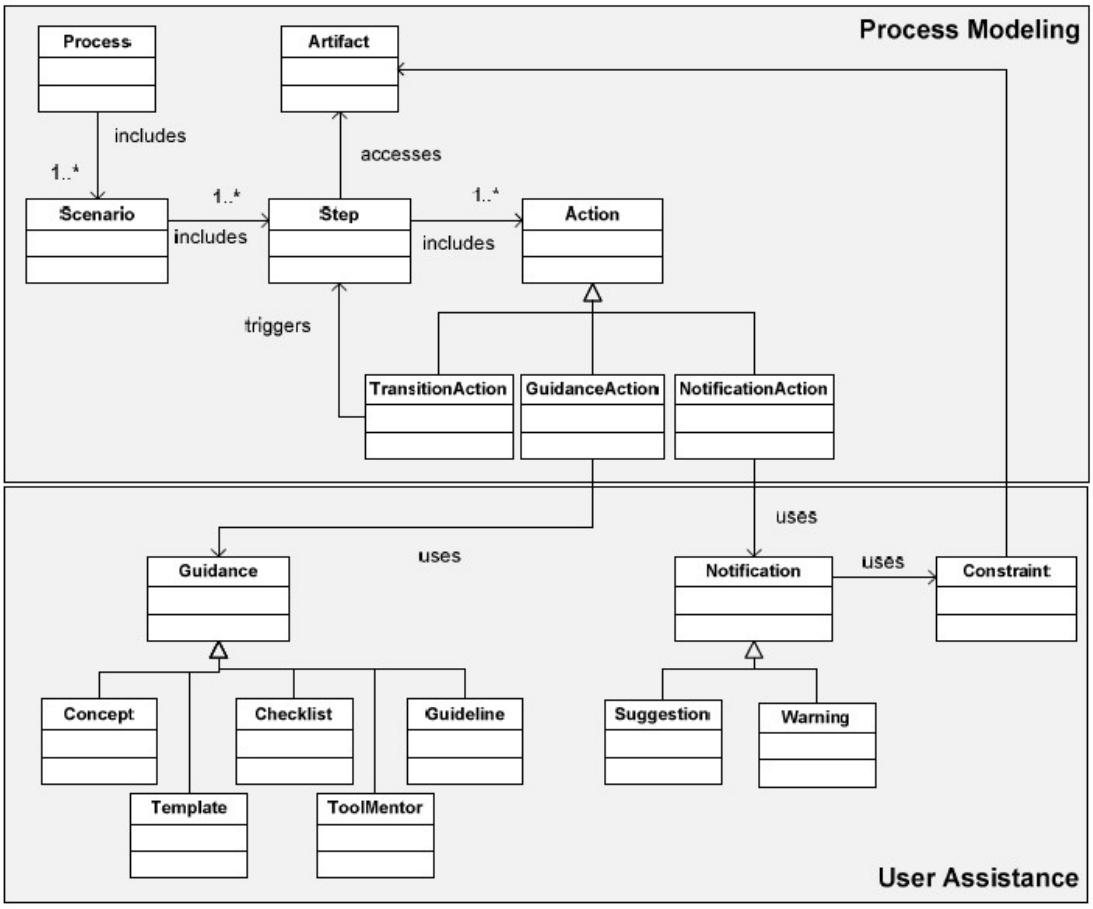

Figure 5.6: Embedded Process-Sensitive User assistance Model

5.4 Detailed Tool Architecture

Assistant-Pro has been developed specifically in the context of Aselsan [5]. Actually, it has two different development stages, namely versions, throughout the progress of this thesis. Version 1.0 was developed together with the practitioners in Aselsan and academics. The first published paper is [1] which is also the initiator of this thesis making inspirations. We have grounded our further studies on

this first version by using the concepts defined. To better reveal the progress of development, Figure 5.7 shows the basic flow of progress that we have followed.

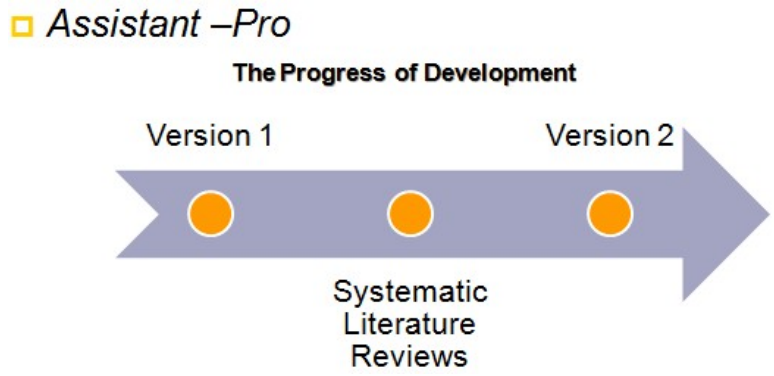

Figure 5.7: The Progress of Development

We have five main code packages and three main actors within the tool framework, and Figure 5.8 shows the usecase-package diagram of *Assistant-Pro* to better reveal how it is being used in practice.

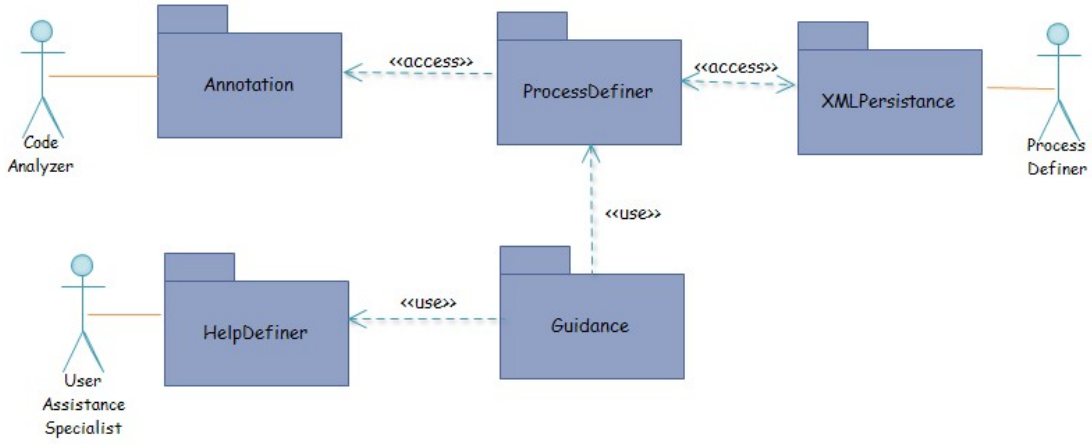

Figure 5.8: Use Case-Package Diagram of Assistant-Pro v2

As it is shown in Figure 5.9, *Assistant-Pro* is grounded on two major parts: *Framework* and *Application*. In the *Framework* part, we have *Context Definition Tool* and *Help Definition Tool*. *Context Definition Tool* is used by *Process Definer* for modeling the processes with possible strict flows to be followed. *User Assistance Specialist* uses *Help Definition Tool* to create help files corresponding

to process steps. Through the instrumentality of *Guidance Aspect*, the application is extended modularly with embedded process-sensitive user assistance. Detailed information about the framework's flow of work is given in the following subsections.

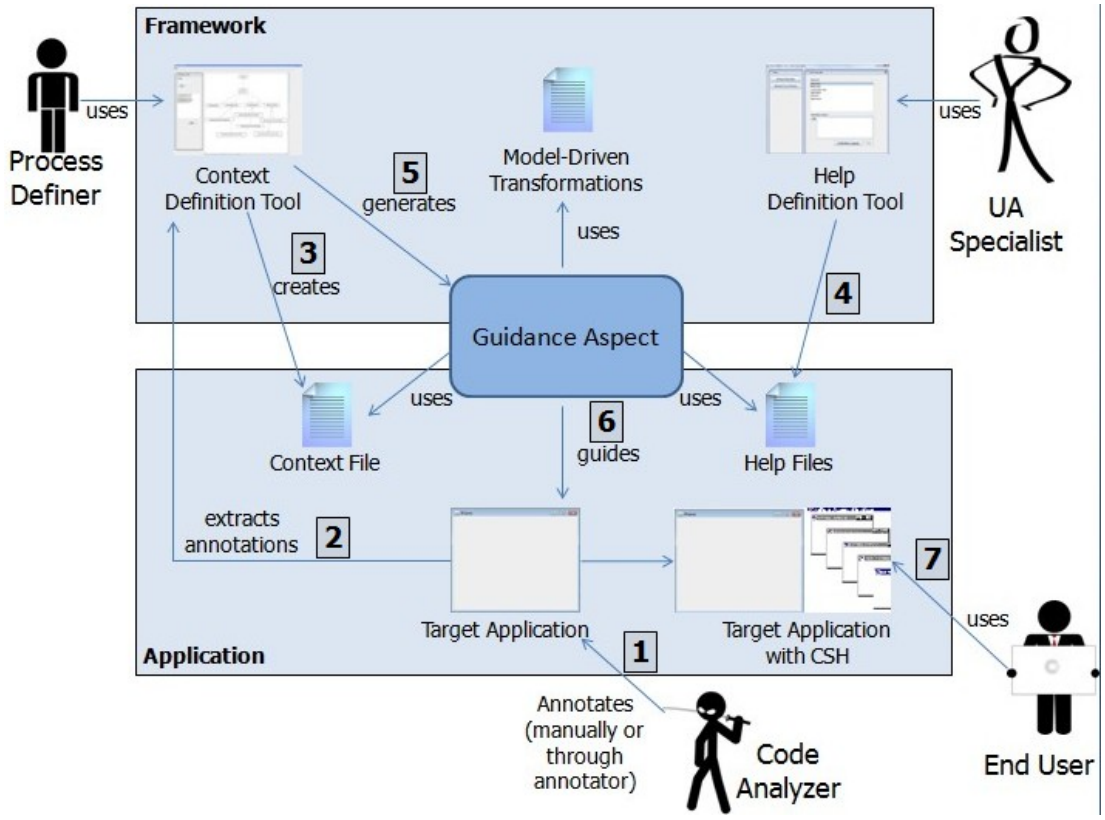

Figure 5.9: The Workflow of Assistant-Pro

We can further improve Assistant-Pro with Eclipse Modeling Project [131] in the cases where we define a user assistance meta-model for our domain of interest. In such cases, the flow of work would be similar to Figure 5.10.

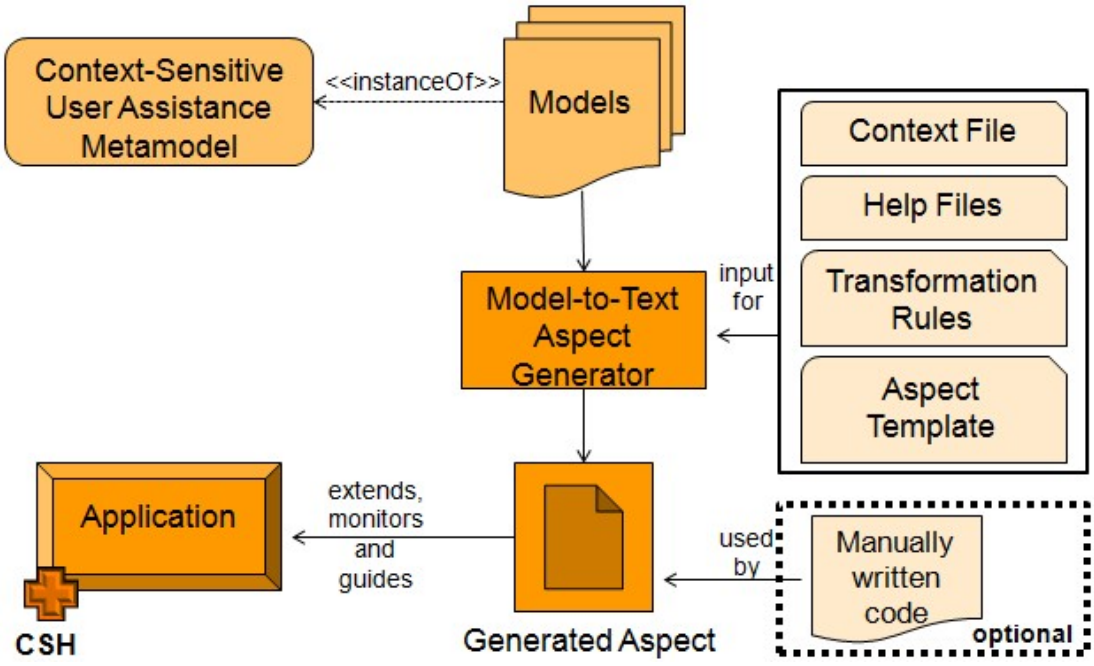

Figure 5.10: Model-Driven Development of User Assistance

5.4.1 Annotation

At this very beginning step, *Code Analyzer* is supposed to annotate the user interface components of the target application with proper annotations. There are two main annotations specified in the Aselsan case [1]:

@ApplicationInitialization is used for annotating the main interface component of the target application in order to create a reference to the *Guidance Aspect*.

@Event (name="") is used for determining the method calls that lead to transitions. For this purpose, graphical user interface states have to be known in advance to associate them with relevant process steps. The property *name* is to be unique among all events, and they are further used within the *Process Definition Tool*.

Version 1 of Assistant-Pro required *Code Analyzer* to manually annotate the code, which brings comprehensive analysis. Along with Version 2, we have incorporated an *Annotator* component which explicitly relies on the target application code. *Annotator* component requires the specification of annotations seamlessly; in other words, we have to know the annotations that we want to examine for weaving help files. The annotation process can be done either by command-line statements or by an associated user interface. By this way, we can generalize the use of *Assistant-Pro* through automating the whole process of code analysis. In the following subsections we will describe the annotation mechanisms that we adopted for the tool framework.

5.4.1.1 Annotation Processing Tool (apt)

apt is a command-line utility for automating the annotation process. It is mainly composed of a set of reflective APIs and supporting infrastructure to insert annotations on the program code. The reflective APIs enables us to bring up a build-time, source-based, read-only view of program structure. Basically, they are dedicated to model JAVA type system after the addition of generics [132]. The principal work-flow of *apt* is:

1. Running annotation processors that can produce new source code and other files.

2. Compiling both original and generated source files, thus easing the development cycle.

Firstly, *apt* reveals the annotations existing on the source code in hand. Then, an interesting feature of *apt* comes into play that *apt* tries to extract annotation processor factories that are written. The tool interrogates the factories about the annotations they are related to. Hereafter, *apt* asks a factory to form an annotation processor if the factory processes an annotation present in the program code. Finally, the annotation processors are ready to run. If the processors have

created new source files, *apt* will repeat this process until no new source files are generated.

Manually maintaining consistency among the entire set of files is not required, and only the base file would need to be configured since the derived files are generated. The *apt* tool is designed for creating the derived files. We could have only employed doclets that are the programs written in the JavaTM programming language that use the doclet API to specify the content and format of the output of the Javadoc tool. However, generating the derived files based on annotations with *apt* is more feasible, compared to writing our own parser, since it has [132]:

- A cleaner model of the declarations and current type structure of programs
- A more contemporary API design
- A support for recursive processing of newly generated files and can automatically cause compilation of original and generated source files
- A support reflective programming tasks.

The associated *Mirror API* is used for modeling the semantic structure of an application. It reveals representations of the entities stated in a program code, such as classes, methods, and fields. However, the structures below the method level, like individual statements and expressions, are not represented. Writing annotation processors to examine and process the annotations of program elements is the main support included in this API. An annotation processor can create new source files and XML documents to be used in conjunction with the original code [133].

5.4.1.2 Annotation File Utilities

The *Annotation File Utilities* provide three tools to read and write annotation files [134].

insert-annotations reads annotations from an annotation file and inserts them into a class file

extract-annotations reads annotations from a class file and writes them out to an annotation file

insert-annotations-to-source reads annotations from an annotation file and inserts them into a Java source file

There is no *extract-annotations-from-source* tool, which is supported by *apt* [132]. That is why, we merged the functionalities of the two annotation processing mechanisms (i.e. [132, 134]) to ensure a fully-automated annotator dedicated to embedded process-sensitive user assistance.

Consequently, we can now execute command line statements for annotation processing such as:

1. insert-annotations TargetApplication.ClassName AnnotationFile.jaif
2. extract-annotations TargetApplication.ClassName
3. insert-annotations-to-source indexFile.jaif TargetApplication/MyClass.java

5.4.2 Process Modeling and Definition

As it is already shown in Figure 5.9, *Process Definer* uses *Context Definition Tool* to specify the process steps under a specific work flow. In order to precisely define the process, the following activities are to be performed as was in *Version 1 of Assistant-Pro* [1]:

- Extract annotated events
- Define the state abstractions of the user interface
- Define the transitions between states

- Associate transitions with the annotated events of the application
- Define the processes that represent a set of scenarios.

Depending on the annotation style (i.e. manual or automated), *Context Definition Tool* parses the files including annotated events. In case we have several kinds of annotations, we should set them within the *Context Definition Tool* accordingly. By using the tool, *Process Definer* defines the process along with the states and transitions. *Context Definition Tool* allows us to enter both states and transitions. The names of states are specified in the tool, and also the transitions associate two states both in the process model and the applicaton code (i.e. events). *Process Definer* does not need to deal with the exact signatures of the events since, by using annotations, we acquire an oblivious tool in that sense.

Process Definer outputs the context file (in Figure 5.9), including the whole set of states and transitions. In Figure 5.11, there is an illustration of the process definition in LLVR (Listing and Listening of Voice Records) application of Aselsan. The defined steps include the following in this application:

- The Record is Stopped,
- The Record is being Played
- The Record is Paused

The upper level processes related to these steps are:

- Play the Record
- Stop the Record
- Pause the Record

We can define new processes and steps through the tool by using the associated menu items. All in all, the final output of *Context Definition Tool* (i.e. context

file) reveals a number of paths that can be followed, but in practice, there can be some obligatory and optional processes at the same time. This situation is also handled within the context file.

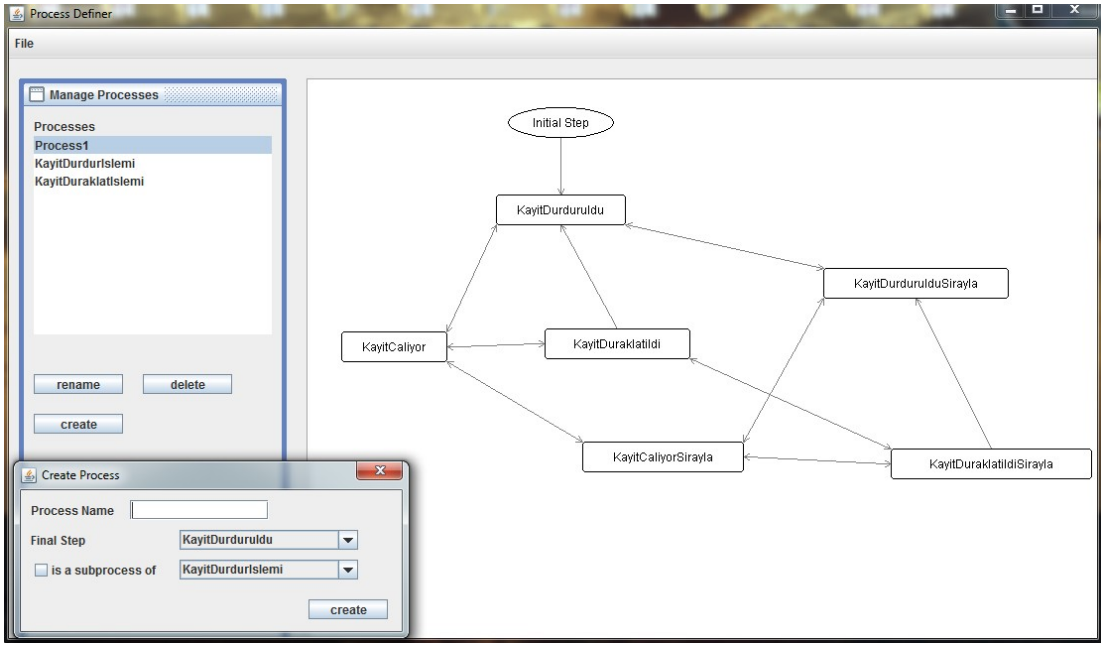

Figure 5.11: Process Definition Tool - LLVR

5.4.3 Help Definition

One of the most crucial stages in the *Assistant-Pro*'s flow of work is the definition of help elements. *User Assistance Specialist* assumes the control of the tool by defining associated help content related to transitions. There are two main actions in this tool: *Manage Step Helps* and *Manage Process Names*. *Manage Step Helps* option is related to defining the actual help content related to steps as shown in Figure 5.12 shows the snapshot of this option. Also, we can enter the names of processes in different languages as shown in Figure 5.13. Each supported language is listed in a tabbed content pane in which we can define new languages with proper abbreviations.

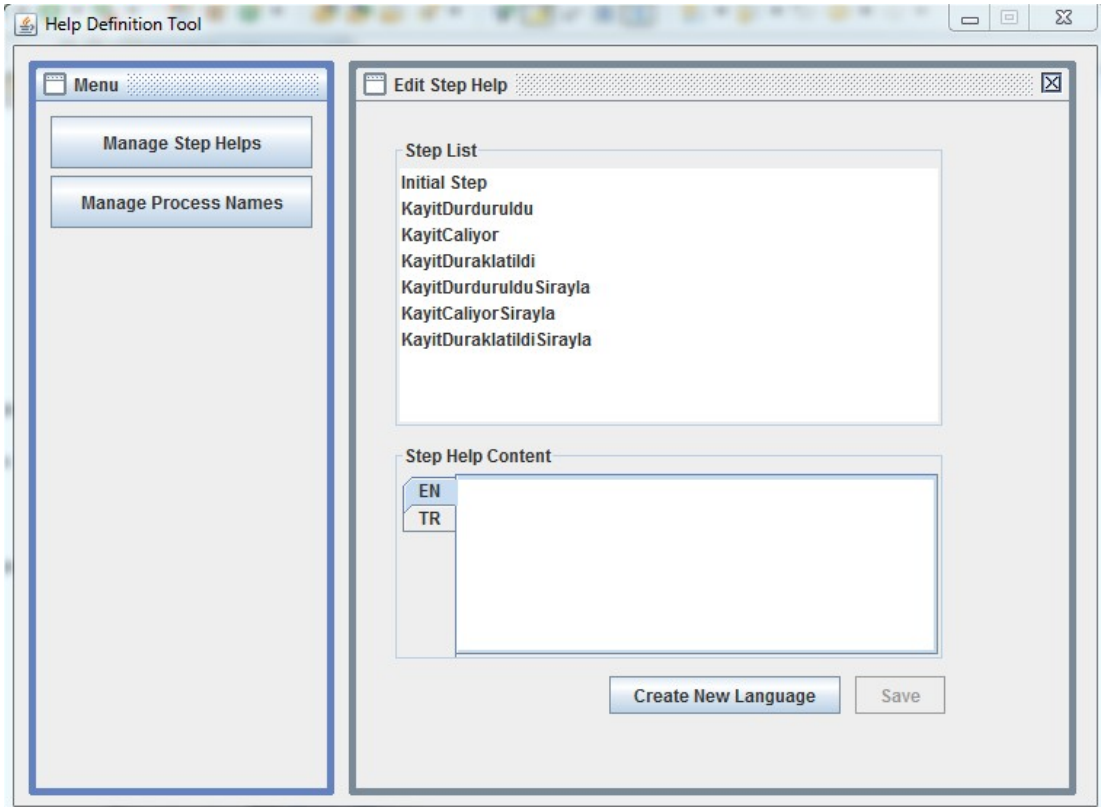

Figure 5.12: Definition of Help Content

5.5 Aspect-Oriented Implementation

Context-aware systems are supposed to sense the environment and adapt its behavior according to some contextual conditions. However, such context-awareness concerns are usually implemented by using traditional Object-oriented mechanisms, in which *if* statements are scattered over the applications to achieve context-dependent behavior. The utilization of usual programming techniques leads to complicated design which ruins reusability and maintainability since context acquisition and adaptation concerns are usually scattered in a tangling way.

Aspect-oriented programming has recently been proposed in order to modularize several parts of a context-aware application, such as context acquisition, location/proximity context and context-dependent behavior. This approach allows

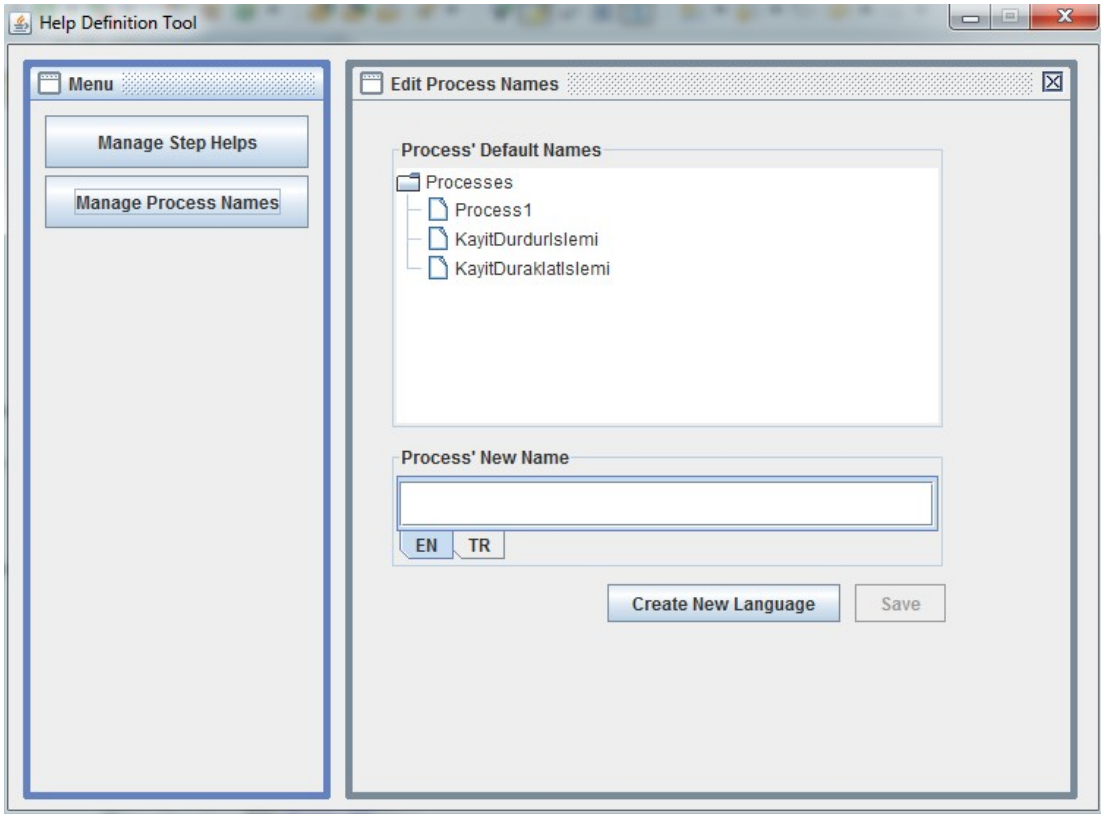

Figure 5.13: Managing Multilingual Process Names

creating aspects capable to perform context fusion/fission and adapting modeled context into non-modeled contextual information that may be required by application to perform the context-dependent behavior adaptation [135].

We have used AspectJ [131] in order to define *Guidance Aspect* which includes methods for reading context file and help file. The aspect is first defined as *abstract* from which concrete aspects can be generated by little or no manual modifications. The generation of aspect is carried out within the tool framework, going along with the target application.

A sample concrete Guidance Aspect is given in the Listing below. This example is mostly about catching button actions. The upper-level specification of catching the events triggered is an abstract pointcut which is extended in the below example as button actions.

Listing 5.1: A Sample Concrete *Guidance* Aspect

```
1  package help;
   import java.util.ArrayList;
3  import java.awt.Component;
   import java.awt.event.*;
5  import javax.swing.*;
   import xmlpersistence.XMLManager;
7  import com.my.example.case.application.*;
   public privileged aspect GuidanceAspect
9  {
       HintFrame hintFrame;
11     WizardFrame wizardFrame;
       ProcessManager processMgr;
13     int helpOptionPanelHeight = 40;
       JFrame mainFrame;
15     pointcut mainFrameInit() : call(Window.new(..));
       after() returning (JFrame mainFrame): mainFrameInit()
17     {
           this.mainFrame = mainFrame;
19         processMgr =
           (ProcessManager)XMLManager.readFromXML("help.xml");
21         processMgr.setCurrentStep(processMgr.getInitialStep());
           hintFrame = new HintFrame( "Hint", mainFrame.getWidth());
23         wizardFrame = new WizardFrame(mainFrame.getHeight(),
                                         processMgr.getProcesses());
25         ArrayList<String> processes = new ArrayList<String>();
           processes.add("Process1");
27         String transitionAction =
                  thisJoinPoint.getSignature().toShortString();
29         stepChange(transitionAction);
           setHelpWindowLocations();
31         activateHelpWindows(true);
           mainFrame.addWFocusListener(new WindowFocusListener());
33         mainFrame.addWindowListener(new WindowListener());
           new LocationChecker().start();
35     }
       private Component getFrame() {...}
37     public void activateHelpWindows(boolean active){...}
       public void setHelpWindowLocations() {...}
39     class LocationChecker extends Thread{...}
```

```
    pointcut buttonClicked() : call (*.*ButtonClicked(..));
41  void around(): buttonClicked(){...}
    pointcut buttonClickedWithResult()
43          : call (*.*ButtonClicked(..));
    boolean around() : buttonClickedWithResult(){...}
45  after() returning(boolean result)
            : buttonClickedWithResult(){...}
47  pointcut windowClosing() : execution (*.windowClosing(..));
    void around(): windowClosing(){...}
49  public void stepChange(String transitionAction){...}
    pointcut selectedProcessChanged()
51          : execution (void WizardFrame.valueChanged(..));
    after() returning() : selectedProcessChanged(){...}
53  public void giveHelpToWizardFrame(){}
}
```

Aspect generation process can be done either using the *generateAspect* module or manual specifications. *generateAspect* module uses context file defimed by *Context Definition Tool* and help file defined by *Help Definition Tool*. Also, we have a simple aspect specification file that contains the types of frames, panes or bars to be incorporated into the user interface of target application. This process is depicted in the Figure 5.14.

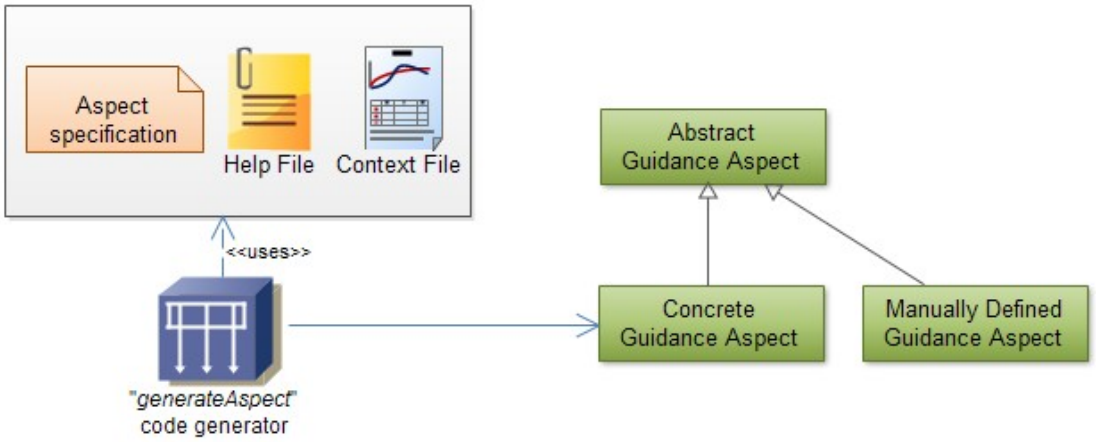

Figure 5.14: Aspect Generation Procedure

Version 1 of Assistant-Pro was of domain-specific nature in the context of Aselsan. The abstract *Guidance Aspect* was reusable for the applications within the company. We have re-designed the package structure and classes in order to provide a more generalized solution as a context-sensitive embedded user assistance system. We have incorporated an *XML Aspect* that monitors the read and write operations during the assistance. This aspect resolves the issues of scattered XML statements that tangle the pure source code of classes. Also, with some little configurations, users are able to adapt new models like user model, interest model and collaboration models into schemes. This inference is a result of the systematic review that we discussed in Chapter 3. Also, we created an extra help schema for the specification of images that can be required to show during user assistance. Moreover, process names and step names are kept in *(*).properties* files that are filled by the operations of *Context Definition Tool*, respectively.

Listing 5.2: A Sample *Help.xml* File

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.5.0_13" class="java.beans.XMLDecoder">
 <object class="help.ProcessManager">
  <void property="processes">
   <void method="add">
    <object id="Process0" class="help.Process">
     <void property="processName">
      <string>myProcess</string>
     </void>
    </object>
   </void>
   <void method="add">
    <object id="Process1" class="help.Process">
     <void property="processName">
      <string>myProcess2</string>
     </void>
    </object>
   </void>
   <void method="add">
    <object id="Process2" class="help.Process">
     <void property="processName">
      <string>myProcess3</string>
     </void>
```

```
24      </object>
       </void>
26    </void>
      <void property="steps">
28     <void method="add">
        <object class="help.Step">
30       <void property="name">
          <string>First Step</string>
32       </void>
         <void property="transitions">
34        <object class="java.util.ArrayList">
            <void method="add">
36            <object class="help.Transition">
                <void property="buttonName">
38                <string>initial step </string>
                </void>
40            </void>
          </void>
42        ........ <!-- more code -->
        </void>
44    </void>
    </void>
46  </object>
```

Help.xml file defines the processes and steps. Additionally, we can insert user models, interest models and (optional)collaboration models for different cases. We have tested the functionality of *Assistant-Pro* in several target applications. As was previously stated, Version 1 was tested and validated in the context of Aselsan in two case applications, MMS and LLVR. Version 2 has also been evaluated in quite different environments. For example, interestingly, an aspect-oriented game application was enhanced with the functionalities of *Assistant-Pro Version 2* which gave promising performance results. More importantly, a software project management tool (*4PM*), which was developed by me, was tested in conjunction with *Assistance-Pro Version 2*. The software project management tool (*4PM*) is noncommercial for now, and it involves the following functionalities:

- Managing software projects globally in international standards by the mediation of web, and having
- Basic project tracking,
- Viewing the situation of each and every team member throughout the project,
- Accessing the current project plan and its situation,
- Resource monitoring, resource allocation throughout the project and its activities,
- Accessing the documents and notes whenever needed.

The user types of this tool are: admin, project manager, lead developers and developers. However, the developer is not synonymous with the one in Aselsan case in which developers actually develop the user assistance systems. In *4PM*, developers are the ones working within a company under a software project, and they use the tool for project management purposes. The system overview of *4PM* is depicted in Figure 5.15. Also, Figure B.1, Figure B.2 and Figure B.3 demonstrate how 4PM operates in terms of its architecture, the packages under the control of *Guidance aspect* and major activities performed.

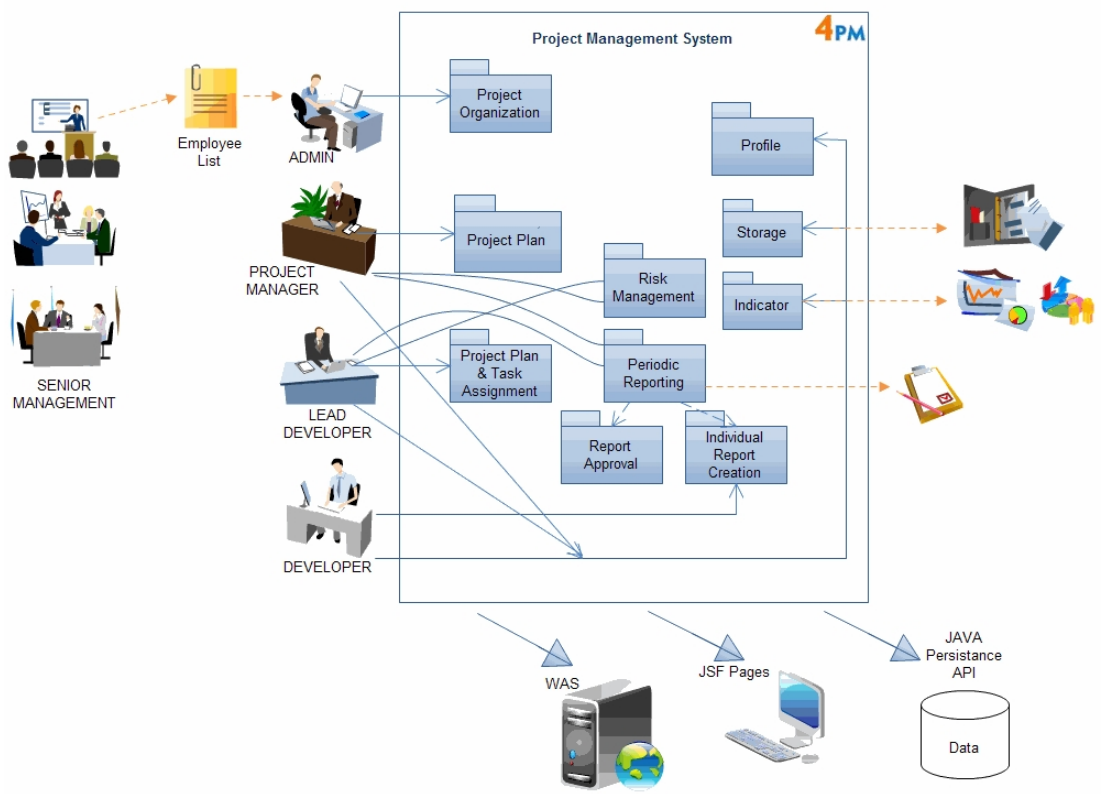

Figure 5.15: The system overview of 4PM-Project Management Tool

The utilization of *Assistant-Pro Version 2* in the software project management tool was achieved in several types compared to *Version 1*. For example, the members can see their assigned tasks in the projects directly at the main user interface. The changes in task assignments and the actions to take about them are presented to the user as a means of user assistance. Also, most of the pages support both context-sensitive and process-sensitive help. Figure 5.16 shows a sample of pages in which *Assistant-Pro* operates by providing hints.

Figure 5.17 shows one of the case applications of Aselsan which was modularly extended with *Assistant-Pro*. Other experiments were based on somewhat subjective evaluation which covered applications written in Eclipse IDE.

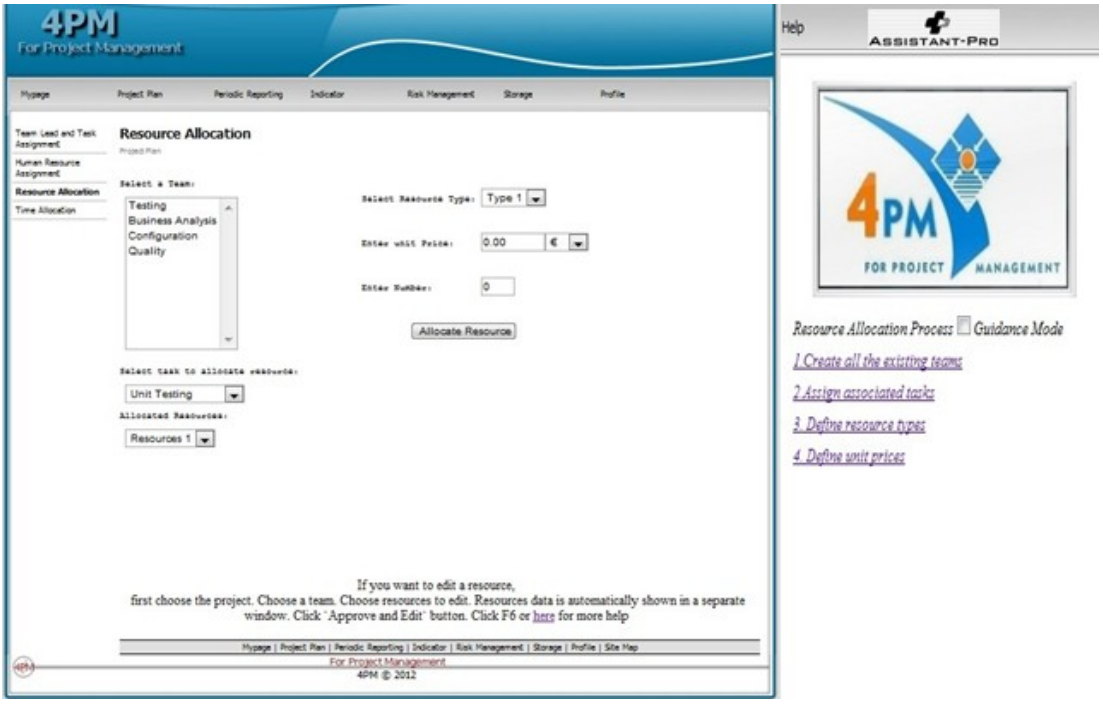

Figure 5.16: Sample Customer Preview of 4PM with Assistant-Pro

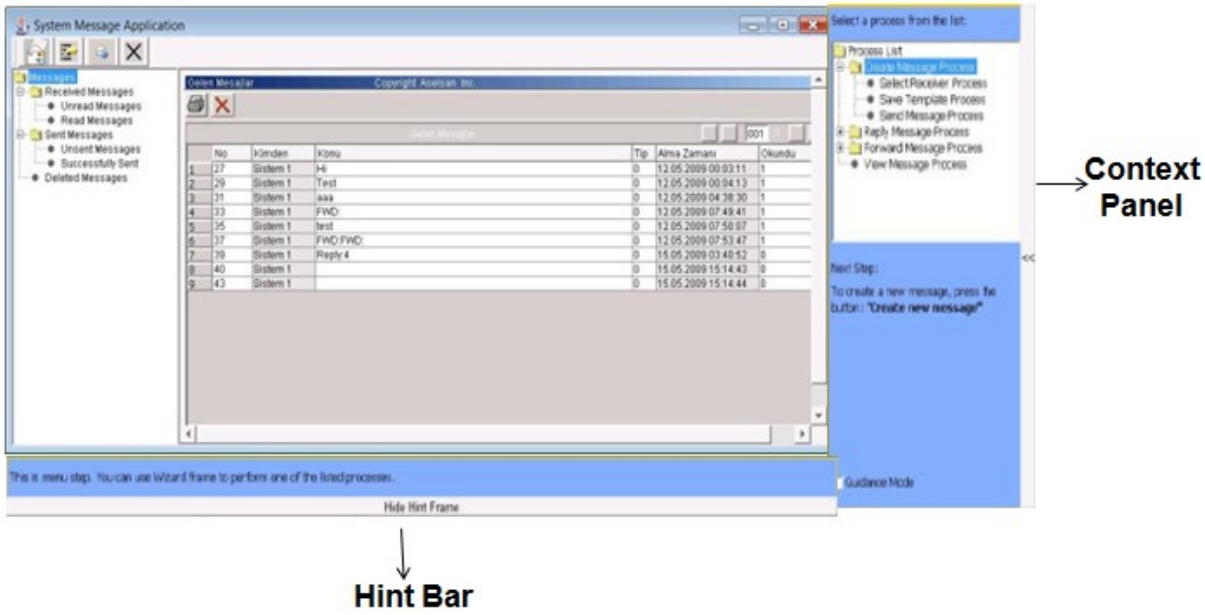

Figure 5.17: Assistant-Pro in action for System Message Application

Content panel lists the flow of work to be completed, and once the user selects one of the processes, the associated help content is revealed. In all cases, the original structure of the user interface of the target application is not ruined due to the execution of Assistant-Pro, meaning that the application is both modularly extended and enhanced with the guidance functions.

5.6 Cost Model for Evaluation

Version 1 of *Assistant-Pro* went through a formal evaluation process in Aselsan [1, 5]. It was ensured that *Assistant-Pro* operates in a cost-effective manner. Figure 5.18 highlights the components of cost model. The overall cost model adopted for the evaluation stage is defined as below:

$$C = C_{Learning} + C_{ContextAnalysis} + C_{ContextModeling} + C_{Help} + C_{AspectImplementation}$$

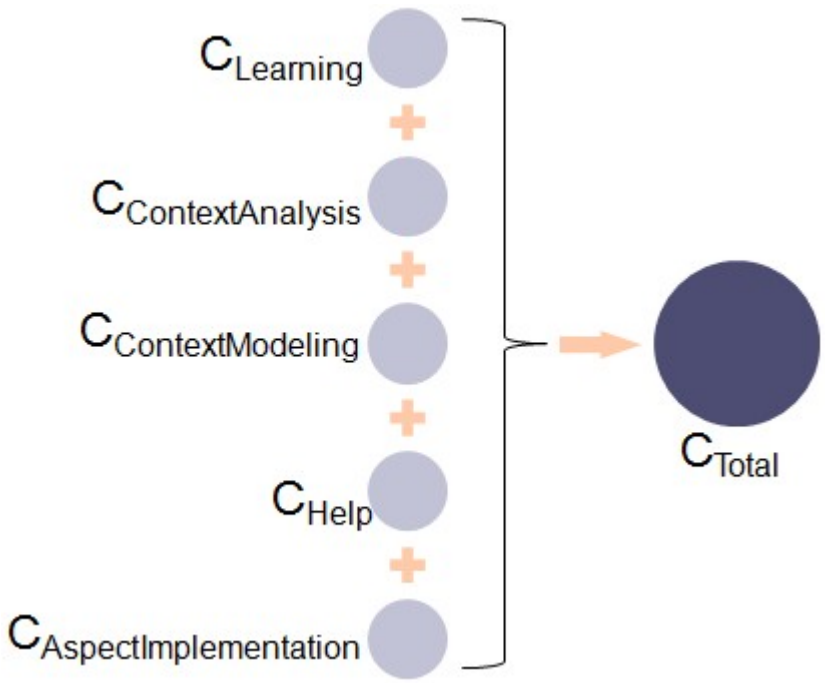

Figure 5.18: Cost Model Used for Evaluating Assistant-Pro

Five different cost elements are defined in this cost model. These elements are specified in response to the meetings with software development groups and senior professionals. We can define the elements as follows:

Cost of Learning $\Rightarrow$ training the individuals that will use *Assistant-Pro*.

Cost of Context Analysis $\Rightarrow$ defining the context related to the application.

Cost of Context Modeling $\Rightarrow$ modeling the context transitions.

Cost of Help $\Rightarrow$ help specifications.

Cost of Aspect Implementation $\Rightarrow$ implementing *Guidance Aspect*.

$C_{Learning}$ is a very little constituent of cost since using *Assistant-Pro* is intuitive with no complicated functions. Thus, users can easily learn the use of *Context Definition Tool* and *Help Definition Tool* though the user-friendly interfaces. $C_{ContextAnalysis}$ depends on the complexity of target applications to define associated context files. In any case, it will not lead to infeasible situations if the analysis and design of application is carefully done. $C_{ContextModeling}$ is directly proportional to $C_{ContextAnalysis}$ depending on the structure of application. C_{Help} is analogous to defining traditional user assistance documents such as PDF and HTML formats. $C_{AspectImplementation}$ is totally left to users' choice since it can be done either autonomously and manually. In most cases, manual definition will definitely take more time than automated generation.

All in all, the evaluation of both versions of *Assistant-Pro* has revealed very promising results in terms of the specified cost model. The adoption of *Assistant-Pro* only took a couple of hours to proceed. Also, it was observed in the evaluation of Version 2 that the less complex the target application is, the less the cost model wraps. This is declared due to the remarks that, in secondary evaluation stage, the target applications like a software project management tool have been seamlessly extended with *Assistant-Pro*.

Table 5.1 shows the evaluation results of both versions of *Assistant-Pro* based on the specified cost model.

Case	$C_{Learning}$	$C_{ContextAnalysis}$	$C_{ContextModeling}$	C_{Help}	$C_{AspectImplementation}$	TOTAL
Version 1						
MMS	24	2.5	1.5	2	2	32
LLVR	24	3	2	2.5	3	34.5
Version 2						
4PM	-	5	4	5	3.5	17.5
AO-Game	-	2	1.5	2.5	1.5	7.5

* Cost unit: hour

Cost determinants

Table 5.1: Evaluation results of the cost model

We can list the benefits of using *Assistant-Pro* as follows in a subjective way:

- User-friendly appearance
- Easier and even faultless development of user assistance
- Invigorating the communication between the stakeholders
- Enhancing reusability, modularity and extensibility
- Saving the effort required to read help manuals that could be of hundreds of papers
- Support for easier integration with legacy code
- Better user experience through better embedded user assistance

Chapter 6

Related Work

In our approach, we have adopted annotation-based aspect-oriented programming in order to explicitly declare the points that we would like to provide user assistance. In this way, we get expressive and robust pointcuts through the instrumentality of annotations. Alternatively, we may have selected joinpoints based on the signatures of language elements with respect to naming conventions or structural patterns, but the resultant pointcut descriptions would be more fragile, possibly targeting irrelevant joinpoints. Especially, Version 1 of *Assistant-Pro* aimed at mission-critical defense applications in which error margin should definitely be avoided. Thus, robust pointcuts are strictly required for accurate capturing of joinpoints.

The technologies used in our approach are stated as practicable in the studies [136,137] for interactive software designs. As the interaction between users and software applications is a well example for this case, we can observe the appropriateness of aspect-oriented and model-driven approaches for context-sensitive user assistance.

When we analyze the fields of interest that focus on process and context modeling, we observe several different domains. Method engineering [138, 139] and model-driven software development [140], especially meta-modeling [141], are some of the most popular areas in this field. *Situational method engineering*

which generally focuses on project-specific method construction is a specialized approach for process modeling. There have been some proposals in order to build *situational method engineering* on firm ground as follows:

OPEN Process Framework provides a flexible, disciplined, public domain framework for developing high-quality software intensive applications within a predictable schedule and budget [142, 143].

OMG's Software Process Engineering Metamodel (SPEM) is a metamodel proposal in response to the OMG RFP for software process engineering [130].

ISO/IEC 24744 is a standard for software engineering metamodeling for development methodologies [144, 145].

Our process modeling approach could be enhanced in response to these proposals, but we have followed a simpler approach using only a set of elements described within them in order to ease the use of *Assistant-Pro*. Also, context-sensitive modeling can be achieved by means of the specification of root processes in an application. Version 2 of *Assistant-Pro* has mostly focused on any type of context-sensitive help instead of guiding only the flow of processes.

The study [6] can be considered as an introductory summarization for embedded user assistance. The authors analyze some conventional methods for the provision of embedded user assistance, and they state the shortcomings of these methods. Also, they deliberate the issue that off-line documents do not suffice for effective user assistance. In this study, it is signalized that the individual embedded user assistance approaches do not actually work along with the systems they belong to. Instead, the approaches are conceptually undertaken as being embedded. Thus, we can make an inference that a proper embedded user assistance approach should be kept in the same level with the target applications at run-time.

The authors of study [18] emphasize that online user assistance should be

discussed at different levels of software applications, and also the need for modularizing user assistance concerns is revealed. In the proposed design, the users specify how and which levels that online user assistance should be provided without any user modeling task. Also, in this study, Kearsley's [146] model of dimensions for help systems, which includes the users task experience, program experience, and computer experience, is embodied. Inspired by these dimensions, the authors propose a three-dimensional model in which the outcomes of dimensions are rendered as functions for each and every user. Although this approach is practicable in conformity with its design, the authors do not focus on embedded user assistance, and besides, they do not explain how we can sample user assistance concerns directly from these models. The term *model* in this study and *Assistant-Pro*'s model concept are quite disparate, considering the technologies used in implementation. Our main objective is to model the process and/or the context of target applications, and by this way we can ultimately obtain platform-independent models on the basis of our tool framework.

The author of study [62] states that user experience and user assistance concerns should be integrated. User assistance elements are characterized under a template as herein defined, leading to consistent provision of user assistance across applications. The mentioned template is correspondingly defined to the concept of pattern language that Architect Christopher Alexander proposed as a guiding design principle [147]. This approach is rather conceptual without any explanation on how we can technically integrate user assistance into the software systems. Hypothetically speaking, this approach is supposed to resolve the communication issues between people concerned about user assistance and software developers. As was previously stated, *Assistant-Pro* involve four different user types as: code analyzer, context definer, user assistance specialist and end-user, and it includes tools as a matter of technical easiness on the basis of a solid framework.

The authors of study [29] focus on task and/or process modeling in user assistance. By analyzing the users operation on an application, some semantic concepts are gathered, which can enhance user assistance. As we have proposed the utilization of user modeling, this approach falls into a line with *Assistant-Pro*

in a sense.

Embedded user assistance affect the future and progress of software-intensive systems as stated in [59]. The authors mention the lack of tools and methods that are required to provide well-integration of user assistance concerns. Here, we can infer the complaints about specialized tool frameworks for context-sensitive user assistance, and *Assistant-Pro* comes into fray by providing a fine-grained infrastructure.

Web-based user assistance has also been studied as in study [79], and even in this field, there are some integration problems despite effective Web technologies. The authors speak about detailed user assistance that will help users to attain their objectives through revising interface designs. It has been tested that embedded user assistance definitely improves productivity.

The study [34] undertakes model-driven user modeling in an effort to provide customized user assistance. However, modeling only the users is not necessarily enough to seamlessly integrate user assistance content into the applications. We need to perform a cross-modeling approach involving process, context, user and interest.

Embedded user assistance is not seen as a supporting element, but it appears as a functionality for the users instead [85]. Users generally do not receive separate user assistance mechanisms favorably just because the flow of work is ruined owing to this separate effort to find appropriate guidance. In our approach, we strongly consider the continuity and stability of work-flows by means of a context-sensitive solution.

The study [148] mention about an approach based on *finite state machines* for the purpose of user assistance in interactive systems. This approach basically aims at providing associated user assistance in response to state transitions. However, the authors complain about the lack of programming structures that correspond to the abstraction of states in user assistance. In our approach, the aspect-oriented design has been thought of as a solution to this problem, which extracts all user assistance concerns into an aspect.

Context-sensitive user assistance has been proven to be more effective than stand-alone solutions. The proposals in the study [78] are all attached to applications, so to say, as patchworks. However, in our approach we consider user assistance as a major independent concern, considering reusability and maintainability issues. Thus, we have followed an aspect-oriented approach that includes modeling of concepts.

Chapter 7

Conclusion

The domain of user assistance is increasingly growing, having evolved in many real-life examples. In this report, we have focused on revealing the state-of-the-art advances in user assistance. To analyze the background and acquire a solid insight in user assistance systems, we have carried out two distinct systematic reviews. Firstly, we have started with analyzing automated user assistance which is a general category of almost all user assistance solutions, involving embedded guidance. Secondly, we have gone down one-level to embedded user assistance which directly fits into our main interest. In total 1125 papers were analyzed by conducting systematic literature reviews for both cases, and 54 of them were selected as primary studies that are more related to context-sensitive user assistance.

We have used the results and insight from the systematic literature reviews and the survey of help authoring tools into the first version of our tool framework *Assistant-Pro* along with some well-reasoned enhancements. Since developing a tool framework should be of a general purpose way, we have somehow simplified the process of developing context-sensitive user assistance. It appeared that several concerns of user assistance systems cannot be easily localized into separate models. In particular the concerns of help and flow of control tend to be scattered over the system. Because of the crosscutting property, developing help systems

is rather difficult. Aspect-oriented software development is the leading technology that we have adopted in the implementation. This technology enables us to modularize the crosscutting user assistance concerns into aspects which promotes reusability across multiple applications.

The tool framework was first intended for process-sensitive user assistance; however, with the development of Version 2, we see that it can easily be adapted for any context-sensitive user assistance system. Creating the context files and specifying related help content are the major operations that need to be done seamlessly in any case. The problems of conventional or manual approach in developing user assistance such as faulty behavior and inconsistent results necessitate the adoption of automated mechanisms like a dedicated tool.

We have shown the usefulness of tool framework in several examples along with some noteworthy benefits like easier and even faultless development of user assistance. Also, the adopted cost model to evaluate *Assistant-Pro* has shown very encouraging results. Compared to traditional user assistance, embedded context-sensitive user assistance is worth employing according to the effort it takes to adopt.

The systematic literature reviews have also provided the important open research problems. We have addressed some of these important research problems. Additionally, we have discovered the research directions to be followed in the evolution of user assistance solutions. The strength of evidence is not high in respect of our comprehensive systematic reviews, meaning that there should be further studies, specifically experiments, in this field. As a future work, we think of several potential movements. First, the tool framework would be deployed in several industrial settings. Also, we can directly adopt model-driven software development techniques to present a more domain-independent tool framework. Finally, *Assistant-Pro* would be turn into a completely stand-alone application which could be integrated with several development environments.

Bibliography

[1] B. Tekinerdogan, S. Bozbey, Y. Mester, E. Turanciftci, and L. Alkislar, "An aspect-oriented tool framework for developing process-sensitive embedded user assistance systems," in *Transactions on Aspect-Oriented Software Development VIII* (S. Katz, M. Mezini, C. Schwanninger, and W. Joosen, eds.), vol. 6580 of *Lecture Notes in Computer Science*, pp. 196–220, Springer Berlin/Heidelberg, 2011.

[2] Interface Design for Computer-based Learning Environments, `http://www2.gsu.edu/~wwwitr/docs/idguide/`. Accessed June 12, 2012.

[3] WebHelp, `http://www.knopf.com/resources/help/webhelp.html`. Accessed June 12, 2012.

[4] Designing Embedded Help, `http://www.clickstart.net/presentations/embedded.htm`. Accessed June 12, 2012.

[5] Aselsan. `http://www.aselsan.com/`. Accessed July 11, 2012.

[6] A. L. Ames, "Just what they need, just when they need it: an introduction to embedded assistance," in *Proceedings of the 19th annual international conference on Computer documentation*, SIGDOC '01, (New York, NY, USA), pp. 111–115, ACM, 2001.

[7] S. Delisle and B. Moulin, "User interfaces and help systems: From helplessness to intelligent assistance," *Artificial Intelligence Review*, vol. 18, pp. 117–157, 2002. 10.1023/A:1015179704819.

[8] B. A. Kitchenham, T. Dyba, and M. Jorgensen, "Evidence-based software engineering," in *Proceedings of the 26th International Conference on Software Engineering*, ICSE '04, (Washington, DC, USA), pp. 273–281, IEEE Computer Society, 2004.

[9] T. Dyba, B. A. Kitchenham, and M. Jorgensen, "Evidence-based software engineering for practitioners," *IEEE Softw.*, vol. 22, pp. 58–65, Jan. 2005.

[10] M. S. Ali, M. Ali Babar, L. Chen, and K.-J. Stol, "A systematic review of comparative evidence of aspect-oriented programming," *Inf. Softw. Technol.*, vol. 52, pp. 871–887, Sept. 2010.

[11] V. B. Kampenes, T. Dyb, J. E. Hannay, and D. I. Sjberg, "A systematic review of effect size in software engineering experiments," *Information and Software Technology*, vol. 49, no. 1112, pp. 1073 – 1086, 2007.

[12] W. Afzal, R. Torkar, and R. Feldt, "A systematic review of search-based testing for non-functional system properties," *Inf. Softw. Technol.*, vol. 51, pp. 957–976, June 2009.

[13] S. Ali, L. Briand, H. Hemmati, and R. Panesar-Walawege, "A systematic review of the application and empirical investigation of search-based test case generation," *Software Engineering, IEEE Transactions on*, vol. 36, pp. 742 –762, nov.-dec. 2010.

[14] D. Šmite, C. Wohlin, T. Gorschek, and R. Feldt, "Empirical evidence in global software engineering: a systematic review," *Empirical Softw. Engg.*, vol. 15, pp. 91–118, Feb. 2010.

[15] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Inf. Softw. Technol.*, vol. 50, pp. 833–859, Aug. 2008.

[16] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering - a systematic literature review," *Inf. Softw. Technol.*, vol. 51, pp. 7–15, Jan. 2009.

[17] V. Alvarez-Cortes, B. Zayas-Perez, V. Zarate-Silva, and J. Uresti, "Current trends in adaptive user interfaces: Challenges and applications," in *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pp. 312 –317, sept. 2007.

[18] O. D. Andrade and D. G. Novick, "Expressing help at appropriate levels," in *Proceedings of the 26th annual ACM international conference on Design of communication*, SIGDOC '08, (New York, NY, USA), pp. 125–130, ACM, 2008.

[19] N. K. Sondheimer and N. Relles, "Human factors and user assistance in interactive computing systems: An introduction," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 12, pp. 102 –107, march 1982.

[20] B. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," Tech. Rep. EBSE 2007-001, Keele University and Durham University Joint Report, 2007.

[21] P. Tell and M. A. Babar, "An evaluation of systematic search approaches for selecting software engineering relevant papers," tech. rep., 2010.

[22] H. Zhang and M. Ali Babar, "On searching relevant studies in software engineering," in *Proceedings of the 14th international conference on Evaluation and Assessment in Software Engineering*, EASE'10, (Swinton, UK, UK), pp. 111–120, British Computer Society, 2010.

[23] O. Dieste and A. G. Padua, "Developing search strategies for detecting relevant experiments for systematic reviews," in *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement*, ESEM '07, (Washington, DC, USA), pp. 215–224, IEEE Computer Society, 2007.

[24] WordStat, `http://www.provalisresearch.com/wordstat/Wordstat.html`. Accessed June 7, 2012.

[25] SimStat, `http://www.provalisresearch.com/simstat/simstw.html`. Accessed June 7, 2012.

[26] Provalis Research, `http://www.provalisresearch.com/`. Accessed June 5, 2012.

[27] O. D. Andrade and D. G. Novick, "Expressing help at appropriate levels," in *Proceedings of the 26th annual ACM international conference on Design of communication*, SIGDOC '08, (New York, NY, USA), pp. 125–130, ACM, 2008.

[28] V. Alvarez-Cortes, B. Zayas-Perez, V. Zarate-Silva, and J. Uresti, "Current trends in adaptive user interfaces: Challenges and applications," in *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pp. 312 –317, sept. 2007.

[29] A. E. Kohlhase and M. Kohlhase, "Modeling task experience in user assistance systems," in *Proceedings of the 27th ACM international conference on Design of communication*, SIGDOC '09, (New York, NY, USA), pp. 135–142, ACM, 2009.

[30] A. E. Kohlhase and M. Kohlhase, "Semantic transparency in user assistance systems," in *Proceedings of the 27th ACM international conference on Design of communication*, SIGDOC '09, (New York, NY, USA), pp. 89–96, ACM, 2009.

[31] S. Biundo, P. Bercher, T. Geier, F. Mller, and B. Schattenberg, "Advanced user assistance based on ai planning," *Cognitive Systems Research*, vol. 12, no. 34, pp. 219 – 236, 2011. ¡ce:title¿Special Issue on Complex Cognition¡/ce:title¿.

[32] A. D. Stefano, G. Pappalardo, C. Santoro, and E. Tramontana, "A multi-agent reflective architecture for user assistance and its application to e-commerce," in *Proceedings of the 6th International Workshop on Cooperative Information Agents VI*, CIA '02, (London, UK, UK), pp. 90–103, Springer-Verlag, 2002.

[33] H. Wandke *, "Assistance in humanmachine interaction: a conceptual framework and a proposal for a taxonomy," *Theoretical Issues in Ergonomics Science*, vol. 6, no. 2, pp. 129–155, 2005.

[34] J. Fischer, *On the Development of Personalized User Agents: A Model-driven Approach*. Dissertation, Departamento de Informtica, Universidade Rior De Janerio, 2010.

[35] M. Virvou and K. Kabassi, "Reasoning about users' actions in a graphical user interface," *HumanComputer Interaction*, vol. 17, no. 4, pp. 369–398, 2002.

[36] R. Zilz and P. Forbrig, "Smart user assistance based on dynamic model composition," in *Design, User Experience, and Usability. Theory, Methods, Tools and Practice* (A. Marcus, ed.), vol. 6769 of *Lecture Notes in Computer Science*, pp. 706–714, Springer Berlin / Heidelberg, 2011.

[37] D. Gorecky, S. F. Worgan, and G. Meixner, "Cognito: a cognitive assistance and training system for manual tasks in industry," in *Proceedings of the 29th Annual European Conference on Cognitive Ergonomics*, ECCE '11, (New York, NY, USA), pp. 53–56, ACM, 2011.

[38] Y. Zhang, H. Ghenniwa, and W. Shen, "Enhancing intelligent user assistance in collaborative design environments," in *Computer Supported Cooperative Work in Design, 2005. Proceedings of the Ninth International Conference on*, vol. 1, pp. 107 – 112 Vol. 1, may 2005.

[39] S. Wu, H. Ghenniwa, W. Shen, and K. Ma, "Intelligent user assistance in collaborative design environments," in *Computer Supported Cooperative Work in Design, 2004. Proceedings. The 8th International Conference on*, vol. 2, pp. 259 – 266 Vol.2, may 2004.

[40] J. Harkin, M. Callaghan, T. McGinnity, and L. Maguire, "Intelligent user-support in learning environments for remote experimentation," in *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on*, vol. 2, pp. 119 –124, july 2005.

[41] A. Steiniger and A. Uhrmacher, "Modeling and simulation for user assistance in smart environments," in *Simulation Conference (WSC), Proceedings of the 2010 Winter*, pp. 490 –499, dec. 2010.

[42] M. J. Callaghan, J. Harkin, T. M. McGinnity, and L. P. Maguire, "Adaptive intelligent environment for remote experimentation," in *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, (Washington, DC, USA), pp. 680–, IEEE Computer Society, 2003.

[43] S. Wu, H. Ghenniwa, Y. Zhang, and W. Shen, "Personal assistant agents for collaborative design environments," *Comput. Ind.*, vol. 57, pp. 732–739, Dec. 2006.

[44] X. Li and Q. Ji, "Active affective state detection and user assistance with dynamic bayesian networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, pp. 93 – 105, jan. 2005.

[45] W. Liao, W. Zhang, Z. Zhu, and Q. Ji, "A decision theoretic model for stress recognition and user assitance," in *Twentieth National Conference on Artifical Intelligence (AAAI*, pp. 529–534, 2005.

[46] V. Loia, W. Pedrycz, S. Senatore, and M. I. Sessa, "Web navigation support by means of proximity-driven assistant agents: Special topic section on soft approaches to information retrieval and information access on the web," *J. Am. Soc. Inf. Sci. Technol.*, vol. 57, pp. 515–527, Feb. 2006.

[47] M. Virvou and K. Kabassi, "Evaluating an intelligent graphical user interface by comparison with human experts," *Knowledge-Based Systems*, vol. 17, no. 1, pp. 31 – 37, 2004.

[48] B. J. Jansen, "Seeking and implementing automated assistance during the search process," *Information Processing and Management*, vol. 41, no. 4, pp. 909–928, 2005.

[49] D. Godoy and A. Amandi, "Exploiting user interests to characterize navigational patterns in web browsing assistance," *New Generation Computing*, vol. 26, pp. 259–275, 2008. 10.1007/s00354-008-0044-x.

[50] B. J. Jansen and M. D. McNeese, "Evaluating the effectiveness of and patterns of interactions with automated searching assistance: Research articles," *J. Am. Soc. Inf. Sci. Technol.*, vol. 56, pp. 1480–1503, Dec. 2005.

[51] A. Casamayor, A. Amandi, and M. Campo, "Intelligent assistance for teachers in collaborative e-learning environments," *Computers and Education*, vol. 53, no. 4, pp. 1147–1154, 2009. ¡ce:title¿Learning with ICT: New perspectives on help seeking and information searching¡/ce:title¿.

[52] J. A. Macías, "Intelligent assistance in authoring dynamically generated web interfaces," *World Wide Web*, vol. 11, pp. 253–286, June 2008.

[53] D. P, A. Bhamidipaty, and S. Challa, "Intelligent user assistance for cost effective usage of mobile phone," in *Proceedings of the 13th international conference on Intelligent user interfaces*, IUI '08, (New York, NY, USA), pp. 317–320, ACM, 2008.

[54] J. Liu, C. K. Wong, and K. K. Hui, "An adaptive user interface based on personalized learning," *Intelligent Systems, IEEE*, vol. 18, pp. 52 – 57, mar-apr 2003.

[55] B. A. Forsyth and K. E. MacLean, "Predictive haptic guidance: Intelligent user assistance for the control of dynamic tasks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, pp. 103–113, 2006.

[56] Y. Zhang, H. Ghenniwa, and W. Shen, "Agent-based personal assistance in collaborative design environments," in *Proceedings of the 9th international conference on Computer Supported Cooperative Work in Design II*, CSCWD'05, (Berlin, Heidelberg), pp. 284–293, Springer-Verlag, 2006.

[57] G. Noblit and R. Hare, *Meta-Ethnography: Synthesizing Qualitative Studies*. Qualitative Research Methods, Sage Publications, 1988.

[58] D. Atkins, D. Best, P. A. Briss, M. Eccles, Y. Falck-Ytter, S. Flottorp, G. H. Guyatt, R. T. Harbour, M. C. Haugh, D. Henry, S. Hill, R. Jaeschke, G. Leng, A. Liberati, N. Magrini, J. Mason, P. Middleton, J. Mrukowicz, D. O'Connell, A. D. Oxman, B. Phillips, H. J. Schünemann, T. T.-T. T. Edejer, H. Varonen, G. E. Vist, J. W. Williams, S. Zaza, and GRADE Working Group, "Grading quality of evidence and strength of recommendations.," *BMJ (Clinical research ed.)*, vol. 328, June 2004.

[59] M. Ellison, "Embedded user assistance: the future for software help?," *interactions*, vol. 14, pp. 30–31, Jan. 2007.

[60] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement:the kappa statistic," *Family Medicine*, vol. 37, no. 5, pp. 360–3, 2005.

[61] A. P. Martin, M. Y. Ivory, R. Megraw, and B. Slabosky, "Exploring the persistent problem of user assistance," tech. rep., Information School, University of Washington, Washington, August 2005.

[62] M. Hughes, "A pattern language for user assistance," *interactions*, vol. 14, pp. 27–29, Jan. 2007.

[63] H. Lang, C. Mosch, B. Boegel, D. Benoit, and W. Minker, "An avatar-based help system for web-portals," in *Human-Computer Interaction. Interaction Techniques and Environments* (J. Jacko, ed.), vol. 6762 of *Lecture Notes in Computer Science*, pp. 537–546, Springer Berlin / Heidelberg, 2011.

[64] A. Bahrami, J. Yuan, P. R. Smart, and N. R. Shadbolt, "Context aware information retrieval for enhanced situation awareness," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*, pp. 1 –6, oct. 2007.

[65] K. L. Halsted and J. H. J. Roberts, *Eclipse Help System : An Open Source User Assistance Offering*, pp. 49–59. ACM Press, 2002.

[66] A. Kehoe, F. Neff, I. Pitt, and G. Russell, "Improvements to a speech-enabled user assistance system based on pilot study results," in *Proceedings of the 25th annual ACM international conference on Design of communication*, SIGDOC '07, (New York, NY, USA), pp. 42–47, ACM, 2007.

[67] Y. Aoki, M. Shinozaki, and A. Nakajima, "Interactive web forms based on assistance rules," in *Systems, Man and Cybernetics, 2002 IEEE International Conference on*, vol. 7, p. 8 pp. vol.7, oct. 2002.

[68] Z. Jrad and M.-A. Aufaure, "Personalized interfaces for a semantic web portal: Tourism information search," in *Knowledge-Based Intelligent Information and Engineering Systems* (B. Apolloni, R. Howlett, and L. Jain,

eds.), vol. 4694 of *Lecture Notes in Computer Science*, pp. 695–702, Springer Berlin / Heidelberg, 2007.

[69] M. Schneider, M. Velten, and J. Haupert, "The objectrules framework - providing ad hoc context-dependent assistance in dynamic environments," in *Intelligent Environments (IE), 2010 Sixth International Conference on*, pp. 122 –127, july 2010.

[70] R. Ponnusamy and T. Gopal, "A user-adaptive self-proclamative multi-agent based recommendation system design for e-learning digital libraries," in *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pp. 1 –7, june 2006.

[71] P. Hsuan, C.-R. Dow, K.-H. Chen, Y.-Y. Chen, and Y.-H. Li, "An ubiquitous teaching assistant using knowledge retrieval and adaptive learning techniques," in *Integration of Knowledge Intensive Multi-Agent Systems, 2007. KIMAS 2007. International Conference on*, pp. 121 –126, 30 2007-may 3 2007.

[72] P. Lehsten, A. Gladisch, and D. Tavangarian, "Context-aware integration of smart environments in legacy applications," in *Ambient Intelligence* (D. Keyson, M. Maher, N. Streitz, A. Cheok, J. Augusto, R. Wichert, G. Englebienne, H. Aghajan, and B. Krse, eds.), vol. 7040 of *Lecture Notes in Computer Science*, pp. 126–135, Springer Berlin / Heidelberg, 2011.

[73] J. Bergasa-Suso, D. Sanders, and G. Tewkesbury, "Intelligent browser-based systems to assist internet users," *Education, IEEE Transactions on*, vol. 48, pp. 580 – 585, nov. 2005.

[74] A. Kohlhase, "Towards user assistance for documents via interactional semantic technology," in *KI 2010: Advances in Artificial Intelligence* (R. Dillmann, J. Beyerer, U. Hanebeck, and T. Schultz, eds.), vol. 6359 of *Lecture Notes in Computer Science*, pp. 107–115, Springer Berlin / Heidelberg, 2010.

[75] M. Virvou and K. Kabassi, "Reasoning About Users' Actions in a Graphical User Interface," *Human-computer Interaction*, vol. 17, pp. 369–398, 2002.

[76] B. Tekinerdogan, S. Bozbey, Y. Mester, E. Turanciftci, and L. Alkislar, "An aspect-oriented tool framework for developing process-sensitive embedded user assistance systems," in *Transactions on Aspect-Oriented Software Development VIII* (S. Katz, M. Mezini, C. Schwanninger, and W. Joosen, eds.), vol. 6580 of *Lecture Notes in Computer Science*, pp. 196–220, Springer Berlin / Heidelberg, 2011.

[77] S. DeLoach, "Designing, localizing, and customizing web-based embedded assistance," in *Professional Communication Conference, 2005. IPCC 2005. Proceedings. International*, pp. 176 – 180, july 2005.

[78] M. Ellison, "User-centred design of context-sensitive help," in *Australasian Online Documentation and Content Conference*, 2009.

[79] T. Grayling, "If we build it, will they come? a usability test of two browser-based embedded help systems," *Technical Communication*, vol. 49, no. 2, pp. 193–209, May.

[80] F. Ali, S. W. Lee, Z. Bien, and M. Mokhtari, "Combined fuzzy state q-learning algorithm to predict context aware user activity under uncertainty in assistive environment," in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on*, pp. 57 –62, aug. 2008.

[81] H. Zhang, C. Schreiner, K. Zhang, and K. Torkkola, "Naturalistic use of cell phones in driving and context-based user assistance," in *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services*, MobileHCI '07, (New York, NY, USA), pp. 273–276, ACM, 2007.

[82] D. Godoy and A. Amandi, "Exploiting user interests to characterize navigational patterns in web browsing assistance," *New Generation Computing*, vol. 26, pp. 259–275, 2008.

[83] E. N. Webb, R. Matsil, and J. Sauro, "Benefit analysis of user assistance improvements," in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, (New York, NY, USA), pp. 841–850, ACM, 2011.

[84] D. Lammers, "Context sensitive help concepts," 2000.

[85] S. DeLoach, "Best practices for embedded user assistance," 2007.

[86] helpndoc, "Overview of help authoring tools," Accessed July 16, 2012.

[87] M. Ellison, "Help authoring tool comparison," in *Australasian Online Documentation and Content Conference*, 2010.

[88] HubPages, "Top 10 commercial help authoring tools," Accessed July 16, 2012.

[89] indoition Information Development, "Choosing a help authoring tool," Accessed July 16, 2012.

[90] indoition Information Development, "Help authoring tools," Accessed July 16, 2012.

[91] Flare - `http://www.madcapsoftware.com`. Accessed July 16, 2012.

[92] Help & Manual - `http://www.helpandmanual.com`. Accessed July 16, 2012.

[93] RoboHelp - `http://www.adobe.com/products/robohelp/`. Accessed July 16, 2012.

[94] HelpStudio - `http://www.innovasys.com`. Accessed July 16, 2012.

[95] Fast-Help - `http://www.fast-help.com`. Accessed July 16, 2012.

[96] HelpSmith - `http://www.helpsmith.com`. Accessed July 16, 2012.

[97] Author-IT - `http://www.author-it.com`. Accessed July 16, 2012.

[98] Doc-To-Help - `http://www.componentone.com`. Accessed July 16, 2012.

[99] Help Producer - `http://www.mgtek.com`. Accessed July 16, 2012.

[100] OfficeHelp - http://www.officehelp.de. Accessed July 16, 2012.

[101] West Wind HTML Help Builder - http://www.west-wind.com. Accessed July 16, 2012.

[102] Doc-O-Matic - http://www.doc-o-matic.com. Accessed July 16, 2012.

[103] Document! X - http://www.innovasys.com. Accessed July 16, 2012.

[104] TeeGofer Help Author for .NET Components - http://www.steema.com. Accessed July 16, 2012.

[105] VSdocman, VBdocman - http://www.helixoft.com. Accessed July 16, 2012.

[106] GenHelp - http://www.frasersoft.net. Accessed July 16, 2012.

[107] QuickHelp - http://www.excelsoftware.com. Accessed July 16, 2012.

[108] HelpLogic - http://www.ebutterfly.com/helplogic/. Accessed July 16, 2012.

[109] HelpBlocks - http://www.helpblocks.com. Accessed July 16, 2012.

[110] Helen - http://www.software7.biz. Accessed July 16, 2012.

[111] ScreenSteps Desktop - http://www.bluemangolearning.com. Accessed July 16, 2012.

[112] HelpBurner - http://www.helpburner.com. Accessed July 16, 2012.

[113] Help Generator - http://www.helpgenerator.com. Accessed July 16, 2012.

[114] Dr. Explain (Cognitive Force) - http://www.drexplain.com. Accessed July 16, 2012.

[115] TechWriter - http://www.adivo.com. Accessed July 16, 2012.

[116] HelpConsole - http://www.extremeease.com. Accessed July 16, 2012.

[117] HelpIQ - http://www.helpiq.com. Accessed July 16, 2012.

[118] HelpServer - http://www.helpserver.eu. Accessed July 16, 2012.

[119] exxDoc - http://www.pintexx.com. Accessed July 16, 2012.

[120] tomeCMS - http://www.tomecms.org. Accessed July 16, 2012.

[121] HelpMaker - http://www.vizacc.com. Accessed July 16, 2012.

[122] HelpSetMaker - http://www.cantamen.de/helpsetmaker.php?lang=en. Accessed July 16, 2012.

[123] HelpNDoc - http://www.helpndoc.com. Accessed July 16, 2012.

[124] C. James-Tanny, "Choosing a help authoring tool," JTF Associates, Inc., 2004.

[125] C. James-Tanny, "Choosing help authoring tools:what factors affect your decision?," Boston Broadside:January/February issue, 2003.

[126] C. James-Tanny, "Choosing help authoring tools:what are your choices?," Boston Broadside:November/December issue, 2002.

[127] indoition Information Development, "Help authoring tool (hat) comparison matrix," Accessed July 16, 2012.

[128] Fast-Help. http://fast-help.com/. Accessed July 13, 2012.

[129] WritersUA - Tools, http://www.writersua.com/restools.htm. Accessed July 13, 2012.

[130] Software&Systems Process Engineering Meta-Model Specification - Version 2.0, http://www.omg.org/spec/SPEM/2.0/PDF/. Accessed July 01, 2012.

[131] Eclipse - AspectJ, http://www.eclipse.org/aspectj/. Accessed June 25, 2012.

[132] Annotation Processing Tool(apt), http://docs.oracle.com/javase/1.5.0/docs/guide/apt/index.html. Accessed June 28, 2012.

[133] Mirror API, http://docs.oracle.com/javase/1.5.0/docs/guide/apt/mirror/overview-summary.html. Accessed June 28, 2012.

[134] Annotation File Utilities, `http://types.cs.washington.edu/annotation-file-utilities/`. Accessed June 29, 2012.

[135] F. Dantas, T. Batista, N. Cacho, and A. Garcia, "Towards aspect-oriented programming for context-aware systems: A comparative study," in *Proceedings of the 29th International Conference on Software Engineering Workshops*, ICSEW '07, (Washington, DC, USA), pp. 190–, IEEE Computer Society, 2007.

[136] R. E. Filman and D. P. Friedman, "Aspect-oriented programming is quantification and obliviousness," tech. rep., 2000.

[137] F. Paterno, "Model-based design of interactive applications," *Intelligence*, vol. 11, pp. 26–38, Dec. 2000.

[138] B. Henderson-Sellers, "Method engineering for oo systems development," *Commun. ACM*, vol. 46, pp. 73–78, Oct. 2003.

[139] C. Rolland and N. Prakash, "A proposal for context-specific method engineering," in *Proceedings of the IFIP TC8, WG8.1/8.2 working conference on method engineering on Method engineering : principles of method construction and tool support: principles of method construction and tool support*, (London, UK, UK), pp. 191–208, Chapman & Hall, Ltd., 1996.

[140] M. Voelter and I. Groher, "Product line implementation using aspect-oriented and model-driven software development," in *Software Product Line Conference, 2007. SPLC 2007. 11th International*, pp. 233 –242, sept. 2007.

[141] C. Atkinson and T. Kühne, "Model-driven development: A metamodeling foundation," *IEEE Softw.*, vol. 20, pp. 36–41, Sept. 2003.

[142] B. Henderson-Sellers, "Process metamodelling and process construction: Examples using the open process framework (opf)," *Annals of Software Engineering*, vol. 14, pp. 341–362, 2002. 10.1023/A:1020570027891.

[143] D. Firesmith and B. Henderson-Sellers, *The Open Process Framework: An Introduction*. Open Series, Addison-Wesley, 2002.

[144] *Iso/iec 24744:2007 Software Engineering - Metamodel for Development Methodologies*. Iso, 2007.

[145] C. Gonzalez-Perez, "Supporting situational method engineering with iso/iec 24744 and the work product pool approach.," in *Situational Method Engineering*, vol. 244 of *IFIP*, pp. 7–18, Springer, 2007.

[146] G. Kearsley, *Online help systems: design and implementation*. Norwood, NJ, USA: Ablex Publishing Corp., 1988.

[147] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*. Boston: Addison-Wesley, 1977.

[148] H. Thimbleby and M. Addison, "Intelligent adaptive assistance and its automatic generation," *Interacting with Computers*, vol. 8, no. 1, pp. 51 – 68, 1996.

Appendix A

Output from Systematic Reviews

Repository	String
ACM	*Title*:("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive") *Abstract*:("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive") *Keywords*:("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive")
IEEE	("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive")
ISI Web of Knowledge	TS=(("user assistance") AND ("automated" OR "context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive"))
Science Direct	*Abstract, Title, Keywords*: ("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive")
Springer	*Title, Abstract*: ("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive")
Wiley Interscience	*Publication Titles, Article Titles, Abstract, Keywords*: ("user" AND "assistance") AND ("context sensitive" OR "context-sensitive" OR " process-sensitive" OR "process sensitive" OR "context aware" OR "context-aware" OR "embedded" OR "intelligent" OR "adaptive")

Figure A.1: Search Strings for Automated User Assistance Systems

Study description	Extraction element	Contents
General Information		
1	ID	Unique id for the study
2	SLR Category	○ Include ○ Exclude
3	Title	Full title of the article
4	Date of Extraction	The date it is added into repository
5	Year	The publication year
6	Authors	
7	Repository	ACM, IEEE, ISI Web of Knowledge, ScienceDirect, Springer, Wiley Interscience
8	Type	○ Journal ○ Conference ○ Other (dissertation, grey literature etc.)
9	Does it repeat already reviewed paper?	○ Yes (Repeated ID) ○ No
Relevance		
10	Does it relate to a specific field of computer science?	○ Yes ○ No ○ To some extent
11	Does it relate to user assistance?	○ Yes ○ No ○ To some extent
12	Subjects	○ Academics ○ Industry/Real world
Study Description		
10	Main theme of the study	
11	Motivation for the main theme	
12	Study aims	
13	Targeted domain	Task-specific environments, Adaptive, multi-layer and multi-dimensional user interfaces, Collaborative Environments, Interactive systems, Information Retrieval Systems, Agent-based environments, Mobile devices, Learning Environments for Remote Experimentation
14	Automated user assistance solution used	
15	Examples of application of solution	Critical, Web-based, Portable appliances, Non-functional concerns, Real-world, Computer-aided
16	Research method used	Case study, Multiple-case study, Experiment, Benchmarking, , Survey
17	Assessment approach	Qualitative, Quantitative or Both
18	Findings	
19	Constraints/limitations	
20	Implications for future research	
21	Major conclusions	
Evaluation		
22	Personal note	The opinions of the reviewer about the study
23	Additional note	Publication details (supported by grants etc.)
24	Quality Assessment	Detailed quality scores

Figure A.2: Data Extraction Form for Automated User Assistance Systems

	Reporting				Relevance	Rigor			Credibility		
	Aim	Scope, Context and Design	Evaluation Rationale	Description of study participants	Implications in practice and research	Validity and reliability of variables	Explicitness of measures	Adequacy of reporting	Creditability, validity and reliability	Limitations	
Primary Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	**TOTAL** (out of 10)
[53]	1	0.5	1	1	0.5	1	0.5	1	0.5	0.5	7.5
[54]	1	1	1	0.5	0.5	0.5	1	1	1	0.5	8
[9]	1	1	0.5	1	1	0.5	0	0	0.5	0	5.5
[48]	1	1	1	0.5	1	0.5	0.5	0.5	1	0	7
[49]	1	1	1	0.5	1	0.5	0.5	0.5	0.5	0.5	7
[56]	1	1	1	0.5	0.5	0.5	1	1	0.5	0	7
[17]	1	1	0.5	1	0.5	0.5	0.5	1	0.5	0.5	7
[73]	1	1	1	0.5	1	0.5	1	1	0.5	0	7.5
[40]	1	1	1	1	0.5	1	0.5	1	0.5	0.5	8
[66]	1	1	0.5	0.5	1	0	0.5	1	1	0.5	7
[32]	1	1	1	1	1	1	1	1	0.5	0	8.5
[41]	1	1	1	1	1	0.5	0.5	1	1	0	8
[19]	1	1	1	0.5	1	1	0.5	1	0.5	0	7.5
[75]	1	0.5	0.5	1	1	0	1	0.5	0.5	0	6
[30]	1	1	1	1	1	1	0.5	0.5	0	0.5	7.5
[71]	1	1	1	1	1	0.5	0.5	1	1	0.5	8.5
[83]	1	0.5	0.5	1	0.5	0.5	0.5	0.5	0.5	0	5.5
[47]	1	1	1	0.5	1	1	0.5	1	1	0.5	8.5
[84]	1	1	1	0.5	0.5	0.5	0.5	0.5	0	0.5	6
[79]	1	1	1	1	0.5	0.5	0.5	0.5	0.5	0	6.5
[34]	1	0.5	0	1	0.5	0	0.5	0	0.5	0	4
[61]	1	0.5	0.5	0.5	1	0.5	0	0.5	0.5	0	5
[55]	1	0.5	1	1	0.5	0.5	1	0.5	0	0.5	6.5
[6]	1	1	1	0.5	1	0	0.5	0.5	1	0.5	7
[82]	1	1	1	0.5	1	0	0.5	0.5	0.5	0.5	6.5
[78]	1	1	0.5	0.5	1	1	1	1	0.5	0.5	8
[37]	1	1	1	1	1	0.5	0.5	0.5	0.5	0.5	7.5
[67]	1	1	1	0.5	1	0.5	0.5	0.5	0.5	0.5	7
[31]	1	1	0.5	1	0.5	0.5	0.5	0.5	0.5	0.5	6.5
[18]	1	0.5	0	1	0.5	0	0.5	0.5	0.5	0	4.5

Figure A.3: Study Quality Assessment of Automated User Assistance

Study description	Extraction element	Contents
General Information		
1	ID	Unique id for the study
2	SLR Category	+Include -Exclude
3	Title	Full title of the article
4	Date of Extraction	The date it is added into our repository
5	Year	The publication year
6	Authors	
7	Repository	ACM, IEEE, ISI Web of Knowledge, ScienceDirect, Springer, Wiley Interscience, IngentaConnect, Taylor&Francis, Other
8	Type	+Journal +Conference +Other (dissertation, grey literature etc.)
9	Does it repeat already reviewed paper?	+Yes (Repeated ID) -No
Relevance		
10	Does it relate to a specific field of computer science?	+Yes –No xPartly
11	Does it relate to user assistance?	+Yes –No xPartly
Study Description		
12	Definition of embedded user assistance given in study	
13	Motivation	
14	Objectives/Research directions	
15	Background	Industrial or Academic
16	Subjects of investigation	Academics or Industry/Real world or both
17	Target area	
18	Technique for embedded user assistance	
19	Study design	Case study, Multiple-case study, Experiment, Survey, Benchmarking
20	Assessment approach	Qualitative / Quantitative or Both
21	Findings	
22	Benefits/limitations of embedded user assistance	
23	Implications for future research	
24	Major conclusions	
25	Concerns	
Evaluation		
26	Personal note	The opinions of the reviewer about the study
27	Additional note	Publication details (supported by grants etc.)
28	Quality Assessment scores	Detailed quality scores

Figure A.4: Data Extraction Form for Embedded User Assistance Systems

	Reporting				**Relevance**	**Rigor**				**Credibility**		
	Depth of research	Aims Objectives Rationale	Scope and Context	Coherence of reporting	Usefulness in practice and research	Description of Subjects of investigation	Clarity of evaluative appraisal	Explicitness of approach	Assumptions /theoretical perspectives/ values	Justification and credibility of results	Limitations	
Primary Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	**TOTAL** (out of 11)
[1]	1	1	1	0.5	1	0.5	1	0.5	1	0.5	0.5	8.5
[2]	1	1	1	0.5	1	0.5	1	1	0.5	0.5	0.5	8.5
[3]	1	1	1	0.5	1	0.5	0	0.5	0.5	0.5	0	6.5
[4]	1	1	1	0.5	1	1	0.5	0.5	0.5	1	0.5	8.5
[5]	1	1	0.5	0.5	1	1	0.5	0.5	0	0.5	0	6.5
[6]	1	1	0.5	0.5	1	1	0.5	0.5	0.5	0.5	1	8
[7]	1	1	1	0.5	1	1	0.5	0.5	0.5	1	0	8
[8]	1	1	1	1	1	1	1	1	0.5	0.5	0.5	9.5
[9]	1	1	0.5	0.5	1	1	1	0.5	1	1	0.5	9
[10]	1	1	1	0.5	1	0.5	0.5	1	0.5	0.5	0.5	8
[11]	1	1	1	0.5	1	0.5	1	0.5	1	0.5	0.5	8.5
[12]	1	1	0	0.5	1	1	0.5	0.5	0.5	0.5	0.5	7
[13]	1	1	0.5	0.5	1	0.5	0.5	1	0.5	0.5	0	7
[14]	1	1	1	1	1	0.5	0.5	1	0.5	1	0.5	9
[15]	1	1	1	0.5	1	0.5	0.5	1	0.5	1	0	8
[16]	1	1	1	0.5	1	0.5	0.5	1	0.5	0.5	0.5	8
[17]	1	1	0.5	0.5	1	1	0.5	0.5	0.5	0.5	0.5	7.5
[18]	1	1	0.5	0.5	1	0.5	0.5	0.5	0.5	0.5	0	6.5
[19]	1	1	1	0.5	1	0.5	1	0.5	1	0.5	0.5	8.5
[20]	1	1	1	0.5	1	0.5	1	0.5	0.5	1	0.5	8.5
[21]	1	1	1	1	1	1	1	1	0.5	1	0.5	10
[22]	1	1	1	1	1	1	1	0.5	1	0.5	1	10
[23]	1	1	1	0.5	1	1	1	0.5	0.5	0.5	0	8
[24]	1	1	1	0.5	1	0.5	0.5	0.5	0.5	1	0	7.5

Figure A.5: Study Quality Assessment of Embedded User Assistance

Appendix B

Case Applications

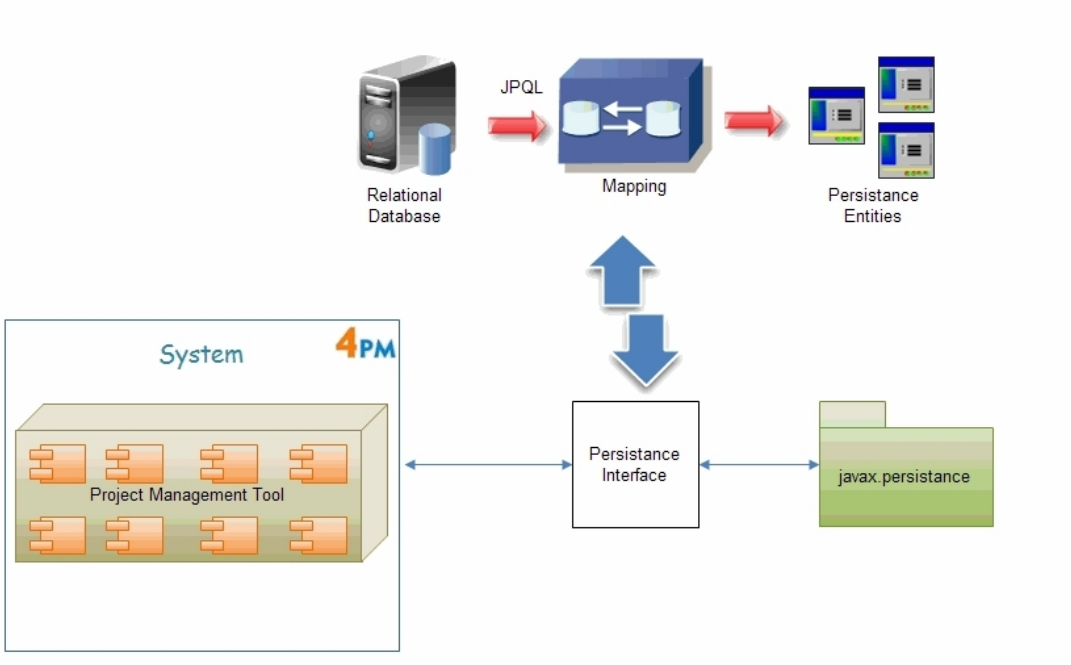

Figure B.1: The overall architecture of 4PM

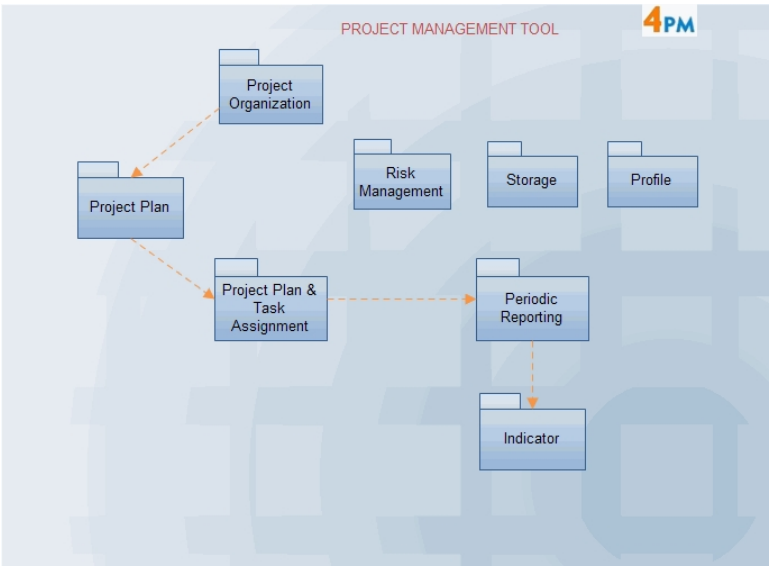

Figure B.2: The package diagram of 4PM

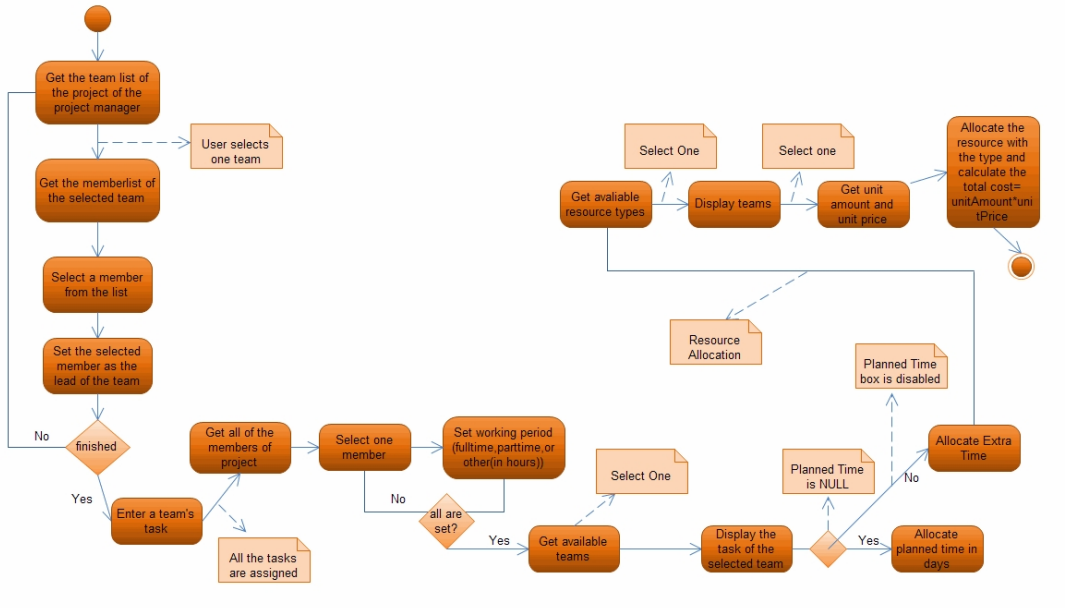

Figure B.3: The activity diagram of project planning in 4PM