重庆理工大学毕业设计（论文）

Graduation Project(Thesis) of Chongqing University of Technology

文　献　翻　译

Literature Translation

学院(College) 计算机科学与工程学院

班级(Class) 6201703LI

学生姓名(Student Name) Md Anower Hossain

学号(Student ID) 62017010084

Original 1.

Advocate Notepad: An Android Application

ABSTRACT

Smartphones are steadily gaining popularity with an estimated 1.6 billion mobile device users by 2013.With the advancement of technology, creation of new android based application plays an important role in making common man life easy and smart. A number of Android applications are developing in the market by a rapid rate. Managing common man task in various fields like entertainment, lifestyle, business, social, games etc. with these applications is today's demand.

Motivated with this, a free Android app named Advocate Notepad is proposed and developed in this paper for advocates to help them in maintaining the records of the cases on day to day basis with the details of the nature of the proceedings conducted on a particular day. Common man can use this App to know and learn about basic Laws that help common people to get rid of being exploited by one way or another. This app helps user in knowing basic Laws like air pollution, water pollution, waste management, chemical safety, forest resources under environment law. Time to time Reminders to advocates about their cases is an additional feature incorporated in the app.

Keywords

Android SDK, Android Application, Modules

1. INTRODUCTION

On September 23 2001, Google developed a mobile operating system called Android. An Android app is a software application running on the Android platform. Android Applications are written primarily in java programming language using the Android Software development kit (SDK) which includes a comprehensive set of development tools. By July 2013, the

Google Play store consists over one million Android applications published, and over 50 billion applications downloaded worldwide. With the passage of time, various versions of android like Frodo, Gingerbread, Ice Cream Sandwich, Jellybean, KitKat, Lollipop and the latest one is Lollipop MR 1 are introduced in the market with new one overcoming the bugs and shortcomings of previous version. There were about 67 million smartphones in India in 2013 out of which, nearly 90% are android based. Thus the knowledge of "legal education" via an Android application will help a lot to solve the general problems as it is a widely used technology at present. Law is a system of rules that helps to manage politics, economics and society in a number of ways. Our constitution has provided certain fundamental Rights for every citizen as well as for every person and unawareness of these fundamental rights result to their violation. Thus Basic and fundamental knowledge is a must for these common people so that they may not get exploited and their knowledge of the laws helps them to take benefit of their rights.

Thus basic knowledge of certain laws and regulations is must for a person even though it does not relate to legal field. Let's take an example: Some situation may arise like breaking traffic rule in hurry or by mistake that can result you to pay unnecessary Money to the traffic police as bribe; but the person's accurate knowledge or availability to these laws using this app can help him to pay or take exact decision as per the Law and not what is wrong. Usually a common man has to pay certain amount to the Lawyer to whom he may concern for small issues like Dowry, environmental issues, family issues etc. In such situation, the availability of legal information as first-aid will help the common man in protecting his fundamental rights.

Another aspect of Legal World relates with Advocates as they are an important part of it. Usually advocate hire attendant for maintaining their regular case details and past history, but maintaining a manual diary may not be as effective with today's advance technologies. Thus, incorporation of technology such as android app in this field can be of great use to advocate as well as common men. It will also be more secure and speedy.

Therefore, an android application is proposed in this paper to incorporate technology in the field of Law for advocates and common man. This android application will play a key role in providing a simple, systematic and speedier life to the advocates. This will help the advocates in raising their standards of working and in providing a better service to their clients while saving their time and resources. The proposed app will provide all information regarding day-to-day cases, client’s personal information, completed and ongoing cases, basic Laws and Rights at their fingertips anytime, anywhere. Searching, notifications, setting reminder are other features that will enhance the importance of this app for advocates. The objective of the app also focusses in resolving the problem of violation and exploitation of common man fundamental rights by providing them the basic knowledge of Laws and Rights such as penalties or punishments associated with exploitation of rules and regulations.

The rest of the paper is organized as follows. In Section II, we will present the study of related work done regarding the incorporation of technology in the field of Law. In Section III, we propose the design of the proposed app. In Section IV, we discuss the implementation details regarding the development of app. In Section V, we present the results of the developed app. In Section VI, we conclude the paper.

2. RELATED WORK

In this section, we present the existing websites and android applications that aim to fulfil the same criteria as ours but are lacking behind due to some reasons. Some of them are discussed below.

2.1 Manual Diary

The concept of maintaining a **Manual Diary** is one of the earliest methods for storage and retrieval at later date by the advocates and is now replaced by other methods due to the various shortcomings like absence of alert system, lack of privacy, difficulty in updating, wastage of time and paper, limited flexibility, high maintenance etc.

2.2 Web-based Applications

These applications use a website as an interface between user and application that allow user to access anytime, anywhere using browser and internet facility. Some of these are discussed below

2.2.1 Thevakeel.com

This site is used to maintain the details to be managed by Advocates about their different Cases, Detail of their Clients, Search various Cases via various ways etc. The major con of this site is its non-availability on Smart-phones, hosting cost, Large overhead to manage bulk of data, portability and high maintenance, internet connection. As Android App can provide these facilities with more without Laptop or PC usage, it will be better to port such applications on smartphones.

2.2.2 Indian Kanoon

It is a Search Engine for different Rights and Law by various keywords like acts related to Women, Property Act, Acts defining traffic rules, Acts against Exploitation of Children, Domestic Violence Act etc. But the understanding of various sections and their further subdivision may be difficult for common people. So a direct source of information for common Laws and rights seems better.

2.2.3 Advocate-Diary

This tool is designed to provide an electronic assistant to the advocates to manage their case and client's information, to maintain their schedule and fees details etc. The availability of all case details at a place enhance this website, but the major con is its trial period of 15 days only after which charges are imposed for further usage. Also, it does not provide features like notifications, reminder, information regarding laws and rights.

2.3 Android Applications

Since the usage of smart phones is more common than computers today and they offer many features in a portable and simpler way, Android applications are much in trend. Some of the android based applications for advocates are given below:

2.3.1 Advocate Diary Case Mgt. Free

This app incorporates certain features like schedule reminder on email either weekly or monthly, manage cases, offline application functionality, multiple website support etc. This android app involves Pro version charges- 499/- for One year and includes services like database collection. The cost charged for the app to get installed is a major drawback of it along with the non-availability of rights information for common man.

2.3.2 Advocates' Diary

This android application was Created in July 23,2012 for the advocates to alter their manual diary by providing fast and simple way for its maintenance. Being one of the earlier android app for the use of advocates, the maintain drawback of it is its installation charges of Rs.300 per year.

2.3.3 The Advocate

This app gives us access to up-to-date local news of various fields like Law, sports, entertainment etc. This app maintains the up-to-the-minute local news coverage, in-depth coverage of SU, LSU, and Saints etc. but the major drawback of this app is that it does not allow advocates to maintain their case and client details as well as lack behind in keeping updated Laws and Rights.

According to the market survey, the above mentioned are some of the web-based and android based tools developed for advocates, each having pons and cons that make their usage limited and inappropriate. To overcome the shortcomings of these existing work and providing advocates much more ease, we developed an Android Application which will be available at Google Play Store Free of cost and also provides high Quality Service to the Advocates as well as easy understandability to the Common man.

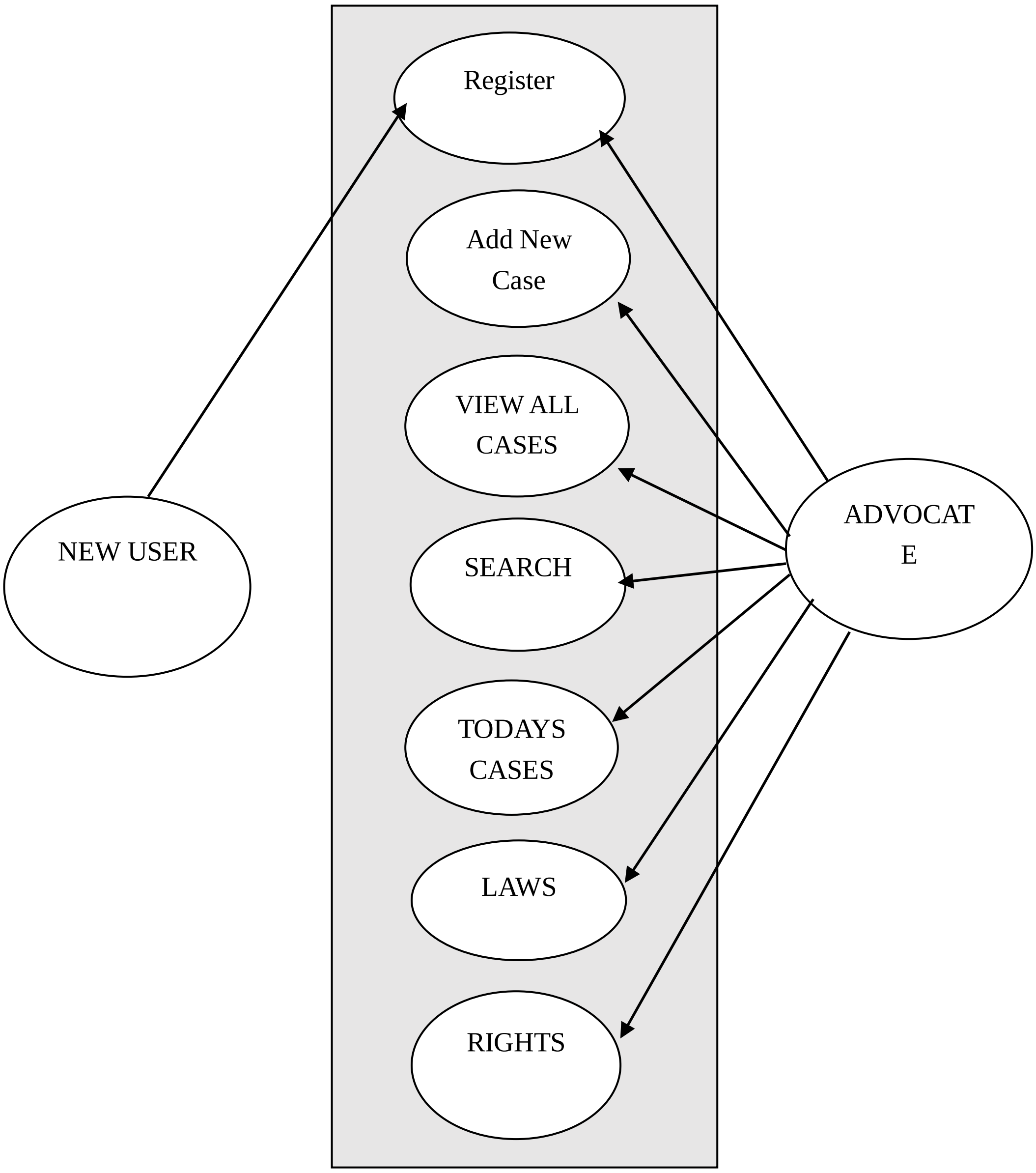


Fig 1: Use Case diagram

3. PROPOSED DESIGN

An Android app named ADVOCATE NOTEPAD is developed and presented in this paper for advocates to help them in maintaining the records of the cases on day to day basis with the details of the nature of the proceedings conducted on a particular day. The life of an Advocate revolves and guided by his/her Diary of cases and is virtually handicapped without it. Using this app, they can have all the information they need about all the cases of the clients, at their fingertips, which they can use anytime of the day or night and at any place. This app helps them to manage and co-ordinate their cases, which will enhance their productivity and earning. Time to time Reminders to advocates about their cases is an additional feature incorporated in the app.

The proposed app is represented with the help of a use case diagram as shown in Fig.1 that represents advocate and user that can use the app and access its features at their ease. The app will be available at google play store free of cost. Every person who installed the app needs to sign-in himself at first step. Once sign-in, the advocate will have access to the app with the entire database maintained personally on his phone. Different features are available to the Advocate like add a case, view all cases, search a case (by a keyword), Access to Laws and Rights, get notification for today's case etc.

The project aims to present an organized, secure, feasible and efficient way of maintaining diary of lawyer digitally. The app will ease the maintenance of case records and all related information with the motive of transforming the present day paper-work based storage to full-fledged digital storage system. The system will be providing seamless applications capable of handling large-scale records.

4. MODULAR DIVISION OF APPLICATION

The project is basically divided into following modules: Registration Module, Add A Case Module, All Cases Module, Search A Case Module, Today's Case Module, Laws Module, Rights Module, Admin Portal Module, Settings Module. This section describes the basic functioning idea of each module.

Registration- A new user can register by filling registration form by filling its email-id and phone number a two mandatory fields that creates an account on Advocate Notepad to access its features.

Add a Case Module- An advocate can add new case details in the app such as Name of Client, Mobile No., Email, Address, Case Name, Case Details, Case Category, Date.

All Cases Module- The all cases module enables advocate to look onto his various cases and its history. It provides all the cases added by each advocate.

Search a Case Module- An advocate can use this module to search and get entire details for the case which have been added by him in the past, so its time get saved.

Today's Case- This module enables its advocate to have a look onto the cases which are going to take place today and also gets a notification as a reminder to him.

Laws and right- This module provides information about the laws made in the constitution of India. The bad luck is people of India are not aware of Indian laws and rights available to them and are thus bribed, looted at several places. This app tries to ease this task by making available general basic law and rights category wise.

Admin Portal Module- This module gives rights to the administrator of the system to log in and can add new features to the app as well as update the laws and rights module.

The client side with the database and server with the help of sequence diagram. The vertical line indicates the objects and horizontal line indicates the message exchange to access the data and interaction between the client and server.

5. RESULTS

To evaluate the functioning of each mentioned module, the results of each working module is snapped. We believe the proposed application not only facilitates advocates in maintaining their day-to-day records but also aims at providing them a simple, speedier and systematic life. Even though it can also help the advocates and common people to gain the knowledge of various rights, Laws etc. much quickly and in a simple way. In this aspect, it is important to point out that adopting this application is more beneficial for advocates than any other means of maintaining their records safely. Hence, our goal of developing an android application for advocates and common people is a useful tool for them.

6. CONCLUSION AND FUTURE WORK

We would appreciate further discussions and implementations in the project in the future, as a study must never stop. We have done much to make the application work good and look better and we tried to give our best in fulfilling our duty to grow this application as much as we were able to, in the given period of time. We would appreciate anyone's interest in growing this particular work further if the person wishes to. Some modules in the project like the Push Notification module, server implementation, etc. are still under implementation. We would like to add user panel, some e-books, GPS settings etc. are some other functions that need consideration in the future. Other functions of the like chat between advocate and client, etc. also need to be considered in the fore-coming future. At last, the developers would appreciate any further studies and thoughts that can be added to the android developer, if the reader wishes to.

Author:

- Sonal Gupta Assistant Professor ABES Engineering College Ghaziabad (UP)
- Aayushi Gupta, Student at ABES Engineering College Ghaziabad (UP)
- Aayush Mittal, Student at ABES Engineering College Ghaziabad (UP)

Nationality: INDIA

Originate from: International Journal of Computer Applications (0975 – 8887) Volume 139 – No.10, April 2016

翻译1。

倡导记事本：Android应用程序

摘要

智能手机正在稳步普及，到2013年，估计将有16亿移动设备用户.随着技术的进步，基于android的新应用程序的创建在让普通人的生活变得轻松和智能方面发挥着重要作用。许多Android应用程序正在市场上快速发展。在娱乐、生活方式、商业、社交、游戏等各个领域管理普通人的任务，这些应用是当今的需求。

基于此，本文提出并开发了一款名为Advocate Notepad的免费Android应用程序，以帮助提倡者维护日常案件记录，并提供特定日期诉讼性质的详细信息。普通人可以使用此应用程序了解和了解帮助普通人摆脱被某种方式剥削的基本法律。该应用程序帮助用户了解环境法下的空气污染、水污染、废物管理、化学品安全、森林资源等基本法律。该应用程序还包含了一项附加功能，即向律师提供案件的实时提醒。

关键词

Android SDK、Android应用程序、模块

1.简介

2001年9月23日，谷歌开发了一款名为Android的移动操作系统。Android应用程序是在Android平台上运行的软件应用程序。Android应用程序主要是使用Android软件开发工具包（SDK）用java编程语言编写的，SDK包括一套全面的

开发工具。截至2013年7月，googleplay商店已发布超过100万个Android应用程序，全球下载超过500亿个应用程序。

随着时间的推移，市场上推出了各种版本的android，如Frodo、Gingerbread、Ice Cream Sandwich、Jellybean、KitKat、Lollipop，最新版本是Lollipop MR 1，新版本克服了以前版本的缺陷和不足。2013年，印度约有6700万部智能手机，其中近90%是基于android的。因此，通过Android应用程序了解“法律教育”知识将有助于解决普遍存在的问题，因为它是目前广泛使用的技术。法律是一个规则体系，有助于以多种方式管理政治、经济和社会。我国宪法为每一个公民和每一个人规定了某些基本权利，不了解这些基本权利就会导致这些权利受到侵犯。因此，这些老百姓必须掌握基本的知识，以免被剥削，他们的法律知识有助于他们利用自己的权利。

因此，法律法规的基本知识是一个人必须具备的，即使它不涉及法律领域。举个例子：有些情况可能会出现，比如匆忙违反交通规则或者错误地违反交通规则，导致你向交警支付不必要的钱作为贿赂；但该人的准确知识或可用性，这些法律使用此应用程序可以帮助他支付或作出准确的决定，根据法律，而不是什么是错的。一般来说，一个普通人必须支付一定的费用给他可能关心的小问题，如嫁妆，环境问题，家庭问题等律师在这种情况下，法律信息的可用性作为急救将有助于普通人保护他的基本权利。

法律世界的另一个方面与倡导者有关，因为他们是法律世界的重要组成部分。通常提倡雇佣服务员来维护他们的常规案例细节和过去的历史记录，但是维护

一本手动日记在当今的先进技术下可能没有那么有效。因此，将android应用程序等技术融入这一领域，对倡导者和普通人都有很大的帮助。它也将更加安全和迅速。

因此，本文提出了一个android应用程序，以将技术融入法律领域，为倡导者和普通人服务。这个android应用程序将在为倡导者提供简单、系统和快速的生活方面发挥关键作用。这将有助于倡导者提高他们的工作标准，同时为他们的客户提供更好的服务节省他们的时间和资源。该应用程序将随时随地提供有关日常案件、客户个人信息、已完成和正在进行的案件、基本法律和权利的所有信息。搜索，通知，设置提醒是其他功能，将提高这个应用程序的重要性，为倡导者。该应用程序的目标还侧重于解决侵犯和利用普通人基本权利的问题，向他们提供法律和权利的基本知识，例如与利用规章制度有关的惩罚或处罚。

论文的其余部分组织如下。在第二节中，我们将介绍在法律领域纳入技术方面所做的相关工作。在第三节中，我们提出了应用程序的设计方案。在第四节中，我们讨论了应用程序开发的实现细节。在第五节中，我们介绍了开发的应用程序的结果。在第六节中，我们总结了本文。

2.相关工作

在本节中，我们将介绍现有的网站和android应用程序，这些网站和android应用程序旨在达到与我们相同的标准，但由于某些原因而落后。下面将讨论其中的一些问题。

2.1 手册日记

这个概念维护手动日记是最早的存储和存储方法之一。提倡者后来检索 ,由于 各种缺陷，例如缺少警报系统，缺乏隐私，更新困难, 浪费时间和纸张，灵活性有限，维护成本高等

2.2 基于 Web 的应用程序

这些应用程序使用网站作为用户和应用程序之间的接口，允许用户使用浏览器和互联网设施随时随地访问。下面将讨论其中一些问题

2.2.1 Thevakeel.com

本网站用于维护倡导者管理的有关其不同案例的详细信息、其客户的详细信息、通过各种方式搜索各种案例等。本网站的主要缺点是其在智能手机上不可用、托管成本高、管理大量数据的开销大、可移植性和高维护性、互联网连接。由于 Android 应用程序可以在不使用笔记本电脑或 PC 的情况下为这些设施提供更多的功能，因此最好将这些应用程序移植到智能手机上。

2.2.2 印度卡农语

它是一个搜索引擎，通过各种关键字搜索不同的权利和法律，如与妇女有关的法案、财产法、定义交通规则的法案、禁止剥削儿童的法案、家庭暴力法案等。但是，普通人可能很难理解各个部分及其进一步细分。因此，普通法和权利的直接信息来源似乎更好。

2.2.3 律师日记

此工具旨在为辩护律师提供电子助理，以管理其案件和客户信息，维护其时间表和费用详情等。在一个地方提供所有案件详情可增强本网站的功能，但主要缺点是其试用期为 15 天，之后将收取费用供进一步使用。此外，它不提供通知、提醒、有关法律和权利的信息等功能。

2.3 安卓应用

由于如今智能手机的使用比电脑更为普遍，而且它们以便携和简单的方式提供了许多功能，因此 Android 应用程序成为一种趋势。以下是一些基于 android 的倡导者应用程序：

2.3. 1 建议日记病例管理。免费的

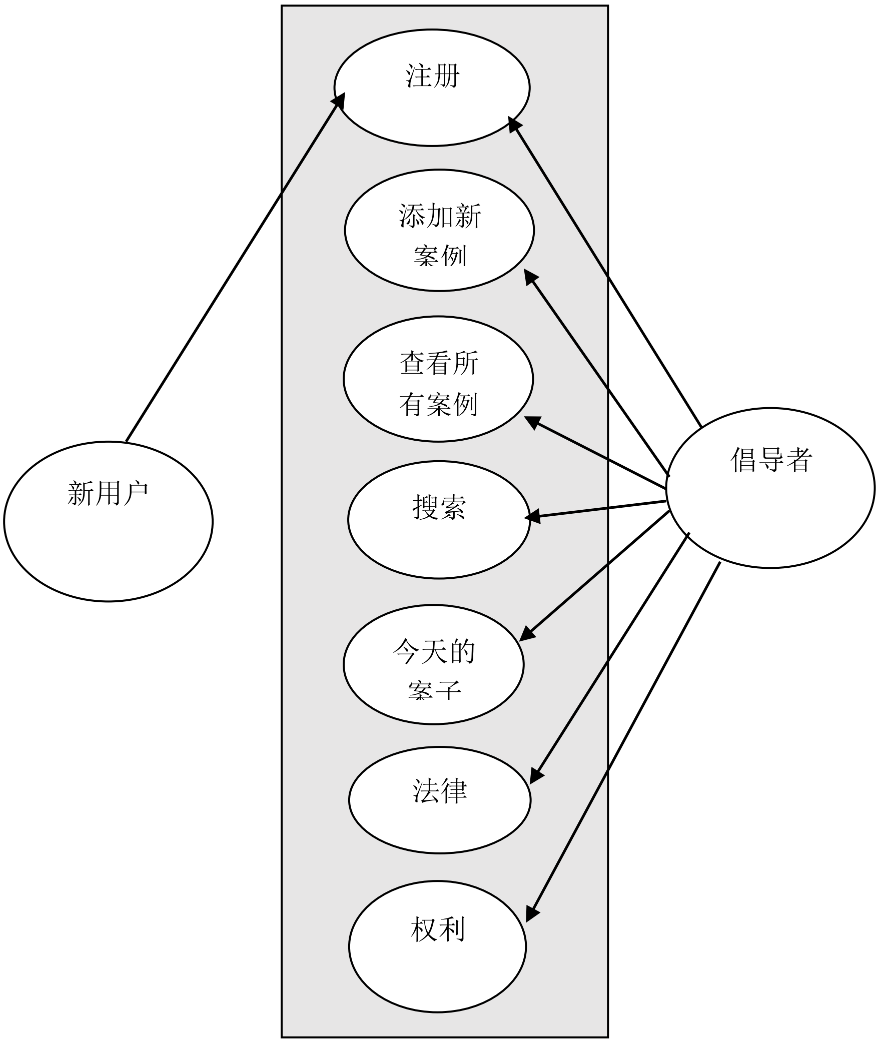
此应用程序包含某些功能，如每周或每月的电子邮件日程提醒、管理案例、离线应用程序功能、多个网站支持等。此 android 应用程序涉及专业版收费-499/-一年，并包括数据库收集等服务。安装该应用程序所需的费用是它的一个主要缺点，同时它还不能为普通人提供权限信息。

2.3.2 律师日记

这个 android 应用程序创建于 2012 年 7 月 23 日，为倡导者们提供了一种快速简单的维护方式来修改他们的手动日记。作为早期的 android 应用程序之一，它的缺点是安装费为每年 300 卢比。

2.3.3 倡导者

该应用程序使我们能够访问法律、体育、娱乐等各个领域的最新本地新闻。该应用程序保持最新的本地新闻报道，对苏、路易斯安那州立大学，但这个应用程序的主要缺点是，它不允许律师维护他们的案件和客户的详细资料，以及缺乏及时更新的法律和权利。



根据市场调查，上面提到的是一些为倡导者开发的基于 web 和 android 的工具，每个工具都有 pon 和 cons，使它们的使用受到限制也不合适。为了克服这些现有工作的缺点，并为倡导者提供更多的便利，我们开发了一个 Android 应用程序，将在谷歌 Play 商店免费提供，同时也为倡导者提供高质量的服务以及普通人易于理解的服务。

3.拟定设计

本文开发并展示了一款 Android 应用程序 advocate NOTEPAD，以帮助辩护律师维护案件的日常记录，并提供特定日期诉讼性质的详细信息。辩护律师的生活是以他/她的案件日记为指导的，没有日记几乎是残疾的。使用这个应用程序，他们可以在指尖上获得所需的关于所有客户案例的所有信息，他们可以在白天或晚上的任何时间和任何地点使用这些信息。这个应用程序帮助他们管理和协调他们的案件，这将提高他们的生产力和收入。该应用程序还包含了一项附加功能，即向律师提供案件的实时提醒。

建议的应用程序是通过一个用例图来表示的，如图 1 所示，该用例图表示可以轻松使用应用程序并访问其功能的倡导者和用户。该应用程序将在谷歌 play 商店免费提供。安装该应用程序的每个人第一步都需要自己登录。一旦登录，倡导者将有权访问该应用程序，并在其手机上亲自维护整个数据库。律师可以使用不同的功能，如添加案例、查看所有案例、搜索案例（通过关键字）、访问法律和权利、获取今天案例的通知等。本计画旨在提供一个有组织、安全、可行及有效的律师日记数位化维护方法。该应用程序将简化案件记录和所有相关信息的维护，目的是将目前基于纸面工作的存储系统转变为成熟的数字存储系统。该系统将提供能够处理大规模记录的无缝应用程序。

4.应用模块划分

该项目主要分为以下几个模块：注册模块、添加案例模块、所有案例模块、搜索

案例模块、今日案例模块、法律模块、权限模块、管理门户模块、设置模块。本节介绍每个模块的基本功能思想。

注册-新用户可以通过填写注册表格进行注册，注册表格包括电子邮件 id 和电话号码两个必填字段，这两个字段在 Advocate Notepad 上创建一个帐户以访问其功能。添加案例模块-律师可以在应用程序中添加新的案例详情，如客户姓名、手机号码、电子邮件、地址、案例名称、案例详情、案例类别、日期。所有案例模块-所有案例模块使辩护律师能够查看其各种案例及其历史。它提供了每个律师添加的所有案例。搜索案例模块-律师可以使用此模块搜索并获取他过去添加的案例的全部详细信息，从而节省时间。今天的案例-此模块使其倡导者能够查看今天将要发生的案例，并收到通知作为对其的提醒。法律与权利-本单元提供有关印度宪法中制定的法律的信息。不幸的是，印度人民不知道印度的法律和他们享有的权利，因此在一些地方受到贿赂和抢劫。这个应用程序试图通过提供一般的基本法和权利类别来减轻这项任务。管理门户模块-此模块授予系统管理员登录的权限，可以向应用程序添加新功能以及更新法律和权限模块。客户端通过顺序图与数据库和服务器连接。垂直线表示对象，水平线表示访问数据的消息交换以及客户端和服务器之间的交互。

5.结果

为了评估每个模块的功能，我们会将每个工作模块的结果截取下来。我们相信，建议中的应用程序不仅有助于倡导者维护他们的权利 每天的记录，也旨在为他们提供一个简单，快速和系统的生活。即使它也可以帮助倡导者和普通人获得

各种权利、法律等，以一种简单快捷的方式。在这方面，重要的是

指出采用此应用程序比任何其他方法对倡导者更有利安全保存他们的记录。因此，我们的目标是为拥护者和普通人是他们的有用工具。

6.结论和未来工作

我们将感谢在未来的项目中进一步的讨论和实施，因为研究永远不能停止。我们已经做了很多工作，使该应用程序的工作良好，看起来更好，我们试图尽我们最大的努力，履行我们的职责，尽我们所能，在给定的时间内增加这个应用程序。如果有人愿意，我们将非常感谢任何人对进一步发展这项工作的兴趣。项目中的一些模块，如推送通知模块、服务器实现等仍在实现中。我们想添加用户面板，一些电子书，GPS 设置等是一些其他功能，需要在未来的考虑。在未来，还需要考虑其他的功能，如倡导者和客户之间的聊天等。最后，如果读者愿意的话，开发人员将非常感谢任何可以添加到 android 开发人员中的进一步研究和思考。

作者:

- Sonal Gupta Assistant Professor ABES Engineering College Ghaziabad (UP)
- Aayushi Gupta, Student at ABES Engineering College Ghaziabad (UP)
- Aayush Mittal, Student at ABES Engineering College Ghaziabad (UP)

国籍: INDIA

从起源: International Journal of Computer Applications (0975 – 8887) Volume 139 – No.10, April 2016

Original 2.

The Designers' Notepad: Supporting and understanding cooperative design

ABSTRACT

We describe the development of a system to support cooperative software design. An iterative development approach has been used, based upon the observation of system use in authentic design sessions. This allows us to correct interface errors, and also to learn more about the nature of collaborative design. The observations of use and the resulting refinements of the system are described. In particular, we note the variability in design activity both amongst designers and according to circumstances. We also note the way in which concepts mutate over time (often involving frequent and rapid revision) leading to an evolution of structure.

1. Introduction

Supporting the work of designers has been a major focus for the developers of cooperative systems and is illustrative of one of the fundamental problems of developing CSCW systems. While we recognize that most design involves collaboration, our understanding of its nature as a cooperative process is limited. Worse, our intuitions as to the support required may themselves be flawed (Grudin88). Consequently, tool developers must discover the nature of the design process while simultaneously developing mechanisms to support and potentially improve it. We believe this interplay between the nature of an activity and the influence of the supporting tool to be a central feature of all CSCW systems development. Over the last two years we have been developing a system which provides support for system design. This paper describes the iterative approach to systems development based upon the observation of system use in realistic design sessions. The observations of use and the various features of the system which have emerged as a result of these observations are also described. The development of co-operative systems is a

problematic endeavor requiring a combination of skills often drawn from a range of different disciplines. By use of iterative development, we can make use of the results of an ongoing ethnographic study of collaborative design (Button
& King 92) as these results become available.

2. Supporting the work of designers

Design is an essential part of the systems development process and has consequently attracted considerable interest from the software development community. This has resulted in a range of different methods including JSD (Jackson 83) and OOD (Booch 91). These approaches prescribe a particular design model and require designers to adopt that perspective. The methods have been increasingly supported by the use of CASE tools (CASE 89). However, while CASE tools have provided techniques to support the enforcement.

The design process is often viewed by designers themselves as a creative and personal activity (Lawson 80). It involves the development and normalization of concepts relating to the artefact being constructed. Designers tend to adopt flexible and personal notations to express these concepts. These take the form of diagrams, sketches and personal notes. These design notes form an integral part of the activity of design. However, they are viewed as personal resources and consequently are not intended to be interpreted by others.

The reality of modern design is that the work is shared both between different designers and across different phases of the development process. This communication and sharing of designs requires the adoption of some standard notation or formalism. To date these notations have been provided by different design methods. The methodological approach adopted by design methods and CASE tools have proven problematic, so that designers tend to design 'away from the tool' and use the tool to document designs after the event**.** Thus little tool support is actually provided for the creative design process.

A number of tensions exist in the design process which need to be addressed to allow support for design as a creative activity. Most notably two tensions which we intend to address are:

The private v communal nature of design.

Although taking place within a cooperative setting many aspects of design are essentially a personal endeavor. The interplay between public and private design is a significant portion of design. approaches suggested by different methods, they have provided very little support for the creative process of design, as distinct from solution structuring, refinement and documentation. Freedom of expression v formalization of shared understanding Initial design concepts require a high degree of freedom in the notation used to express them. However, subsequent activities within the design and development process require a greater degree of formality to alleviate problems of misunderstanding across a community of designers and developers.

Our approach involves the development of techniques to directly address these tensions. We believe these tensions to be central not only to design but to many cooperative activities within real organizational settings. Consequently, the approach we have adopted to development is of interest to the CSCW community in general.

2.1 Previous approaches to design support

Within the domain of cooperative work, research in design support has generally followed one of two approaches. The first focuses on uncovering and recording the rationale used to arrive at different designs. The second concentrates on providing a shared surface for expressing designs and reflecting upon how these surfaces are used by designers. Our aim is to draw from the experiences of both these groups. The intent of design rationale is documentation of the sequence of decisions made in realizing a design. Some systems have adapted existing paper-based methods of externalizing rationales. The most notable of these are gIBIS (Conklin 88) and rIBIS (Rein 91). Other representations have been proposed (Carroll & Kellogg 89, Fischer & Girgensohn 90, Lee 90 and MacLean et al. 89).

A number of problems have been noted with design rationale systems: they may have problems of acceptability (Yakemovik & Conklin 91) and difficulties with forcing users to decide at an early stage on the nature of each piece of information. This can be difficult to do, particularly where collaborators have to agree on the classification (Shipman & Marshall 92, Conklin & Bergman 88). Much of the motivation of current research is in the development of an appropriate theoretical base for the representation of design rationales and on the cognitive process of design. Little consideration has been given to the development of techniques and tools to support the creation of design rational within the early creative portion of the design process.

The research on shared space (for example Bly 90, Greenberg 91 and Ishii 91) adopts a contrasting approach to information capture. These systems provide a space upon which designers can collectively express and structure their ideas in a relatively unconstrained way. They have often focused on drawing as the primary means of expression and have studied how users interact with and through the shared surface. The Cognoter tool (Stefik et al. 87) is an example of this group. It provided facilities to allow ideas to be expressed, collected and commented upon by other users. However, later studies of its use revealed communication breakdowns and suggested that these were a consequence of various assumptions in its design which did not relate to actual tool use (Tatar 91).

The NoteCards approach uses hypertext as a kind of shared surface. Although offering many advantages, classification problems similar to that for gIBIS have been observed (Shipman & Marshall 92, Monty 90). Users had difficulty chunking information into cards, naming cards and filing cards. Typed links were rarely used (and then, inconsistently). Link direction and link semantics also proved to be problematic. Many of these difficulties again spring from design assumptions which did not match actual systems usage.

3. Developing the DNP: an iterative approach

Our work builds upon the experiences of these previous approaches to design support and aims to tackle the difficulties that they unearthed. While we recognize the need to represent and record design information in a form which may ultimately be useful during later parts of the development process, we also acknowledge the problems of requiring designers to commit to a particular interpretation too early in the design process. Indeed, an overriding criterion for any system must be acceptability (Grudin 88). Therefore, our main focus is on the development of an interface that users find easy to operate and which is appropriate for the early stages of design. We do not have a theory of design that we wish to impose or test on designers, but rather a desire to discover their requirements. This has led us to an approach to tool development based upon rapid prototyping.

We support the notion that idea representation tools such as Cognoter or more general design surfaces provide a medium of representation within the conversation surrounding the design process (Tatar et al. 91). Previous studies have examined the use of traditional whiteboards and shared drawing media in the design process (Suchman 88, Tang 89). However electronic drawing surfaces are different from these traditional media (Tatar et al. 91) and gaining a clear understanding of how these are exploited to express design concepts is a crucial part of our approach. Our central problem is that the medium of expression plays a central role in the representations. In fact, we would argue that the possibilities offered by electronic systems to represent designs and the nature of these representations are sufficiently intertwined that neither can be adequately addressed in isolation. Thus, not only is little known about the software design process, whether by one or many people, but the development of computer systems to support the process makes some features necessarily unknowable. This is because the computerized design tool is likely to change the nature of the design process as the word processor has changed the nature of writing (Haas 89). Therefore, we have adopted an iterative approach to development based upon developing facilities in close cooperation with the designers using the system.

We started by providing an initial set of core facilities which allowed designs to be expressed. These facilities were then used by designers over a prolonged period to support a variety of real design tasks. This usage was observed and videotaped and the participants invited to comment on their experience with the system. The aim of the sessions was: -.

1. To isolate problems caused by features of the user interface.
2. To examine the use made of the system and to highlight additional functionality required by designers.
3. To assess the usefulness of existing functionality before adding to the system.

4. To provide information about the process of design itself. The prototyping approach is shown diagrammatically in Figure 1.

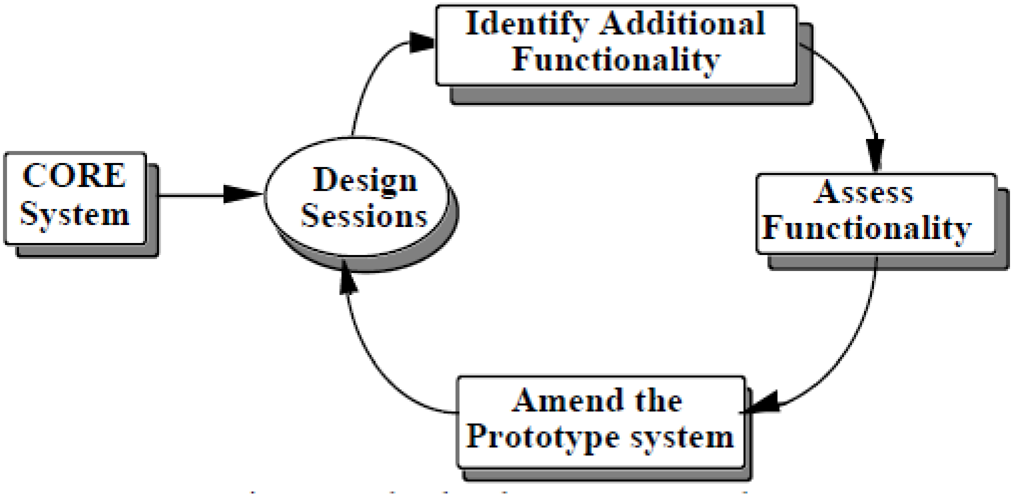


Figure 1. The development approach.

Our approach has links with participatory design (Bodker 91) and is similar to the approach proposed by (Tang 91). It also borrows from other domains including HCI (Hartson 91) and AI and Education (Twidale, in press). The latter domain has many similarities with CSCW in that both focus on the use made by people of computer systems and concern the support of human activities (learning and work respectively) that are imperfectly understood. In AI&ED it has been observed that a single flaw in an interface can have a substantial negative effect in learning outcomes, but also that a feature of the interface intended for one purpose can have additional beneficial purposes (Twidale 91). Thus features of the interface have the potential of swamping the effects of deeper, more interesting and sophisticated modules of a system.

An iterative approach to development can help to alleviate this problem. By testing basic versions of the system in simple circumstances, (such as with a single designer, two designers sitting at the same terminal or two designers with a terminal each sitting next to each other), we can eliminate the grosser interface errors. This can inform the design when it is extended to the necessarily more complex cases (such as many designers separated in space and with limited bandwidth communication). For this approach to work, we need to focus more on the interface difficulties rather than the successes. We can be confident that the difficulties will scale up to the more complex cases if nothing were done about them, while the successes may not.

3.1 The Software Design Process

Rather than considering all aspects of design as an abstract process, we are primarily concerned with the software development process s. This allows us to make particular assumptions about the design setting within which any development systems will be applied. In particular, the software design process is characterized by:

The development of structure

Software design is closely concerned with the transition from partially defined, loosely structured ideas and concepts to fully defined and structured design descriptions. These descriptions form the basic plans for subsequent development. The majority of these design descriptions are specified diagrammatically using some form of network diagram. These diagrams show a number of software entities which are linked to describe corresponding relationships.

The work of groups

Any design of a significant size is the endeavor of a number of designers. A team of software designers will work together in a variety of different ways over a substantial time period to realize a completed design. The patterns of work involved and the different forms of cooperation which take place are only partially understood by the participants in the design process.

This view of the design process forms the context for our construction of the DNP. It is important to note that software design forms only the initial phase of the overall development process and some benefits of supporting design activities may only be realized much later in the software lifecycle. To reap these benefits though, the design tool must be used by software designers. Meeting the needs of designers by meshing with their work practice becomes an issue of paramount importance.

3.2 Design Sessions

When investigating a domain so poorly understood as collaborative software design it is vital to be in contact with designers as early as possible. A tool that designers find unusable or even just awkward to use will not be adopted.

Given that our development of a system to support software design is necessarily exploratory, it is inevitable that we will make mistakes. The aim of our approach is to increase the speed and ease of identifying and correcting these mistakes. Considerable attention is given to mistakes relating to the interface. It is clear that the interface to any system has a substantial effect on its use. If care is not taken over the interface, problems of use may not be due to some feature of the support for CSCW that is counterproductive or missing but merely to an interface problem. It is possible to identify many of these issues when the system is being used by a single user, even though it is intended for collaborative work: difficulties that a single user has are more than likely to be also experienced with multiple users.

It is for this reason of simplicity that we chose to start by concentrating on a system for multiple simultaneous use of a single workstation by two or three people. A distributed version for two workstations has been developed and tested, but the bulk of the work has been on the single version. In the case of the distributed version we had the workstations side by side. We chose this in order to maximize the potential communication bandwidth between the participants. In this way any problems observed could not be ascribed to bandwidth limitations and so would scale up to true distributed usage if not remedied. The approach is intended to be complementary to ongoing research addressing the challenging technological and social implications of the more complex forms of collaboration such as synchronous working over long distances (Clark & Scrivener 92).

3.3 The Mini-DNP

Our starting point for providing support for designers was the development of a minimal system with a set of simple core facilities called the mini-DNP. This system was influenced by a previous system to support design by a single user (Sommerville et al. 1990) and the experiences gained from developing and using that system. The essence of mini-DNP is a means of creating entities and links between them (Figure 2).

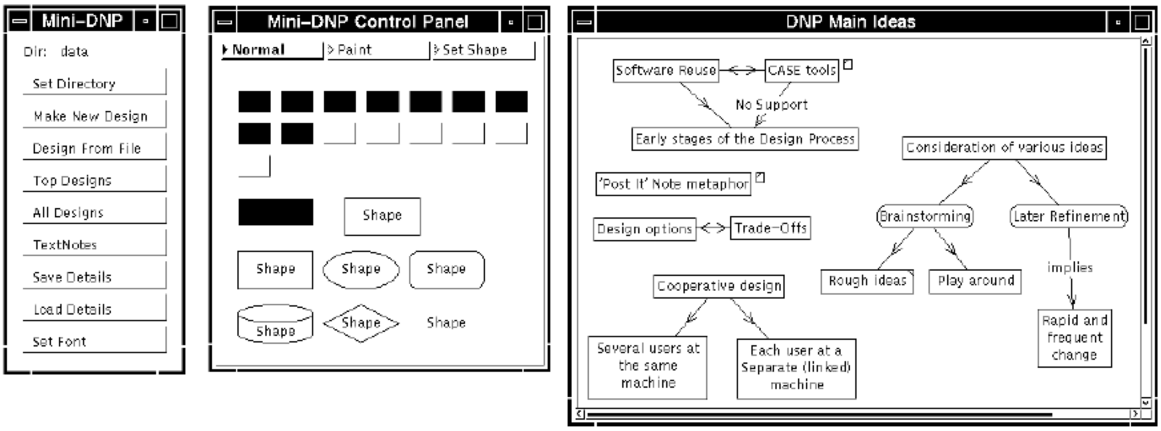


Figure 2. The minimal DNP interface.

The user creates an entity by typing in a design window. He or she may then move it with the mouse. Linking is done by selecting one entity and then clicking on the entity one wants to link to with the shift key down. This linking of entities forms the diagrams ultimately needed to specify design descriptions. We also provided text notes which are based on the Post-It Note metaphor (see figure 5) and allow designers to attach one or more notes to an entity. These can be used by design teams for more textual comments, ideas, opinions, code fragments, references etc. A variety of text note types are provided and users may define them own (including form structures). An entity with text notes has an icon attached (e.g. the entity 'CASE tools' in figure 2) and the notes can be examined by clicking on the icon. Designs can be saved and loaded from a file and a paper report may be created containing a screen dump of the design and a list of the entities and them text notes. Each entity may itself be expanded to become a sub design. A new window is opened and entities and links created in the normal way. Sub designs may contain entities that are themselves sub designs. A loose form of typing for entities and links is provided using color, shape and labels. The user controls the degree to which s/he wants to use typing. The type of an entity or link can easily be changed at any time.

The initial system has certain characteristics which make it suitable for supporting the design process:

1. A fast, easy-to-use interface which supports the creation of directed graphs. These may be created at any number of levels with simple navigation from one level to another.
2. Untipped entities and relations. This is critically important in the early stages of an analysis where it is unrealistic to fit an entity into a type hierarchy.
3. Extensive annotation facilities which allow system entities and relations to be annotated with structured and unstructured text.
4. Post-creation type attribution. These facilities also allow the type of an entity to be easily changed.
5. Report generation facilities.

3.4 Assessment

We have used a variety of users to assess the system. Given that the aim is continual refinement, one can initially use members of the development team to test the system. At first sight it may seem a problem that they will be more motivated than most to use the system and consider it in a favorable light. Nevertheless, provided that they have genuine tasks (such as the ongoing design of the system itself) they are likely to discover some of the grosser interface errors. We can be confident that such errors will scale out: a feature that a developer finds hard to use will certainly be difficult for an outsider (and of course if the developers don't like using the tool on a regular basis it is unlikely anyone will). Later we tried the system on users not involved in its development in order to discover any hidden assumptions in the developer community about ease of use and needed features.

The system has been simultaneously subject to use and under development for approximately two years. During this time a substantial community of user’s have exploited the system. Table 1 summarizes the variety of system use.

By designer:	Single designer
	Two designers at one workstation
	Three designers at one workstation
	Two designers at two adjacent workstations
by task:	Software design
	Information organization and browsing
	Project management
	Lecture course design
	Designing papers (including this one)
	Designing talks
By user:	System developer
	Project members
	PhD students
	Sociology colleagues
	Visiting academics
	Undergraduates

Table 1. Nature of usage of the system in tests.

In all cases of use, whether by individuals or groups, we asked our volunteers to bring along a task that they had to undertake anyway. We believe it is important to develop the system on real-world tasks as these have features that are very hard to replicate in contrived tasks. These features include ambiguity, open-endedness, history and engagement of the user, who ideally will be focusing more on the problem than the tool. Our instructions to the user were to try and use the tool, asking where necessary how to achieve anything they want to do. Our assumption is that the tool will be usable up to some point when it becomes frustrating because it prevents the user doing something s/he wants. We can identify what s/he wants to do and then can assess what sort of functionality should be added to the system.

Given the desire to use authentic tasks we are somewhat at the mercy of the needs of our volunteers. Fortunately, a tool intended to support the very early stages of software development is equally useful for many other creative design tasks such as project management, making ethnographic notes and the design of papers and reports (Sommerville et al. 93). Indeed, the initial structure and elements of this paper were collaboratively developed using DNP.

Some of the system use has been in one-off design sessions, others have continued over a number of sessions and yet others (particularly those involving project management and the design and management of research activities) are ongoing over many months. The system is now accessible from a number of different locations on campus. In particular, two users have access to the DNP from their office and use it on a day to day basis. These latter forms of use are beginning to reveal the nature of the requirements for management of the complexity of designs over time.

4. Design Support in the DNP

The intent of the design sessions was to discover the various patterns of use of the DNP (including difficulties) for different designers. Our observations of use informed the development of appropriate support facilities. This section briefly highlights the results of this process.

4.1 Variability of use

Our studies have confirmed the great variability in design activities both between users and by the same users over time and circumstance (see figures 3, 4, 5 & 6). For example, some designers use very terse entity names with text notes to contain details, whereas others use phrase- or sentence-like names. Some designers use many links to indicate connectedness whilst others use two-dimensional proximity (Marshall & Rogers 92). Some use a great deal of color and many different shapes, whilst others use black rectangles all the time. As examples of variability over time, we have observed cycles involving bouts of entity creation and rough positioning. These involve very rapid and intense activity where the minimum of options are employed. After such a bout there is a recovery period where the display is 'tidied up'. Links are created and the entities rearranged to convey additional meanings by their proximity to other entities as well as to reduce the clutter of areas of great activity.

This variability confirms the need for flexibility and ease of revision in the DNP. Much of this flexibility comes from our decision to avoid associating semantics with entities, links and sub designs and to allow these to be used primarily as a means of expression by users. The meaning of these initial design graphs is left to the interpretation of different designers. The intense 'bursts' of activity surrounding the generation of entities caused us to focus on supporting the rapid entry of entities (just typing and hitting return between entities generates entries positioned in a list format). Once entered these entities can be tidied by altering their position, color and shape. For example, contrast the design in figure 3 showing the early stages of the design with a later stage shown in figure 5.

4.2 The Evolution of structure

The elements of a design that are added to the Notepad in the form of entities are necessarily ambiguous; they consist of a label of a few words referring to a concept. However, not only is that label capable of misinterpretation by anyone

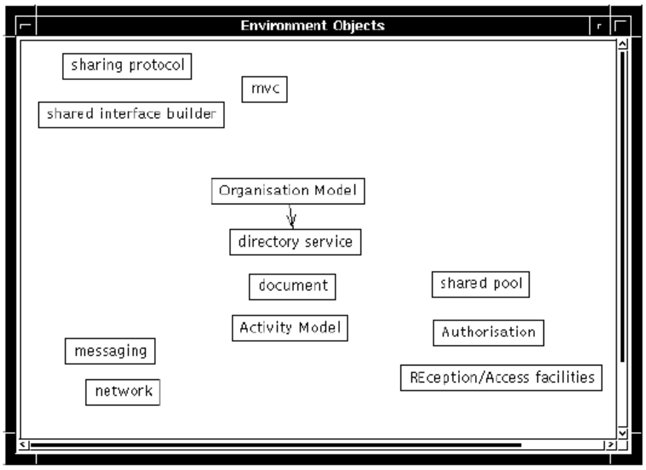


Figure 3. An initial design in the DNP.

other than its creator, but even for the creator it seems that deliberately ambiguous terms are chosen (for example, entities were often called "object" or "user"). This is an instance of postponing decisions about details in order to deal with overview concepts. Later on in the design process, the meaning of the initial concepts will be refined. This can involve qualifying the entity by editing its name (including completely changing the name), creating more entities for the constituent concepts in an ambiguous description, adding more information by attaching text notes, creating a sub design for the entity, or completely replacing the entity.

It would seem that designers need to have a certain degree of ambiguity during design so as to not get too held down by details. Gradually these ambiguities are addressed and refined. This has been observed in other situations (Marshall & Rogers 92) and reasons proposed for why users wish to avoid formalization (Shipman & Marshall 92). In a similar manner, the meaning of a link can change over time. Initially its meaning may be 'these entities are in some way connected'. Eventually this is refined into a more precise meaning. The gradual evolution of precision is often associated with the usage of link typing; links with a similar meaning are now given the same color or label.

The attributes of entities and links are used within the DNP as a mechanism to support the evolution of structure. The use of color and shape as a means of typing both entities and links is directly supported by the DNP. Once created these different attributes of entities and links can be modified by users over time from the control panel shown in figure 2. This freedom allows appropriate structure to emerge within the design after the entities have been created and allows the type of entities to change after the creation of the entity. Figures 3 and 5 illustrate this migration. This is in contrast to previous systems which supported the definition of a set of types but required the designer to select the type for each entity as it is created.

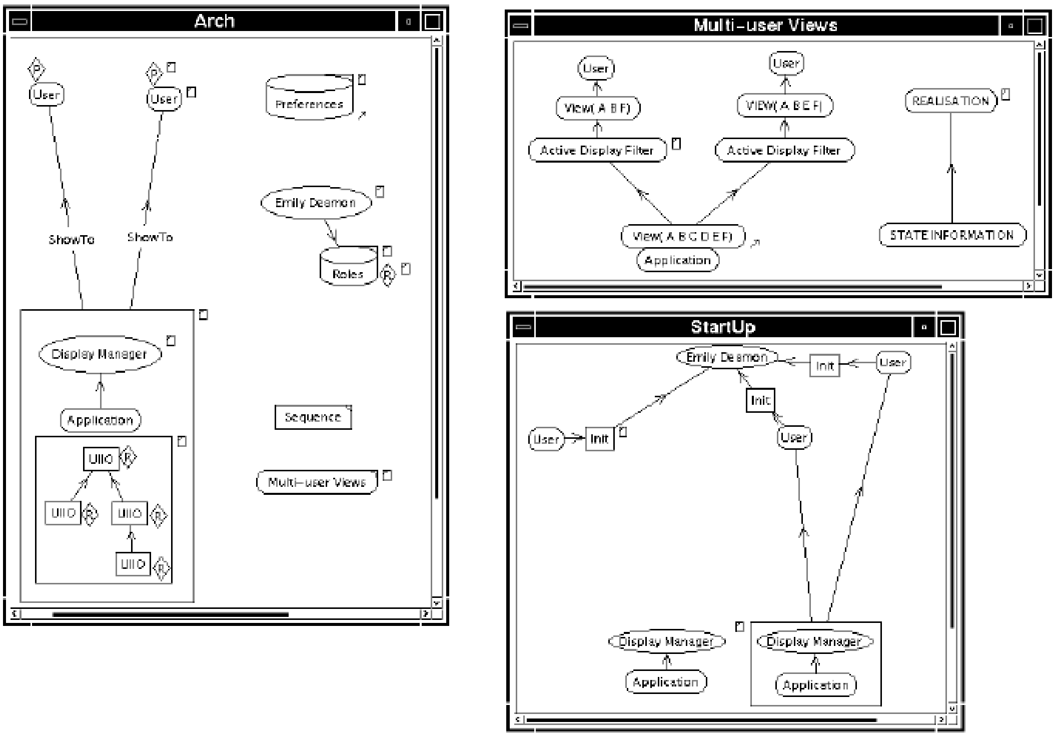


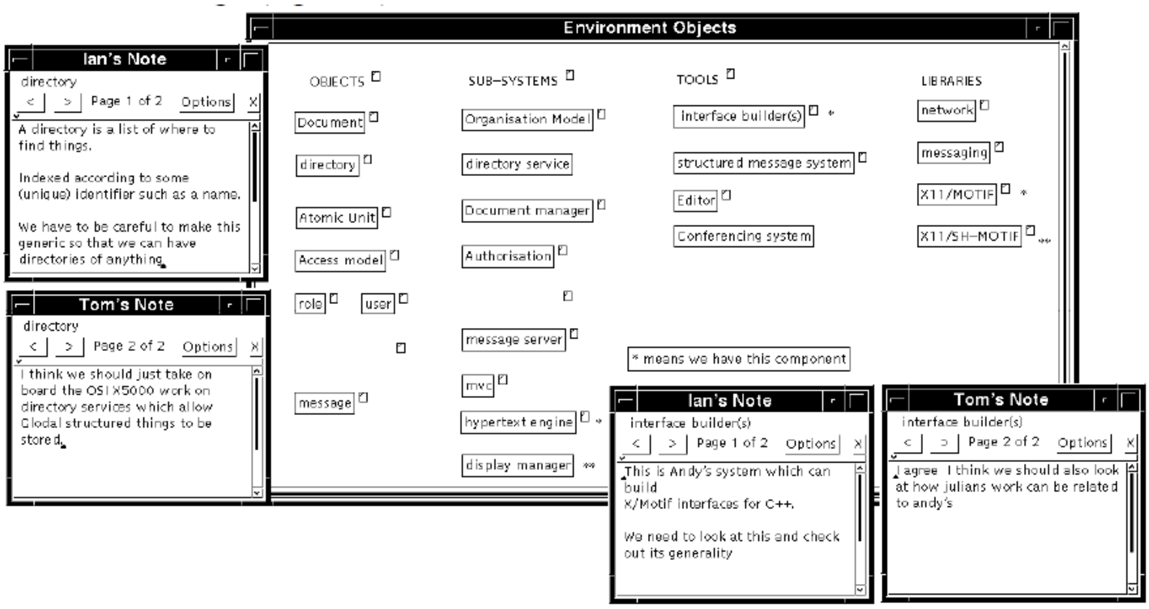
Figure 4. The arrangement, framing and grouping of entities.
The two designs on the right are sub designs of the one on the left. 11 colors were used in this design

The relationship of entities within any non-trivial design is extremely complex. In addition to links, space is often used to distinguish this relationship as appropriate entities are clustered (Marshall & Rogers 92). Users expressed a need to abstract from these groupings once they were made. Sub designs were used to facilitate this. The user can frame entities to form a group. S/he may then 'push down' the group into a sub design to be replaced by a single entity (figure 4).

4.3 Cooperative and Single use

The most visible difference between cooperative and single use is speed; single designers enter, refine and revise items far more quickly than do groups. This is mainly due to the need of group participants to justify and explain their actions to others. No action is completely unambiguous and the degree to which features can be left implicit by a single user are naturally far less than when ideas have to be shared. Not only is greater explanation and elaboration required, but actions such as revisions and additions have to be negotiated, leading to a debate about the appropriateness of the activity. By contrast, single designers create entities and links and rearrange them with at times almost bewildering speed. We take this as evidence for the need for the tool to support frequent and rapid revision of designs provided by the DNP.

The sharing of a keyboard and mouse on the whole did not appear to cause great difficulties, with users taking turns to enter text or rearrange entities. Control of the keyboard can involve either a position of power, temporarily controlling the collaboration, or it can involve a secretarial function, minting the deliberations of the others. In the distributed version of the DNP, despite the close proximity of the machines involved, users exploited the shared view of the design to quickly partition the design activity and work independently upon different portions of the overall design. Substantial use was made of text notes to annotate features of the other user's design (figure 5).



Ian's Note
directory
Page 1 of 2
A directory is a list of where to find things.

Indexed according to some (unique) identifier such as a name.

We have to be careful to make this generic so that we can have directories of anything

Tom's Note
directory
Page 2 of 2
I think we should just take on board the OSI X5000 work on directory services which allow Glodal structured things to be stored.

Ian's Note
interface builder(s)
Page 1 of 2
This is Andy's system which can build
X/Motif interfaces for C++.

We need to look at this and check out its generality

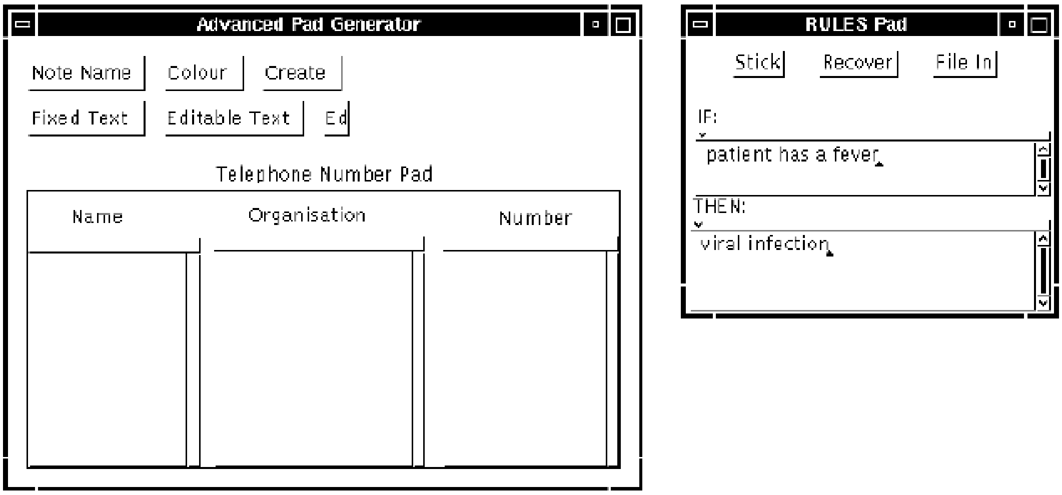
Tom's Note
interface builder(s)
Page 2 of 2
I agree I think we should also look at how julians work can be related to andy's

Figure 5. The use of text notes to comment on a design.

7 colors were used in this design.

An interesting feature of cooperative use in both settings was how entities were moved and highlighted for emphasis, as part of a debate. The user currently controlling the mouse would select an entity s/he wished to discuss, causing it to be highlighted. S/he might move this entity slightly (from side to side or in small circles) as a means of emphasis. This activity of moving entities for emphasis can be contrasted with occasions when the move was more substantial, indicating some semantic feature, by positioning an entity closer to another entity. This could be permanent, or merely a continuous movement as a suggestion. Besides the use of the mouse, more direct interaction was observed, namely users pointing at the screen with their fingers. The cooperative situation also saw another use of the DNP's ability to support the evolution of meaning. Often participants in a group design activity would use an entity as an argument placeholder and use this accepted design entity to refer back to a previous debate.

As a result of our observation of groups, the facilities associated with the placement of text notes were extended to allow users to easily exploit text notes as a method of annotation. These extensions allow facilities for the definition of new note types which can be associated with different users and/or purposes. Figure 6 shows the tool to create a new pad of notes with a user-defined structure, and another created notepad. Facilities are provided to filter and highlight the presence of notes from different designers.



IF: patient has a fever

THEN: viral infection

Figure 6. The generator for creating a structured note and a user defined notepad.

5. Conclusions

An iterative development approach is appropriate for CSCW systems development because it facilitates our need to learn more about the nature of cooperative activity while simultaneously developing the system. The testing of the early versions of the Designers' Notepad have emphasized the variability in design activity both amongst designers and according to circumstances. We also noted the way in which concepts mutate over time leading to an evolution of structure. This can lead to frequent and rapid revision. Ease of revision is important in encouraging the brainstorming of concepts that are necessarily incomplete. Systems to support the early stages of design need to support these features of the process. By means of an iterative development we are able to correct the grosser interface errors that have the potential to swamp the effect of deeper issues concerning design and cooperative work in general. The methodology also enables us to profit by the stream of results that an ethnographic study produces. In its current state the Designers' Notepad is as much a tool for acquiring information about design activity as it is a tool for supporting that activity.

6. Acknowledgements

Michael Twidale is a Science and Engineering Research Council Junior Research Fellow. The development of the Designers' Notepad was funded partially by the Joint Council Initiative in Cognitive Science and COMIC (Esprit basic research project 6225). The authors would like to thank their colleagues in the Sociology department; John Hughes and Val King for their support and advice throughout this project.

Author: Michael Twidale, Tom Rodden, Ian Sommerville Department of Computing, Lancaster University, U.K.
Nationality: U.K.
Originate from: Researchgate.net DOI: 10.1007/978-94-011-2094-4_7

翻译 2。

设计师记事本：支持和理解合作设计

摘要

我们描述一个系统的发展，以支持合作软件设计。在真实设计会话中对系统使用的观察的基础上，使用了迭一开发方法。这使我们能够纠正界面错误，并了解更多关于协作设计的性质。描述了使用观察结果和系统由此产生的改进。我们特别注意到设计师之间和根据情况在设计活动中的差异。我们还注意到概念随时间变化的方式（通常涉及频繁和快速的修订），导致结构的演变。

1. 介绍

支持设计师的工作一直是合作系统开发者的主要关注点，也说明了发展《特定常规武器公约》系统的根本问题之一。虽然我们认识到大多数设计都涉及协作，但我们对其作为合作过程的性质的理解是有限的。更糟的是，我们对所需支持的直觉本身可能有缺陷（Grudin88）。因此，工具开发人员必须发现设计过程的性质，同时开发支持并潜在改进其机制。我们认为，活动的性质与辅助工具的影响之间的这种相互作用是所有 CSCW 系统开发的核心特征。在过去两年中，我们一直在开发一个系统，为系统设计提供支持。本文在现实设计会议中观察系统使用的基础上，描述了系统开发的迭常方法。还描述了这些观测结果所产生的使用观察和系统的各种特征。合作系统的发展是一项有问题的工作，需要将通常来自一系列不同学科的技能结合起来。通过使用迭同开发，我们可以利用正在进行的协作

设计人种学研究的结果（Button & king 92） 因为这些结果变得可用。

2. 支持设计师的工作

设计是系统开发过程的重要组成部分，因此引起了软件开发界的极大兴趣。这导致了一系列不同的方法，包括 JSD（Jackson 83）和 OOD（Booch 91）。这些方法规定了特定的设计模型，并要求设计师采用这一观点。使用案例工具（CASE 89）越来越支持这些方法。但是，虽然 CASE 工具提供了支持执行的技术。

设计过程经常被设计师自己视为一种创造性和个人活动（Lawson 80）。它涉及与正在建造的文物有关的概念的发展和正常化。设计师倾向于采用灵活和个人的语法来表达这些概念。这些以图表、草图和个人笔记的形式出现。这些设计说明是设计活动的一个组成部分。然而，它们被视为个人资源，因此无意被他人解释。

现代设计的现实是，作品由不同的设计师和不同的开发过程的不同阶段共享。这种设计沟通和共享需要采用某种标准符号或形式主义。迄今为止，这些标注是由不同的设计方法提供的。设计方法和 CASE 工具采用的方法论方法已被证明是有问题的，因此设计人员倾向于设计"远离工具"，并在活动结束后使用该工具记录设计. 因此，小工具实际上为创意设计过程提供了支持。

设计过程中存在一些紧张问题，需要加以解决，以便将设计作为一种创造性活动加以支持。最明显的是，我们打算解决的两个紧张局势是：

设计的私人 v 公共性质。

虽然在合作环境中进行设计的许多方面本质上是个人的努力。公共设计与

私人设计之间的相互作用是设计的重要组成部分。通过不同方法提出的方法，它们很少支持设计的创造性过程，这与解决方案结构、改进和文档不同。言论自由与共同理解的正规化初始设计概念需要在用于表达这些概念的符号中具有高度的自由度。然而，在设计和开发过程中的后续活动需要更大的形式来缓解设计师和开发人员社区的误解问题。

我们的方法包括开发直接解决这些问题的技术紧张局势。我们认为，这些紧张局势不仅对设计，而且对许多人都至关重要在真实组织环境中的合作活动。因此我们所采取的发展方法是 CSCW 社区感兴趣的一般来说。

2.1 以前设计支持的方法

在合作工作领域，设计支持研究通常遵循两种方法之一。第一个重点是发现和记录用于得出不同设计的理由。第二个重点是提供一个共享的表面来表达设计，并反思设计师如何使用这些表面。我们的目标是从这两个群体的经验中吸取教训。设计理由的意图是记录在实现设计时做出的决策顺序。一些系统已经调整了现有的纸质理论外化方法。其中最引人注目的是吉比斯 （Conklin 88） 和里比斯 （Rein 91）。提出了其他陈述(Carroll & Kellogg 89, Fischer & Girgensohn 90, Lee 90 and MacLean et al.89)。

设计原理系统注意到了一些问题：它们可能有可接受性问题（Yakemovik & Conklin 91），以及迫使用户在早期阶段决定每一条信息的性质的困难。这很难做到，尤其是在合作者必须就分类达成一致的情况下（Shipman & Marshall 92, Conklin & Bergman 88)。当前研究的主要动机是为设计理念的表达和设计认知过

程的认知过程建立适当的理论基础。很少考虑开发技术和工具，以支持在设计过程的早期创造性部分合理设计创造。

共享空间的研究（for example Bly 90, Greenberg 91 and Ishii 91）在信息捕获方面采用了对比鲜明的方法。这些系统提供了一个空间，设计师可以以相对不受约束的方式集体表达和构建他们的想法。他们经常专注于绘画作为主要的表达方式，并研究了用户如何与共享表面互动并通过共享表面。科格诺特工具（Stefik et al. 87）就是这一群体的一个例子。它提供了设施，使其他用户能够表达、收集和评论想法。然而，后来对其使用的研究表明，通信中断，并表明这些是其设计中与实际工具使用无关的各种假设的结果（Tatar 91）。

NoteCards 方法使用超文本作为共享表面。虽然提供了许多优势，但已观察到与 gIBIS 类似的分类问题（Shipman & Marshall 92, Monty 90）。用户很难将信息分块到卡片、命名卡和归档卡中。很少使用键入的链接（然后，不一致）。链接方向和链接语义也被证明是有问题的。许多这些困难再次源于与实际系统使用不符的设计假设。

3. 开发 DNP：迭一种方法

我们的工作以这些以前设计支持的方法的经验为基础，旨在解决他们发掘出的困难。虽然我们认识到需要以最终在开发过程的后期部分有用的形式表示和记录设计信息，但我们也承认要求设计师在设计过程中过早地做出特定解释的问题。事实上，任何制度的首要标准必须是可接受性（Grudin 88）。因此，我们的主要重点是开发一个界面，用户觉得易于操作，适合设计的早期阶段。我们没有设计理论，

我们希望强加或测试设计师，而是希望发现他们的要求。这导致我们采用基于快速原型的工具开发方法。

我们支持这样一种观点，即想法表示工具（如 Cognoter 或更一般的设计表面）在围绕设计过程的对话中提供了一种表示媒介（Tatar et al. 91）。先前的研究已经研究了传统白板和共享绘图介质在设计过程中的使用（Suchman 88, Tang89）。然而，电子绘图表面不同于这些传统媒体（Tatar et al. 91），并清楚地了解如何利用这些表面来表达设计概念是我们方法的关键部分。我们的核心问题是，表达媒介在表达中起着核心作用。事实上，我们认为电子系统为代表设计提供的可能性和这些陈述的性质是充分交织在一起的，两者都不能孤立地充分解决。因此，不仅对软件设计过程知之甚少，无论是一个人还是很多人，而且计算机系统的发展，以支持这个过程，使一些功能必然不可知。这是因为计算机化设计工具可能会改变设计过程的性质，因为文字处理器改变了写作的性质（Haas 89）。因此，我们采用了基于开发设施的迭一性开发方法，并与使用该系统的设计师密切合作。

我们首先提供一套核心设施，使设计得以表达。然后，设计师们长期使用这些设施来支持各种实际设计任务。观察并录下了这种使用，并邀请参与者评论他们在该系统方面的经验。会议的目的是：-。

1. 隔离用户界面功能引起的问题。
2. 检查系统的使用，并突出显示设计人员所需的其他功能。
3. 在添加到系统之前评估现有功能的有用性。
4. 提供有关设计过程本身的信息。原型处理方法在图 1 中以图表显示。

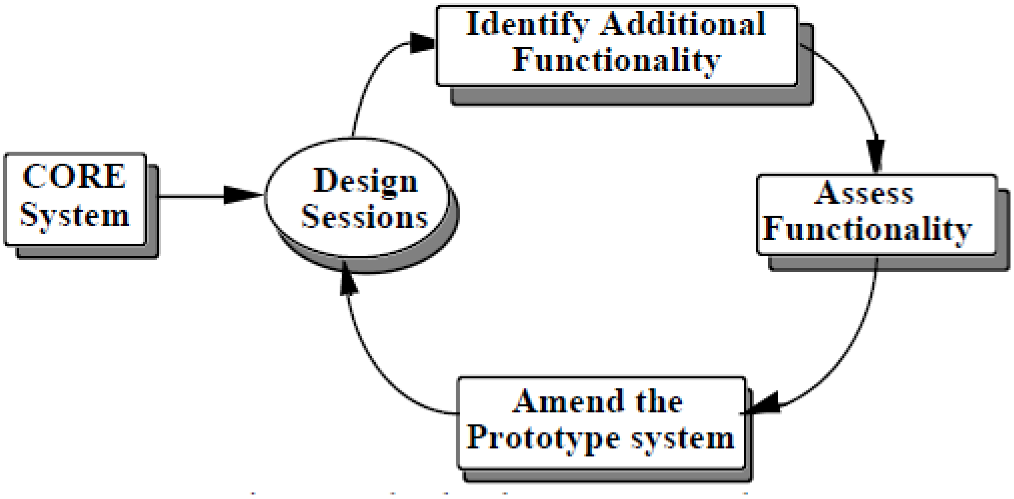


图 1。开发方法。

我们的方法与参与性设计（Bodker 91）有联系，与（Tang 91）提出的方法相似。它还从其他领域借款，包括 HCI（Hartson 91）和人工智能和教育（Twidale，in press）。后一个领域与《特定常规武器公约》有许多相似之处，既注重计算机系统人员的使用，又关注对人类活动（分别学习和工作）的支持，这些都理解不完全。在 AI&ED 中，观察到界面中的单个缺陷可能会对学习结果产生实质性的负面影响，但用于一个目的的界面功能也可能具有额外的有益目的（Twidale 91）。因此，界面的功能有可能淹没系统更深、更有趣、更复杂的模块的影响。

反复的发展方法有助于缓解这一问题。通过在简单的情况下测试系统的基本版本（例如，由一名设计师、两名设计师坐在同一终端或两名设计师，每个设计者与一个终端并排坐在一起），我们可以消除更严重的界面错误。当设计扩展到必然更复杂的案例（例如许多设计师在空间中分离且带宽通信有限）时，这可以告知设计。要使这种方法发挥作用，我们需要更多地关注接口困难，而不是成功。

我们可以相信，如果不采取任何行动，困难将扩大至更复杂的情况，而成功可能没有。

3.1 软件设计流程

我们主要关注的不是设计的所有方面作为一个抽象的过程，而是软件开发过程。这使我们能够对应用任何开发系统的设计设置做出特定的假设。特别是，软件设计过程的特点是：

结构的发展

软件设计与从部分定义、松散结构的想法和概念过渡到完全定义和结构化的设计描述密切相关。这些描述构成了后续发展的基本计划。这些设计描述中的大多数是使用某种形式的网络图在图表上指定的。这些图表显示了一些软件实体，这些实体链接来描述相应的关系。

小组工作

任何规模巨大的设计都是许多设计师的努力。一组软件设计师将在相当长的一段时间内以各种不同的方式合作，实现完整的设计。参与设计过程的参与者只是部分理解所涉及的工作模式和不同的合作形式。

这种设计过程的观点构成了我们构建 DNP 的背景。需要注意的是，软件设计只形成整体开发过程的初始阶段，支持设计活动的一些好处可能要晚得多才能在软件生命周期中实现。不过，为了获得这些好处，软件设计人员必须使用设计工具。通过与设计师的工作实践结合来满足设计师的需求，成为一个至关重要的问题。

3.2 设计会议

在调查一个与协作软件设计不太了解的域时，尽早与设计师联系至关重要。设计者认为无法使用的工具，甚至只是笨拙地使用，将不会被采用。

鉴于我们开发支持软件设计的系统必然是探索性的，因此我们不可避免地会犯错。我们的方法的目的是提高识别和纠正这些错误的速度和容易性。对与界面有关的错误给予了相当大的关注。显然，与任何系统的接口对其使用都有实质性影响。如果不对接口进行小心处理，使用问题可能不是由于对 CSCW 的支持的某些功能产生反作用或缺失，而仅仅是接口问题。当系统被单个用户使用时，可以识别出其中的许多问题，即使它用于协作工作：单个用户很可能也遇到多个用户遇到的困难。

正是由于这种简单性的原因，我们选择首先专注于一个系统，由两三个人同时使用一个单一工作站。已开发并测试了两个工作站的分布式版本，但大部分工作已在单个版本上进行。在分布式版本的情况下，我们有并排工作站。我们选择此是为了最大限度地提高参与者之间的潜在通信带宽。这样，观察到的任何问题都不能归因于带宽限制，因此，如果不补救，将扩展到真正的分布式使用。该方法旨在补充正在进行的研究，解决更复杂的协作形式（如远程同步工作（Clark & Scrivener 92）对技术和社会的具有挑战性的影响。

3.3 迷你 DNP

我们为设计师提供支持的出发点是开发一个最小的系统，配备一套称为迷你DNP 的简单核心设施。该系统受先前系统的影响，以支持单个用户的设计（Sommerville et al.1990）以及开发和使用该系统所获得的经验。小型 DNP 的本质是创建实体和它们之间的链接的一种手段（图 2）。

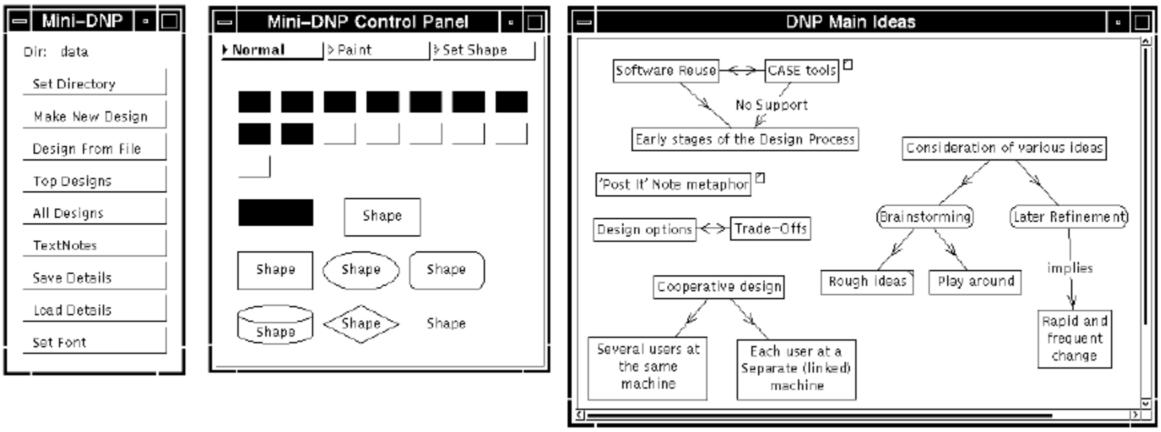


图 2。最小的 DNP 接口。

用户通过在设计窗口中键入创建实体。然后，他或她可以用鼠标移动它。链接是通过选择一个实体，然后单击要链接到的实体与转移键向下完成的。实体的这种链接构成了最终指定设计描述所需的图表。我们还提供了基于"后注"隐喻（见图 5）的文本说明，并允许设计人员将一个或多个注释附加到实体。这些可用于设计团队进行更多的文本评论、想法、意见、代码片段、引用等。提供各种文本注释类型，用户可以定义它们本身（包括表单结构）。带有文本说明的实体附有图标（例如图 2 中的实体"CASE 工具"），可以通过单击图标来检查注释。设计可以从文件中保存和加载，并创建包含设计的屏幕转储和实体列表及其文本说明的纸质报告。每个实体本身都可能扩展为子设计。打开新窗口，以正常方式创建实体和链接。子设计可能包含本身的子设计实体。使用颜色、形状和标签为实体

和链接提供松散的打字形式。用户控制 s/he 想要使用打字的程度。实体或链接的类型可以随时轻松更改。

初始系统具有某些特性，适合支持设计过程：

1. 支持创建定向图形的快速易用界面。这些可以在任意数量级别创建，从一个级别到另一个级别进行简单导航。
2. 未绑起的实体和关系。这在分析的早期阶段至关重要，因为将实体纳入类型等级结构是不现实的。
3. 广泛的注释设施，使系统实体和关系能够用结构化和非结构化文本进行注释。
4. 创作后类型归属。这些设施还允许轻松更改实体的类型。
5. 报告生成设施。

3.4 评估

我们使用各种用户来评估系统。鉴于目标是不断完善，人们最初可以使用开发团队的成员来测试系统。乍一看，他们似乎比大多数人更有动力使用该系统，并以有利的眼光来看待它，这似乎是一个问题。然而，只要他们有真正的任务(如系统本身的持续设计)，他们可能会发现一些更严重的界面错误。我们可以确信，此类错误将扩大范围：对于局外人来说，开发人员认为难以使用的功能肯定会很困难（当然，如果开发人员不喜欢定期使用该工具，则不太可能有人使用)。后来，我们尝试了不参与其开发的用户的系统，以便发现开发人员社区中关于易用性和所需功能的任何隐藏假设。

该系统已同时投入使用和开发约两年。在此期间，大量用户群体利用了该系统。表 1 总结了系统使用的各种内容。

由设计师：单一设计师

一个工作站的两名设计师

一个工作站的三名设计师

两个相邻工作站的两名设计师

按任务分组：软件设计

信息组织和浏览

项目管理

讲座课程设计

设计论文（包括本文）

设计会谈

按用户分类：系统开发人员

项目成员

博士生

社会学同事

访问学者

大学生

表 1。系统在测试中的使用性质。

在所有使用的情况下，无论是个人还是团体，我们要求我们的志愿者承担他

们必须承担的任务。我们认为，开发真实任务系统非常重要，因为这些系统具有在精心策划的任务中很难复制的功能。这些功能包括模糊性、开放性、历史性和用户参与度，理想情况下，用户将更关注问题而不是工具。我们给用户的指示是尝试使用该工具，询问在必要时如何实现他们想要完成的任何操作。我们的假设是，当工具变得令人沮丧时，它最多可以使用，因为它会阻止用户执行他/他想要的事情。我们可以确定他/她想做什么，然后可以评估应该向系统添加什么样的功能。

鉴于我们想要使用真实任务的愿望，我们有点受制于志愿者的需求。幸运的是，一个旨在支持软件开发早期阶段的工具对于许多其他创造性设计任务（如项目管理、制作人种学笔记以及论文和报告的设计（Sommerville et al. 93）同样有用。事实上，本文的初始结构和要素是使用 DNP 协同开发的。

一些系统使用是在一次性的设计会议，其他已经持续了若干会话，但其他会议（特别是那些涉及项目管理和研究活动的设计和管理）正在持续数月。该系统现在可从校园内多个不同地点访问。特别是，两个用户可以从其办公室访问 DNP 并每天使用它。后一种使用形式开始揭示随着时间的推移，管理设计复杂性的要求的性质。

4. DNP 中的设计支持

设计会议的目的是发现不同设计师使用 DNP 的各种模式（包括困难）。我们对使用的意见为适当支助设施的发展通报了情况。本节简要地突出了这一进程的结果。

4.1 使用差异

我们的研究已经确认了用户和同一用户之间在时间和环境下设计活动的巨大差异（见图 3、4、5 和 6）。例如，一些设计人员使用带有文本说明的非常单调的实体名称来包含详细信息，而另一些设计人员则使用短语或句子样名称。一些设计师使用许多链接来表示连接，而另一些设计师则使用二维接近（Marshall & Rogers 92）。有些使用大量的颜色和许多不同的形状，而另一些使用黑色矩形所有的时间。作为随时间变化的例子，我们观察到了涉及实体创建回合和粗略定位的周期。这些涉及非常快速和激烈的活动，其中使用最少的选择。在这样的回合后，有一个恢复期，显示屏被"整理"。创建链接并重新排列实体，通过接近其他实体来传达其他含义，并减少活动频繁的区域。

这种变异性确认了 DNP 中需要灵活性和易于修订。这种灵活性在很大程度上来自于我们决定避免将语义与实体、链接和子设计联系起来，并允许这些语义主要被用户用作表达方式。这些初始设计图的含义留给不同的设计师来解释。围绕实体生成的密集活动"爆发"使我们专注于支持实体的快速进入（只是键入和点击实体之间的回报生成以列表格式定位的条目）。一旦进入这些实体，可以通过改变它们的位置、颜色和形状来整理。例如，将图 3 中显示设计的早期阶段的设计与图 5 中显示的后期阶段进行对比。

4.2 结构的演变

以实体形式添加到记事本中的设计元素必然模棱两可：它们由几个字的标签组成，指的是一个概念。然而，这个标签不仅能够被任何人误解除了它的创造者

，但即使是创造者，似乎故意模棱两可的术语被选择（例如，实体通常被称为"对象"或"用户"）。

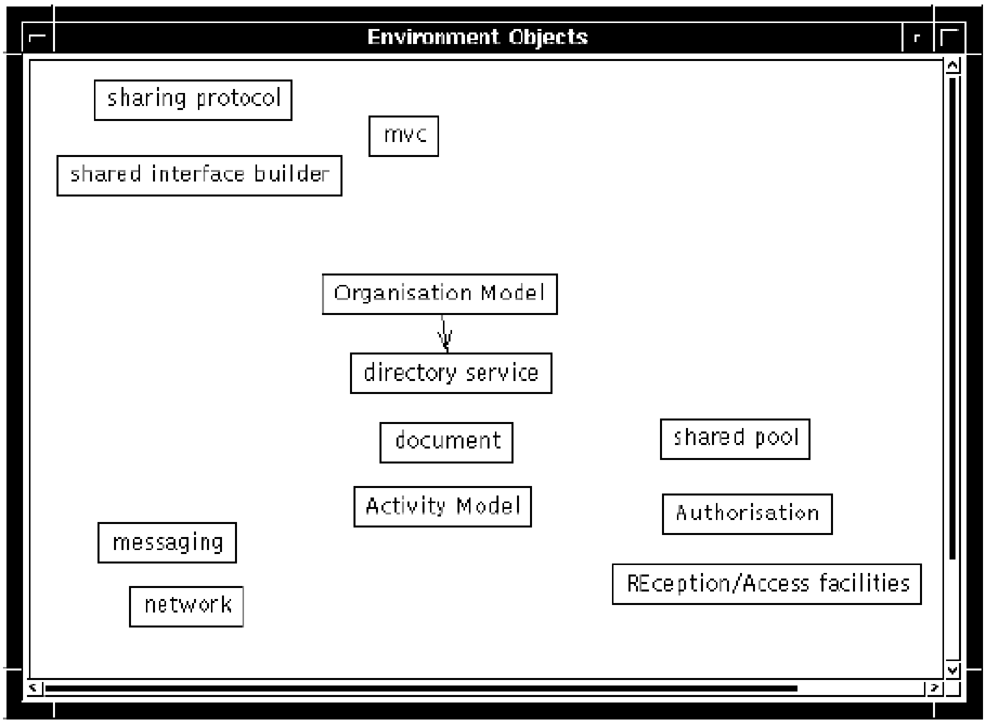


图 3。DNP 中的初始设计。

这是为了处理概述概念而推迟有关详细信息的决定的实例。在设计过程的后期，将完善初始概念的含义。这可以涉及通过编辑实体名称（包括完全更改名称）、在模棱两可的描述中为组成概念创建更多实体、通过附加文本说明添加更多信息、为实体创建子设计或完全替换实体来增加更多信息。

设计师在设计过程中似乎需要有一定的模糊性，以免被细节所拖累。渐渐地，这些模棱两可之处得到解决和改进。在其他情况下（Marshall & Rogers 92）以及用户希望避免正规化的原因（Shipman & Rogers 92）也观察到了这一点。以类似的

方式，链接的含义会随着时间而改变。最初，它的意思可能是"这些实体在某种程度上是相互关联的"。最终，这被提炼成更精确的含义。精度的逐渐演变通常与链接键入的使用有关：具有类似含义的链接现在被赋予相同的颜色或标签。

实体和链接的属性在 DNP 中用作支持结构演变的机制。使用颜色和形状作为键入实体和链接的手段由 DNP 直接支持。创建这些不同的实体属性和链接后，用户可以随着时间的推移从图 2 中显示的控制面板进行修改。这种自由允许在实体创建后在设计中出现适当的结构，并允许实体类型在实体创建后发生变化。图 3 和图 5 说明了这种迁移。这与以前的系统形成鲜明对比，后者支持一组类型的定义，但要求设计人员在创建时为每个实体选择类型。

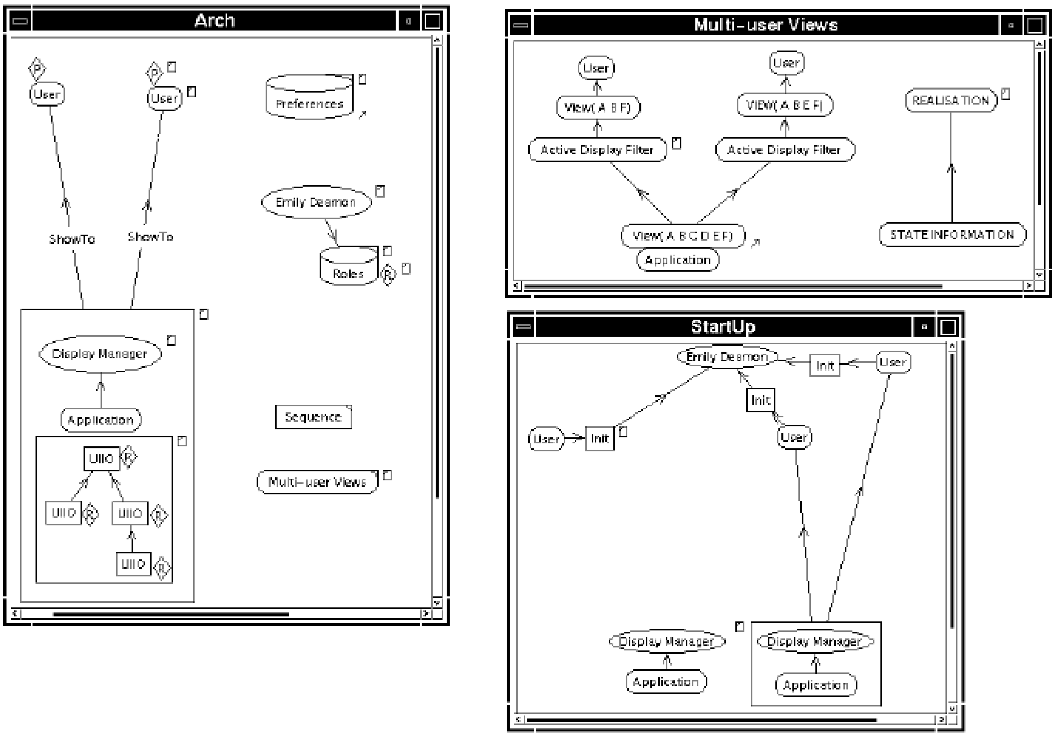


图 4。实体的安排、框架和分组。

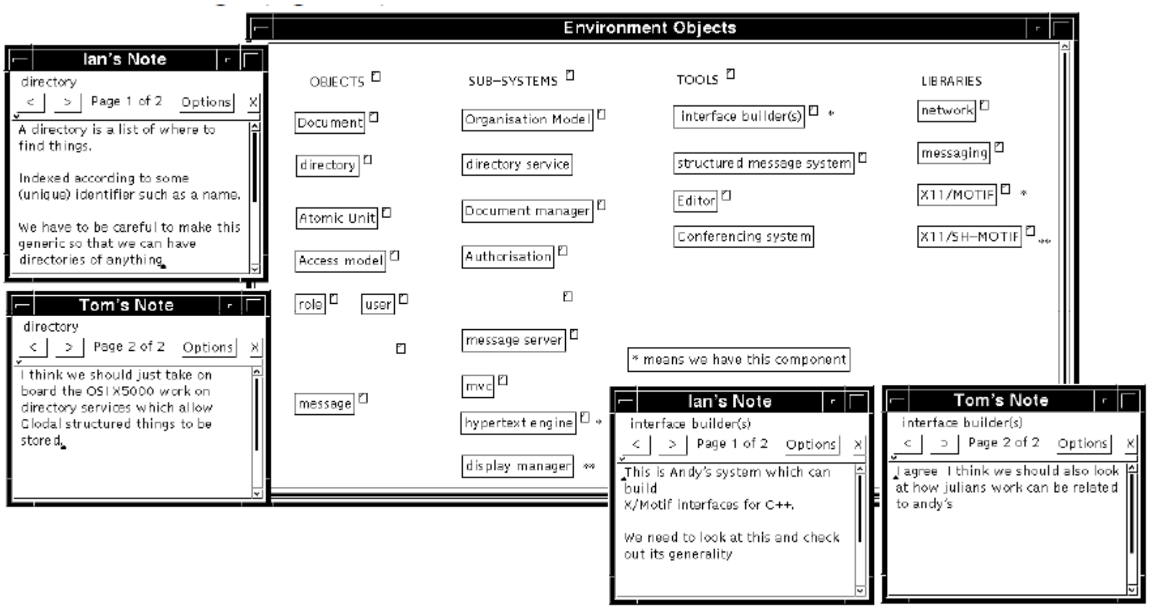
右边的两个设计是左边的子设计。此设计使用了 11 种颜色

任何非平凡设计的实体之间的关系极其复杂。除了链接，空间通常用于区分这种关系，因为适当的实体是聚类的（Marshall & Rogers 92）。用户表示需要从这些分组中抽象出来，一旦它们被制作出来。子设计用于促进这一点。用户可以将实体框成一组。然后，S/he 可能会将组"推倒"到子设计中，由单个实体替换（图 4）。

4.3 合作和一次性使用

合作和一次性使用之间最明显的区别是速度：单个设计人员进入、改进和修改项目的速度远远快于组。这主要是因为小组参与者需要向其他人证明和解释他们的行为。任何操作都不是完全明确的，单个用户可以隐含功能的程度自然远远低于必须共享想法时。不仅需要作出更大的解释和细化，而且必须谈判修订和增加等行动，从而就活动的适当性展开辩论。相比之下，单个设计人员创建实体和链接，并重新排列它们，有时几乎令人费解的速度。我们认为这是需要该工具来支持 DNP 提供的设计频繁和快速修订的证据。

键盘和鼠标的共享总体上似乎并没有造成很大的困难，用户轮流输入文本或重新排列实体。键盘的控制可以涉及权力的位置，暂时控制协作，也可以涉及秘书功能，从而进行其他人的审议。在 DNP 的分布式版本中，尽管相关机器非常接近，但用户利用设计的共享视图快速划分设计活动，并独立于整体设计的不同部分工作。大量使用文本说明来注释其他用户设计的功能（图 5）。



A directory is a list of where to find things.

Indexed according to some (unique) identifier such as a name.

We have to be careful to make this generic so that we can have directories of anything.

I think we should just take on board the OSI X5000 work on directory services which allow Clodal structured things to be stored.

This is Andy's system which can build X/Motif interfaces for C++.

We need to look at this and check out its generality

I agree I think we should also look at how julians work can be related to andy's

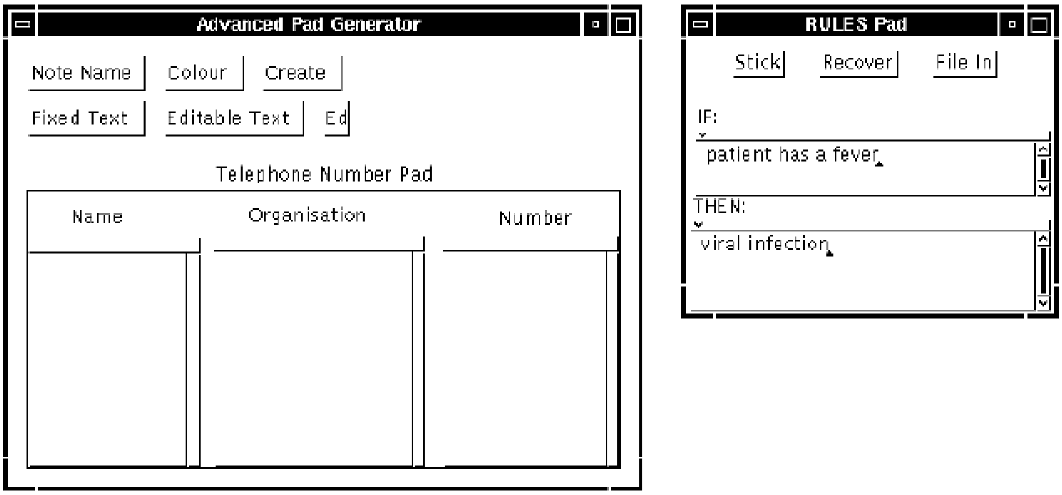
图 5。使用文本说明来评论设计。

此设计使用了 7 种颜色。

在这两种情况下，合作的一个有趣的特点是如何移动实体并突出强调，作为辩论的一部分。当前控制鼠标的用户将选择他希望讨论的实体，从而突出显示该实体。S/he 可能会稍微移动此实体（从一侧移动到另一侧或小圆圈）作为强调的手段。移动实体以强调的这种活动可以通过定位一个实体更接近另一个实体来与移动更实质性的场合形成对比，从而表明某种语义特征。这可能是永久性的，或者仅仅是一个持续的运动作为一个建议。除了使用鼠标外，还观察到更直接的交互，即用户用手指指向屏幕。合作情况还再次利用了 DNP 支持意义演变的能力。通常，组设计活动的参与者会使用实体作为参数占位符，并使用这个接受的设计实体来回溯到以前的辩论。

由于我们对小组的观察，与放置文本笔记有关的设施得以扩展，使用户能够轻松地利用文本笔记作为注释方法。这些扩展允许为可与不同用户和/或目的关联的

新纸币类型的定义提供设施。图 6 显示了创建具有用户定义结构的新笔记板的工具，以及另一个创建的记事本。提供设施以过滤和突出不同设计师的笔记。



IF:
patient has a fever
THEN:
viral infection

图 6。用于创建结构化笔记和用户定义记事本的生成器。

5. 结论

迭一次开发方法适合 CSCW 系统开发，因为它有助于我们需要在开发系统的同时更多地了解合作活动的性质。设计师记事本早期版本的测试强调了设计师之间和根据情况在设计活动中的多变性。我们还注意到概念随着时间而变异导致结构演变的方式。这可能导致频繁和快速的修订。易于修订对于鼓励集思广益，使概念必然不完整。"支持设计早期阶段的系统需要支持流程的这些功能。通过反复开发，我们能够纠正更严重的界面错误，这些错误有可能淹没有关设计和合作工作的更深层次问题的影响。这种方法还使我们能够从人种学研究产生的结果流中获利。在目前的状态下，设计师记事本是获取有关设计活动信息的工具，也是支持该活动的工具。

6. 确认

迈克尔·特威代尔是科学与工程研究理事会初级研究员。设计师记事本的开发部分由认知科学和COMIC联合理事会倡议(Esprit basic research project 6225)供资。作者要感谢他们在社会学系的同事：约翰·休斯和瓦尔·金在整个项目中给予的支持和建议。

作者: Michael Twidale, Tom Rodden, Ian Sommerville Department of Computing, Lancaster University, U.K.

国籍: U.K.

从起源 : Researchgate.net DOI: 10.1007/978-94-011-2094-4_7

指导教师(Advisor)：傅由甲

年(Year) 2021 月(Month) 06 日(Day) 5