



واحد تهران شمال

سمینار کارشناسی ارشد

چارچوب چندلایه بر اساس خواسته‌ها برای تشخیص نیازهای غیر ساختاری کاربران از سیستم نرم‌افزاری

استاد راهنما:

سید علی رضوی ابراهیمی

چنور حسینی

شهریور 1400

چکیده

نیازهای غیر عملکردی[[1]](#footnote-1) تأثیر بسیار زیادی بر نحوه کارکرد یک نرم‌افزار دارند، درواقع می‌توانیم این نیازها را نیازهای کیفی در نظر بگیریم و با اطمینان بگوییم سیستم نرم‌افزاری که در زمان توسعه و پیدایش به این نیازها بی‌توجه بوده کارا نیست و با شکست مواجه خواهد شد. کارهایی که اخیراً در زمینه‌ی مهندسی نرم افزار و در مرحله نخست تولید نرم‌افزار یعنی تجزیه‌وتحلیل نیازمندی‌ها انجام‌شده نشان داده‌اند که ما می‌توانیم با ترکیب کردن اهداف مدنظر کاربران و نحوه تعامل آن‌ها با نرم‌افزار تا حدود زیادی به نیازهای کیفی لازم برسیم. دستیابی به این نیازها به‌سادگی صورت نمی‌گیرد و برای این کار باید به تمام خواسته‌های کاربران شامل خواسته‌های صریح و خواسته‌های ضمنی دست پیداکرده هم‌چنین روابط متقابل بین کاربران، نیازهای به‌دست‌آمده، عوامل سیستم و سناریوها به‌طور دقیق بررسی شوند. در این سمینار که براساس مرجع[1] نوشته‌شده است نحوه به دست آوردن نیازهای غیر عملکردی با استفاده از اهداف و خواسته‌های کاربران و روابط بین آن‌ها بررسی‌شده است؛ در این تحقیق از یک مدل آماری تحت چارچوب سیتو[[2]](#footnote-2) به نام زمینه تصادفی مشروط[[3]](#footnote-3) استفاده‌شده که توسط این مدل خواسته‌های به‌دست‌آمده کاربران با توجه به میزان اهمیت آن‌ها در سطوح مختلف دسته‌بندی می‌شوند، این خواسته‌ها و روابط میان آن‌ها نیازهای غیر عملکردی را نتیجه می‌دهند.

کلمات کلیدی: نیازهای غیر عملکردی، اهداف، روابط بین خواسته‌ها، مهندسی نرم‌افزار

فهرست مطالب

**صفحه عنوان**

فصل اول: کلیات طرح تحقیق 1

1-1 مقدمه 2

2-1 سابقه و ضرورت انجام تحقیق 3

فصل دوم: ادبیات و پیشینه تحقیق 5

1-2 مهندسی الزامات 6

2-2 استخراج الزامات غیر عملکردی 6

3-2 چارچوب سیتو 7

4-2 آگاهی از موقعیت (موقعیت‌آگاهی) 9

5-2 مدل‌های یادگیری تحت نظارت متوالی 10

فصل سوم: متدولوژی 11

1-3 روش تحقیق 12

2-3 جامعه و نمونه آماری 16

3-3 ابزار اندازه‌گیری 17

فصل چهارم: تجزیه‌وتحلیل 22

1-4 تجزیه‌وتحلیل **Error! Bookmark not defined.**

فهرست منابع 24

فهرست اشکال

عنوان صفحه

شکل (3-1): فلوچارت نحوه به دست آوردن نیازهای غیر عملکردی 14

شکل (3-2): رابطه سلسله مراتبی بین نیاز یافتن همکار و نیازهای سطح پایین آن 18

شکل (3-3): رابطه بین خواسته رسیدن به شهرت و خواسته به اشتراک گذاشتن کار 18

فهرست جداول

عنوان صفحه

[جدول (3-1) : روابط ایجاد شده بین نیازها از طریق سیستم زمینه تصادفی مشروط 14](#_Toc83089601)

[جدول(3-2) :نمونه داده های آموزشی برای نرم افزار زمینه تصادفی مشروط 19](#_Toc83089602)

[جدول( 3-3 ): نمونه الگوهای محاسبه شده برای خواسته های سطح پایین 20](#_Toc83089603)

[جدول (3-4) :وزن بدست آمده برای خواسته های سطح بالا و سطح پایین 20](#_Toc83089604)

**فصل اول**

**کلیات طرح تحقیق**

1-1 **مقدمه**

در مراحل طراحی و تولید نرم‌افزارها یکی از اصلی‌ترین و مهم‌ترین مرحله کار تشخیص نیازمندی‌ها است، تقریباً 50 تا 60 درصد از خطاهای موجود در توسعه نرم‌افزار ناشی از تشخیص نادرست است [2] و این باعث می‌شود هزینه اصلاح نرم‌افزارها تا 100 برابر افزایش پیدا کند [2]

نیازمندی‌ها درواقع مواردی هستند که کاربران برای انجام و برطرف کردن آن‌ها به سیستم نرم‌افزاری نیاز دارند، در بسیاری از موارد این مرحله به یک مصاحبه میان کاربر یا کارفرما و مهندس نرم‌افزار خلاصه می‌شود مصاحبه‌ای که احتمال دارد چند جلسه هم به طول بیانجامد اما در طول آن تنها خواسته‌های صریح کارفرما مدنظر قرار می‌گیرد، به این خواسته‌ها نیازمندی‌های عملکردی می‌گویند که اغلب پیاده‌سازی آن‌ها جنبه بصری دارد مثلاً یک کاربر از نرم‌افزار طراحی‌شده این انتظار را دارد که در زمان اتصال به پایگاه داده تعداد رکوردهای موجود در پایگاه را به وی نشان دهد و این‌یک نیاز عملکردی است اما اینکه کاربر بتواند در مدت‌زمان قابل قبولی به رکوردهای مختلف دسترسی داشته باشد یک نیاز غیر عملکردی است. این نیاز در مرحله ابتدایی تولید نرم‌افزار یعنی تشخیص نیازمندی‌ها به‌صورت صریح بیان نمی‌شود اما در کارکرد نرم‌افزار تأثیر به سزایی دارد. علاوه بر این، از دست دادن یک نیاز غیر عملکردی احتمالاً باعث می‌شود بسیاری از نیازهای دیگر کاربران نیز نادیده گرفته شود بنابراین، مهم است که تا حد ممکن نیازهای غیر عملکردی را در مراحل اولیه تجزیه‌وتحلیل نیازها به دست آوریم.

این سمینار بر پایه این نگرش استوار است که نیازمندی‌های مدنظر یک سیستم نرم‌افزاری خواسته‌های کاربران هستند که به‌شدت با اهدافی که برای پیاده‌سازی سیستم بیان‌شده پیوند خورده‌اند و می‌توانیم با دانستن هدف انجام یک کار، به خواسته‌ها و نیازهای پشت این هدف دست پیدا کنیم و از طریق آن‌ها نیازهای غیر عملکردی هم به دست می‌آیند. پس هدف از انجام این تحقیق رسیدن به‌تمامی خواسته‌ها و نیازهای کاربران از سیستم نرم‌افزاری است. درک و تجزیه‌وتحلیل درست این نیازها می‌تواند به تصمیمات طراحی بهتر [3]، تغییرات موردنیاز [4] و تکامل معماری [5] کمک کند.

در ادامه این نوشته مساله‌ای که با آن روبرو هستیم را بیان کرده، سابقه و ضرورت تحقیق نیز ذکرشده است و در فصل دوم اصطلاحات به‌کاررفته را توضیح داده و تاریخچه مختصری از موضوع موردمطالعه خود عنوان کرده‌ایم، در فصل سوم روش کار و متدولوژی به‌کاررفته توضیح داده‌شده است.

2-1 سابقه و ضرورت انجام تحقیق

دریکی از طبقه‌بندی‌هایی که برای نیازهای غیر عملکردی بیان‌شده قابلیت استفاده، امنیت و عملکرد سیستم‌های نرم‌افزاری مانند سیستم‌های اطلاعاتی، سیستم‌های بلادرنگ و سیستم‌های مبتنی بر وب جزو مهم‌ترین نیازها هست که توسط غازی و همکاران[[4]](#footnote-4) انجام‌شده است [6]. در طبقه‌بندی دیگری توسط افرین و همکاران[[5]](#footnote-5)، قابلیت استفاده، عملکرد، قابلیت اطمینان، امنیت و قابلیت نگهداری را به‌عنوان نیازهای غیر عملکردی متداول نام‌برده‌اند [2].

طبق مطالعه اخیر اکهارت و همکاران[[6]](#footnote-6)، 530 دسته‌بندی برای نیازهای غیر عملکردی از 11 موردنیاز سنجی صنعتی استخراج‌شده است که 75 درصد آن نیازهای عملکردی هستند یعنی حتماً باید موردتوجه قرار بگیرند و در هنگام توسعه نرم‌افزار کامپیوتری این نیازها برطرف شوند. [7]

استخراج الزامات غیر عملکردی می‌تواند کاری سخت و زمان‌بر باشد. برخی از محققان با استفاده از تکنیک‌های پردازش زبان طبیعی[[7]](#footnote-7) بر اساس اسناد تولیدشده برای راهنمایی کاربر و اطلاعات به‌دست‌آمده در زمان تجزیه‌وتحلیل نیازمندی‌ها و داده‌هایی که از سیستم جمع‌آوری‌شده است به این پی برده‌اند که استخراج نیازهای غیر عملکردی کار بسیار سختی است.[8] رویکردهای مبتنی بر پردازش زبان طبیعی اغلب فرض می‌کنند که مستندات موردنیاز به‌وضوح بیان‌شده است درحالی‌که بسیاری از محصولات نرم‌افزاری که در دنیای واقعی استفاده می‌شوند به دلیل تکنیک‌های نیازسنجی ناکافی فاقد مستندات موردنیاز هستند و این منجر به شکست استخراج نیازهای غیر عملکردی دقیق و مفید از محتوای متنی در عمل می‌شود. پس نمی‌توانیم با این روش به نیازهای غیر عملکردی دست پیدا کنیم.

یکی دیگر از روش‌های نیازسنجی، مهندسی الزامات هدف‌گرا[[8]](#footnote-8) است که اهداف را موردتوجه و بررسی قرار می‌دهد و نیازمندی‌های سیستم را با استفاده از اهداف تعیین‌شده برای آن در سطوح مختلف انتزاعی بیان می‌کند. [9]

در یک فرایند هدف‌گرا، فعالیت‌های مهندسی الزامات با در نظر گرفتن روابط بین اهداف، عوامل و سناریوها بیان می‌شوند. [9]

در این روش دیدگاه کاربر نقش کم‌رنگی دارد چون با توجه به اهداف از پیش تعیین‌شده که اکثراً کارفرما آن را تعیین می‌کند، روند کار شروع می‌شود و به این دلیل که نیازهای غیر عملکردی کیفیت انجام کار موردنظر از دید کاربران را بیان می‌کند پس‌ازاین روش نیز نمی‌توان نیازهای غیر عملکردی را به‌خوبی به دست آورد.

یکی دیگر از کارهایی که در این زمینه صورت گرفت و این نیازها را از دید توسعه‌دهندگان موردمطالعه قرارداد، توسط ژو و همکاران[[9]](#footnote-9) [10] انجام شد. آن‌ها پست‌ها و مباحث مربوط به یک وب‌سایت پرسش و پاسخ، به نام سرریز پشته[[10]](#footnote-10) را به‌عنوان ورودی متن در نظر گرفتند و از مدل موضوعی برای استخراج نیاز غیر عملکردی از روی این ورودی‌ها استفاده کردند. بااین‌حال، محدودیت روش آن‌ها این بود که نیازهای کاربران نهایی را نمی‌توان به‌راحتی با نظرات مهندسان سیستم برطرف کرد و هم‌چنین آماده‌سازی پست‌های مناسب که بتوان از روی آن‌ها و نظرات بیان‌شده برای آن‌ها تجزیه‌وتحلیل را انجام داد، به مدت‌زمان طولانی غیرقابل‌پیش‌بینی نیاز داشت. علاوه بر این، معنای واقعی از متن بحث را نمی‌توان با تکنیک مبتنی بر مدل موضوع نشان داد.

مطالعات انجام‌شده تاکنون به یک چارچوب توسعه نرم‌افزاری به نام سیتو منجر شده است که نیازها را از دید کاربرها بررسی می‌کند. [11]

این روش برای بیان هر نیاز یک موقعیت در نظر می‌گیرد و روش تائید شده برای به دست آوردن نیازهای غیر عملکردی روش سیتو می‌باشد.

فصل دوم

ادبیات و پیشینه تحقیق

در این فصل با اصطلاحات و عبارات بکار رفته در این تحقیق آشنا می‌شویم و در هر توضیح کارهایی که تاکنون درزمینه‌ی این موارد انجام‌شده را نیز بیان می‌کنیم:

1-2 مهندسی الزامات[[11]](#footnote-11)

مهندسی الزامات را می‌توان فرایندی برای کشف، مستندسازی و مدیریت الزامات سیستم‌های کامپیوتری دانست. [12] هدف مهندسی الزامات ایجاد مجموعه‌ای از نیازمندی‌های سازگار، کامل و مرتبط است و آنچه را که مشتری واقعاً به آن احتیاج دارد بازمی‌گرداند. [12] فرایند سنتی استخراج الزامات یک فرایند یادگیری تعاملی بین مهندسان و کارفرمایان است [13]که از طریق مصاحبه، مشاهده فعالیت‌های مشتریان و بازخورد عملکرد سیستم، می‌توان اطلاعات مربوط به نیازهای کاربر را به دست آورد [14]

مراحل استخراج نیازمندی‌ها بیشتر به حالت ذهنی مهندس متکی می‌باشد و احتمال بروز خطا در آن زیاد است. پس از تولید و راه‌اندازی سیستم نرم‌افزاری نیز تغییر دادن آن برای برطرف کردن نیازهایی که پس از اجرا خود را نشان می‌دهند کار بسیار زمان‌بر و پرهزینه‌ای است [11].بنابراین، ضروری است که محققان رویکردهای سیستماتیک را بر اساس اصول مهندسی مطالعه کنند که منجر به الزامات بهتر و جامع‌تر شدن فرایند استخراج الزامات شود. به این رویکردها مهندسی الزامات می‌گویند.

2-2 استخراج الزامات غیر عملکردی[[12]](#footnote-12)

بسیاری از روش‌های اصلی برای استخراج الزامات یا نیازهای غیر عملکردی بر اساس تکنیک‌های مبتنی بر استخراج متن و پردازش زبان طبیعی است.[9]

تمامی آثار قبلی که در زمینه استخراج نیازمندی‌های سیستم نرم‌افزاری کارکرده‌اند پیشنهاد کرده‌اند که در اسناد مختلف تهیه‌شده برای سیستم نرم‌افزاری الزاماً نیازهای غیر عملکردی وجود داشته و بایستی در نظر گرفته شوند. آن‌ها معتقدند که می‌توان با در نظر گرفتن کلمات کلیدی در این اسناد به‌عنوان شاخص انواع مختلف نیازها را شناسایی کرد این کلمات کلیدی برای طبقه‌بندی نیازهای غیر عملکردی استفاده می‌شود و می‌بایست از اسناد جمع‌آوری‌شده در مرحله تجزیه‌وتحلیل نیازمندی‌ها به دست بیایند.

سلانکاس و همکاران[[13]](#footnote-13) [8] یک روش خودکار برای استخراج نیازهای غیر عملکردی از انواع اسناد نرم‌افزاری ازجمله مشخصات موردنیاز، توافقنامه‌های داده، راهنمای کاربر و غیره پیشنهاد کردند. جملات موجود در آن اسناد با بردارهای کلمه نشان داده‌شده و در 14 دسته طبقه‌بندی شدند. آن‌ها از مدل‌های یادگیری ماشین مانند کی ان ان[[14]](#footnote-14)، نیو بایس[[15]](#footnote-15) و حداقل بهینه‌ساز متوالی[[16]](#footnote-16) برای طبقه‌بندی هر جمله جداگانه استفاده کردند و نتایج نشان داد که روش حداقل بهینه‌ساز متوالی در طبقه‌بندی جملات بر دیگر مدل‌ها برتری دارد.

3-2 چارچوب سیتو

سیتو روشی است که درواقع موقعیت هر نیاز در نظر گرفته می‌شود و آن را با یک سه‌گانه تعریف می‌کند.

در ادبیات رایانه، تعاریف مختلفی از موقعیت وجود دارد:

1. یک موقعیت می‌تواند یک موقعیت اتمی، یک موقعیت ترکیبی منطقی یا یک موقعیت گذرا باشد. یاو و همکاران[[17]](#footnote-17) [16]
2. درک عناصر موجود در محیط در یک‌زمان و مکان، درک معنای آن‌ها و پیش‌بینی وضعیت آن‌ها در آینده است. میکا اندسلی[[18]](#footnote-18) [17]
3. یک موقعیت در زمان t سه‌گانه فعالیت‌های یک کاربر است، مجموعه‌ای از اقدامات و زمینه‌های دستگاه‌ها در یک دوره زمانی است داکی لی و سومی هلال[[19]](#footnote-19) [18]
4. وضعیت در زمان تی[[20]](#footnote-20)، یک سه‌گانه {دی، آ و ای} است. چانگ و همکاران[[21]](#footnote-21) [11]

چارچوب سیتو از روش چانگ و همکاران استفاده می‌کند و وضعیت را به‌صورت سه‌گانه (دی، آ و ای)[[22]](#footnote-22) در یک‌لحظه به‌خصوص در نظر می‌گیرد.

زمینه ذهنی[[23]](#footnote-23) (دی) به حالات روانی انسان اشاره دارد که تاحد زیادی پنهان است، زمینه رفتاری[[24]](#footnote-24) (آ) چگونگی رفتار را مشخص می‌کند و زمینه محیطی[[25]](#footnote-25) (ای) نشان‌دهنده اطلاعاتی است که می‌توان از محیط به دست آورد مانند نور، دما، مکان، سرعت و غیره. چانگ و همکاران همه عناصر لازم برای توصیف کامل یک موقعیت شامل اجزای لازم برای محاسبه وضعیت از دیدگاه علوم کامپیوتر را تعریف می‌کنند.[11] بنابراین، ما از تعریف وضعیت چانگ و همکاران در این سمینار استفاده می‌کنیم.

در این چارچوب نیازهای سیستم را به دودسته تقسیم کرده‌اند، نیازهای مرکب و اتمی.

چارچوب سیتو فرض می‌کند که خواسته‌های انسان را می‌توان به‌صورت سلسله مراتبی و به شکل یک درخت نشان داد که در آن نیازمندی مرکب یعنی نیازمندی که می‌توانیم آن را به نیازهای دیگری تجزیه کنیم در ریشه قرار می‌گیرد.]12] اما نیازمندی‌های موجود در برگ‌های این درخت اتمی هستند و نمی‌توان آن‌ها را تجزیه کرد. [11] در این تحقیق، نیازمندی‌هایی که به سطح ریشه درخت نزدیک‌تر هستند نیازهای سطح بالا و آن‌هایی که به برگ‌ها نزدیک‌تر هستند و خود برگ‌ها نیازهای سطح پایین هستند.

یک نمونه از خواسته سطح پایین می‌تواند «به اشتراک گذاشتن یک کار» در سیستم نرم‌افزاری همکاری تحقیقاتی باشد؛ یک خواسته سطح بالا می‌تواند «به دست آوردن شهرت» در این سیستم همکاری تحقیقاتی باشد.

درخت خواسته در چارچوب سیتو شبیه درخت هدف در مهندسی الزامات هدف‌گرا است. تفاوت عمده این است که یک درخت هدف از جنبه سیستم‌ها برای جلب نیازهای کاربر مورد تجزیه‌وتحلیل قرار می‌گیرد، درحالی‌که یک درخت خواسته از جنبه کاربر برای نظارت بر اهداف آن‌ها تجزیه‌وتحلیل می‌شود [11]. چارچوب چندلایه‌ای که در این سمینار ارائه‌شده است مطابق با ساختار درخت خواسته‌ها در چارچوب سیتو می‌باشد زیرا کاربر محور است و نیازهای کاربران را در یک ساختار سلسله مراتبی در نظر می‌گیرد.

4-2 آگاهی از موقعیت (موقعیت‌آگاهی)[[26]](#footnote-26)

با کار در زمینه نیازمندی‌های سیستم‌های نرم‌افزاری با دو اصطلاح به نام‌های آگاهی از موقعیت و آگاهی از شرایط آشنا می‌شویم که این دو مورد باهم تفاوت دارند. آگاهی از شرایط، انسان‌ها را کمتر در نظر گرفته و داده‌ها را در اولویت قرار می‌دهد. برعکس آگاهی از موقعیت، مستلزم ارائه داده‌های متنی به انسان‌ها است و کاربران را در اولویت قرار می‌دهد.

پس از استقرار یک برنامه نرم‌افزاری، داده‌هایی که از کاربران واقعی در زمان استفاده از سیستم به‌دست‌آمده و به‌روز هستند، نقش مهمی در تکامل سیستم ایفا می‌کنند. حالتی را در نظر بگیرید که دو فرد در یک زمینه رفتاری یا محیطی دارای حالات روانی کاملاً متفاوتی هستند پس موقعیت‌های متفاوتی تجربه خواهند کرد. و این حالات روانی و کل تأثیراتی که کاربران از محیط می‌گیرند آگاهی از موقعیت نامیده می‌شود.

در زمان نیازسنجی سیستم‌های نرم‌افزاری تجزیه‌وتحلیل دیدگاه کاربران از اهمیت ویژه‌ای برخوردار است به همین دلیل است که از مدل زمینه تصادفی مشروط برای استخراج الزامات استفاده می‌شود و این مدل نیز بر طبق چارچوب سیتو بوده و حالت ذهنی کاربر، شرایط محیطی و چگونگی رفتار کاربر در موقعیت مدنظر، از پارامترهای اصلی آن است.

بااین‌حال، باید به این موردتوجه کرد که این روش به‌دقت یادگیری مدل زمینه تصادفی متکی است و با ورودی داده‌های نادرست نتایج نادرستی به دست می‌آیند همچنین روش پیشنهادی بسیار متکی به داده‌های مشاهده‌ای است بنابراین خطرات مربوط به شرایطی که نادیده گرفته‌شده است می‌تواند قطعی نبودن ارزیابی الزامات را افزایش دهد.

5-2 مدل‌های یادگیری تحت نظارت متوالی

در بسیاری از الگوریتم‌های طبقه‌بندی کلاسیک، داده‌های آموزش به‌طور مستقل از توزیع مشترک گرفته می‌شود. بااین‌حال، در داده‌های متوالی، داده‌های نزدیک ارتباط تنگاتنگی باهم دارند [18]؛ بنابراین، برای یادگیری و تجزیه روابط بین مشاهدات مختلف و برچسب‌های استنباط شده در یک دنباله مشاهده، یک مدل یادگیری مناسب موردنیاز است.

با استفاده از الگوهای ویژگی انعطاف‌پذیر، زمینه تصادفی مشروط این مزیت را دارد که می‌تواند روابط بین بخش‌های داده متوالی مرتبط را ضبط کند؛ یعنی توابع زمینه تصادفی مشروط می‌توانند پیوستگی میان مقادیر را نیز در نظر بگیرند.

ازنظر دقت در تبدیل گفتار به متن[[27]](#footnote-27) مشکلاتی که در این زمینه وجود دارد، نام‌گذاری یک موجودیت شناسایی‌شده[[28]](#footnote-28) و تشخیص دست خط[[29]](#footnote-29) بوده و ثابت‌شده است که زمینه تصادفی مشروط از مدل‌های مخفی مارکوف][[30]](#footnote-30)7] و حداکثر مدل‌های آنتروپی مارکوف[[31]](#footnote-31) [19] بهتر عمل می‌کنند. این روش همچنین در تشخیص الگو و یادگیری ماشین [20]، داده‌های توالی بیولوژیکی [21] و تشخیص نفوذ[22] استفاده‌شده است. در مقایسه با مدل‌های مخفی مارکوف، زمینه تصادفی مشروط در تفسیر روابط بین خواسته‌ها، رفتارها و ارزش‌های سیستمی در چارچوب سیتو کارآمدتر است زیرا قادراست با استفاده از توابع ویژگی قابل‌استفاده به‌جای ماتریس احتمال، بر محدودیت‌های مدل‌های مخفی مارکوف غلبه کند.

فصل سوم

متدولوژی

1-3 روش تحقیق

در این سمینار برای دستیابی به اهداف غیر عملکردی از زمینه تصادفی مشروط استفاده‌شده است. زمینه تصادفی مشروط درواقع یک روش آماری تشخیص نیازمندی‌ها می‌باشد. نسخه زمینه تصادفی مشروط استفاده‌شده ( زمینه تصادفی مشروط++)[[32]](#footnote-32) است زیرا با بررسی کردن دقت آن می‌بینیم که نسبت به سایر الگوریتم‌های زمینه تصادفی مشروط ماننداس جی دی، واپیتی[[33]](#footnote-33)، مالت[[34]](#footnote-34) و... دقت بالاتری دارد.

استخراج نیازهای غیر عملکردی در دو فاز صورت می‌گیرد:

1. به دست آوردن نیازها با استفاده از مدل زمینه تصادفی مشروط

2. به دست آوردن نیازهای غیر عملکردی جدید با آنالیز و تجزیه‌وتحلیل روابط بین خواسته‌ها و در نظر گرفتن انتزاع‌های متفاوت.

مراحل کلی کار به‌صورت زیر است:

مرحله اول کار ایجاد یک پایگاه دانش است که با توجه به رویدادی که در حال بررسی آن هستیم و نیازسنجی که در ابتدا توسط مهندس نرم‌افزار به طرق مختلف (مصاحبه، پرسشنامه، مشاهده عملکرد سیستم و... ) انجام‌شده است می‌توانیم اطلاعات به‌دست‌آمده را در قالب جدول در این پایگاه دانش ذخیره کنیم. مطابق با نیازهای به‌دست‌آمده باید نوع فیلدهای جدول پایگاه دانش و تعداد ستون‌های آن را تعیین کنیم.

مرحله دوم آموزش دادن به مدل زمینه تصادفی مشروط است که توسط اطلاعات اولیه و جدول به‌دست‌آمده از مرحله قبل این کار انجام می‌شود. مدل زمینه تصادفی مشروط دو نوع الگوی متفاوت دارد: یونی گرام[[35]](#footnote-35) و بایگرام[[36]](#footnote-36) که یونی گرام روابط بین مشاهدات و خواسته‌های جاری را نشان می‌دهد درحالی‌که بایگرام رابطه بین مشاهدات و ترکیب خواسته‌های جاری و قبلی را شامل می‌شود.

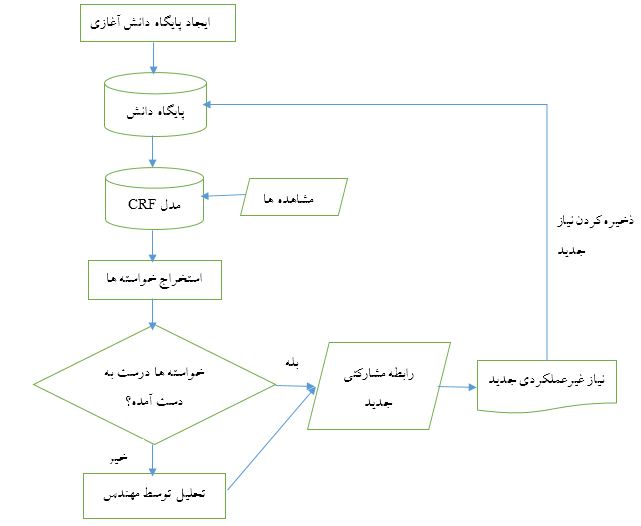
موردی که در این تحقیق بسیار مدنظر قرارگرفته وضعیت سیستم در مدت‌زمان خاصی است (منظور همان آگاهی از موقعیت است)، مثلاً فرض ما این است که کاربری در حال کار با یک سیستم نرم‌افزاری است و از سیستم درخواست مشاهده اطلاعات خود را دارد (با کلیک بر روی یک دکمه فرضی) کاربر از سیستم انتظار دارد که در مدت‌زمان کمی به درخواست او جواب بدهد حال فرض کنید سیستم نرم‌افزاری قادر به انجام درخواست نیست و این احتمال دارد به دلایل بسیاری اتفاق بیفتد مثلاً کلیک‌های مکرر کاربر امکان دارد باعث ایجاد وقفه در کار او شود و یا یک کار که در لحظه‌ای قبل انجام داده و بدون تمام کردن آن سیستم را بسته و دوباره از ابتدا شروع به کارکرده و یا حتی رویدادی مانند قطع برق و یا بار اضافی موجود بر روی حافظه سیستم و... تمام این زنجیره رویدادها، در روند کار سیستم تأثیرگذار است پس در صورت نیازسنجی بررسی همه این موارد ضروری است و باید این موارد نیز در پایگاه دانش ثبت و ذخیره شوند. مدل زمینه تصادفی مشروط که برای استخراج نیازها انتخاب‌شده با استفاده از دو نوع الگوی ذکرشده (یونی گرام و بایگرام) تمام این سلسله رویدادها را در نظر می‌گیرد.

مرحله سوم به دست آوردن خروجی مطلوب است که زمینه تصادفی مشروط برای تمام داده‌ها با استفاده از یک تابع وزن‌دار مقداری تحت عنوان اطمینان اکتسابی محاسبه می‌کند و این مقدار عددی مشخص می‌کند کدام رشته مشاهده‌ها امکان دارد یک خواسته در نظر گرفته شود. سپس با استفاده از توابع نرمال‌سازی به هرکدام از رشته ورودی‌ها مقدار احتمال مابین 0 و 1 تعلق می‌گیرد.

رشته ورودی‌های با بالاترین احتمال به‌عنوان نتیجه مناسب مدنظر قرار می‌گیرند و نتایج به‌دست‌آمده از این طریق نیازمندی‌های غیر عملکردی سیستم هستند.

در شکل 3-1 فلوچارت کلی نحوه استخراج نیازهای غیر عملکردی نشان داده‌شده است:

پس از تشخیص نیازمندی‌ها برای آن‌ها درخت نیازمندی‌ها را تشکیل می‌دهیم، در این درخت گره‌های پایینی زیرمجموعه گره‌های بالایی هستند و گره‌های پدر به گره‌های فرزندان تجزیه می‌شوند. نسبت به محل قرارگیری نیازمندی‌ها آن‌ها را به دودسته نیازهای سطح پایین (خواسته‌هایی هستند که به برگ درخت نزدیک‌تر هستند) و نیازهای سطح بالا (خواسته‌هایی هستند که به ریشه درخت نزدیک‌تر هستند.) تقسیم می‌کنیم همچنین برای هر نیازی بررسی می‌شود که این نیاز مقدار ضریب اطمینان بالایی دارد یا خیر؟



شکل (3-1): فلوچارت نحوه به دست آوردن نیازهای غیر عملکردی

با ترکیب این دو مورد یعنی سطح نیاز (سطح بالا یا پایین) و مقدار ضریب اطمینان جدول زیر به دست می‌آید.

|  |  |  |  |
| --- | --- | --- | --- |
| خواسته سطح بالا | |  | |
| ضریب اطمینان کم | ضریب اطمینان زیاد |
| ال اچ | اچ اچ | ضریب اطمینان زیاد | خواسته سطح پایین |
| ال ال | اچ ال | ضریب اطمینان کم |

جدول (3-1): روابط ایجادشده بین نیازها از طریق سیستم زمینه تصادفی مشروط

این جدول چهار حالت دارد:

حالت اول: نیازهای سطح بالا و پایین با ضریب اطمینان بالا[[37]](#footnote-37)(اچ اچ)

در این حالت نیازی که شناسایی‌شده است از اطمینان بالایی برخوردار است و تفاوتی ندارد که این نیاز سطح بالا باشد یا سطح پایین و به دلیل هم‌زمانی ضریب اطمینان بالا و سطح بالای نیازها این احتمال وجود دارد که این نیاز در پایگاه دانش قبلاً وجود داشته است.

این را می‌دانیم که هر نیاز سطح پایین در ایجاد یک نیاز سطح بالا تأثیر دارد و این تأثیر با یک خط ارتباطی نشان داده می‌شود. حتی این احتمال وجود دارد که هر نیاز سطح بالا از چند نیاز سطح پایین به‌دست‌آمده باشد که این موضوع با خطوط ارتباطی نمایش داده می‌شود.

اگر این خط در پایگاه دانش وجود نداشته باشد به این معنی است که این ارتباط جدید بوده و باید آن را به پایگاه دانش اضافه کنیم.

حالت دوم: خواسته سطح بالا با ضریب اطمینان پایین و خواسته سطح پایین با ضریب اطمینان بالا[[38]](#footnote-38) (ال اچ)

در این حالت خواسته سطح پایین از ضریب اطمینان بالایی برخوردار است و این احتمال قوی می‌شود که این خواسته احتمالاً در پایگاه دانش از قبل وجود داشته. پایین بودن مقدار خواسته سطح بالا نشان می‌دهد که نیازهای سطح پایین در مکان درست قرار نگرفته‌اند و باید والدهای گره‌های همسایگی آن‌ها بررسی شوند و یا اینکه این نیازها یک خواسته پنهانی را تعریف می‌کنند که در نظر گرفته نشده است.

حالت سوم: خواسته سطح بالا با ضریب اطمینان بالا و خواسته سطح پایین با ضریب اطمینان پایین[[39]](#footnote-39)(اچ ال)

در این شرایط خواسته سطح بالا ضریب اطمینان بالایی دارد اما فرزندان آن‌که نیازهای سطح پایین هستند ضریب اطمینان کمی دارند، از کم بودن این مقدار نتیجه می‌گیریم که این نیازها در پایگاه دانش ندارند و این احتمال وجود دارد که زمینه تصادفی مشروط آن‌ها را به‌درستی محاسبه نکرده باشد، در این حالت باید به دنبال خواسته‌های سطح پایین مناسب‌تر باشیم که جایگزین قبلی‌ها شوند.

حالت چهارم: خواسته سطح بالا و پایین هردو مقدار ضریب اطمینان پایین دارند[[40]](#footnote-40)(ال ال)

در مورد این حالت که سخت‌ترین حالت به وجود آمده است باید به دنبال دلایل مختلفی باشیم که باعث به وجود آمدن این شرایط شده‌اند که این دلایل می‌تواند بسیار زیاد و متعدد باشد، پس‌ازآن باید تلاش کنیم نیازهای جایگزین مناسب را به دست آوریم.

برخی از روش‌های آماری که برای تحلیل نتایج در این تحقیق استفاده‌شده است به‌صورت زیر است:

1. آزمون مان – ویتنی[[41]](#footnote-41): جزء [آزمون‌های غیر پارامتری](https://fa.wikipedia.org/w/index.php?title=%D8%A2%D8%B2%D9%85%D9%88%D9%86_%D8%BA%DB%8C%D8%B1_%D9%BE%D8%A7%D8%B1%D8%A7%D9%85%D8%AA%D8%B1%DB%8C&action=edit&redlink=1) است و برای سنجش تفاوت میان نمونه‌ها به کار می‌رود. در این آزمون رتبه‌بندی روی می‌دهد و محاسبات بر روی رتبه‌ها انجام می‌گیرد.

2. آزمون آماری مقدار پی[[42]](#footnote-42): این آزمون به این دلیل استفاده می‌شود که ثابت کنیم آیا تئوری تهی رد می‌شود یا نه؟ مقدار پی حاصل از این آزمون یک مقدار بین صفر و یک است و از انجام محاسبات بر روی دو گروه ورودی به دست می‌آید و میزان تصادفی بودن تفاوت بین دو گروه را نشان می‌دهد به این صورت که مقدار پی کوچک (<= 0.05) تئوری تهی را به‌طور کامل رد می‌کند. مقدار پی بزرگ (>= 0.05) نمی‌تواند تئوری تهی را به‌طور کامل رد کند.

2-3 جامعه و نمونه آماری

برای انجام تحقیق و به دست آوردن داده‌ها در این سمینار از سیستم محیط تحقیقاتی تعاونی**[[43]](#footnote-43)** استفاده‌شده است. محیط تحقیقاتی تعاونی یک سیستم نرم‌افزاری متن‌باز است که مطابق با نیاز این تحقیق توسعه داده‌شده و بستری برای اشتراک ایده‌ها و عقاید محققان درباره مقالات دانشگاهی می‌باشد.

قابلیت‌های این سیستم عبارت‌اند از عضویت در این سامانه و استفاده از ویژگی‌های دانلود، آپلود و مشاهده مقالات، بیان نظرات و ثبت آن‌ها و ...

شرکت‌کننده‌ها در این تحقیق از گروه‌های متفاوت علوم کامپیوتر و از سه دانشگاه متفاوت شامل دانشگاه ایالتی آیووا در ایالات‌متحده، دانشگاه نورث ایسترن چین و دانشگاه نیهون ژاپن انتخاب‌شده‌اند. درمجموع 147 نفر در این تحقیق شرکت کرده‌اند که %78 آن‌ها با علوم کامپیوتر آشنایی داشته‌اند میانگین آشنایی شرکت‌کننده‌ها با سیستم محیط تحقیقاتی تعاونی حدود 3.3 است که نشان‌دهنده آشنایی جزئی آن‌ها است. حدود %85 شرکت‌کننده‌ها حداقل مدرک تحصیلی لیسانس داشته‌اند، سه زبان چینی، ژاپنی و انگلیسی به ترتیب بیشتر از سایر زبان‌ها استفاده‌شده است.

مدت‌زمان انجام این تحقیق 6 ماه بوده، از ماه مارچ تا سپتامبر سال 2018، این داده‌ها زمانی جمع‌آوری‌شده‌اند که مدل زمینه تصادفی مشروط طراحی نشده بود بنابراین از یک متخصص نیز در این زمینه کمک گرفته‌شده و فرضیات مطرح‌شده توسط متخصص برای موارد آموزشی ذخیره شد.

درمجموع تعداد 11960 سطر داده از 949 مورد کار با سیستم جمع‌آوری‌شده است، منظور از کار با سیستم تمام فعالیت‌هایی است که از زمان ورود به سیستم تا زمان خروج از آن انجام‌شده است. البته باید این را در نظر داشت که در بسیاری از این موارد کاربران همکاری کامل را با ما نداشته و با دقت با سیستم کارنکرده‌اند که این داده‌ها باید از دامنه ورودی تحقیق حذف شوند. پس از بررسی داده‌ها و به کار بردن الگوریتم مناسب برای یافتن داده‌های درست و با دقت بالا تعداد داده‌های نهایی به 4388 سطر کاهش پیدا کرد که این تعداد از 792 مورد کار با سیستم به‌دست‌آمده بود.

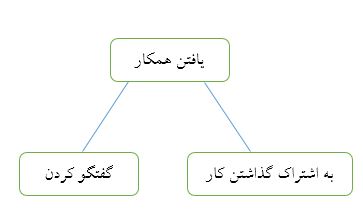
3-3 ابزار اندازه‌گیری

در این قسمت با بررسی یک مثال از روش حل مسئله ابزار اندازه‌گیری را کامل توضیح می‌دهیم:

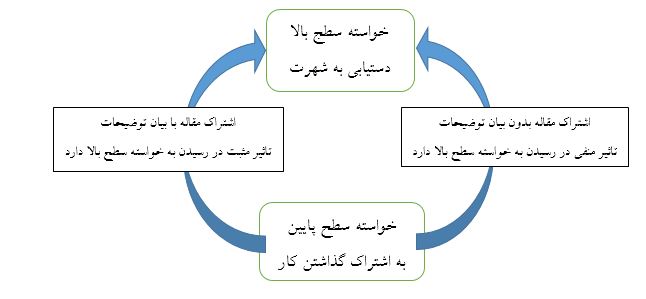
برای جمع‌آوری اطلاعات این تحقیق از سیستم تحت وبی به‌عنوان محیط تحقیقاتی تعاونی استفاده شده است محیطی که به‌منظور اشتراک مقالات علمی و بحث و تبادل‌نظر در این زمینه طراحی‌شده، در این محیط کاربر می‌تواند ثبت‌نام کند، اطلاعات خود را ثبت کند، مقاله یا مقالاتی را در این سامانه بارگزاری کند، راجب مطالب مختلف بحث و تبادل‌نظر کند و ... این امکان تعبیه‌شده که اگر نیازهای کاربر توسط سیستم رفع نشده بود بتواند درخواست اضافه کردن آن به لیست قابلیت‌های نرم‌افزار را داشته باشد.

در این مرحله ما دو نیاز کاربر را در نظر می‌گیریم؛ نیاز یافتن همکار و نیاز دیگر دستیابی به شهرت، ازآنجایی‌که این دو نیاز هردو باید با انجام دادن کارهای دیگری برآورده شوند پس نیازهای سطح بالا هستند. برطرف کردن نیاز یافتن همکار با به اشتراک گذاشتن مقاله و بحث و گفتگو با سایرین امکان‌پذیرتر است همچنین نیاز دستیابی به شهرت بدون به اشتراک گذاشتن کار یا مقاله امکان‌پذیر نیست.

درصورتی‌که کاربر مقالات را با جزئیات آن به اشتراک بگذارد قطعاً تأثیر مثبتی در دستیابی به شهرت آن کاربر دارد.



شکل (3-2): رابطه سلسله مراتبی بین نیاز یافتن همکار و نیازهای سطح پایین آن



شکل (3-3): رابطه بین خواسته رسیدن به شهرت و خواسته به اشتراک گذاشتن کار

با توجه به شکل 3-1 که روند انجام کار را نشان می‌دهد در ابتدای تحقیق باید یک پایگاه دانش ایجاد کنیم، پایگاه دانش ایجادشده برای آموزش مدل زمینه تصادفی مشروط یک جدول با سه ستون است ستون اول زمان رخ دادن رویداد، ستون دوم رشته کارهایی که برای رسیدن به خواسته سطح بالا «جستجوی همکار[[44]](#footnote-44)» انجام‌شده و ستون سوم رشته کارهایی که برای رسیدن به خواسته سطح بالای «رسیدن به شهرت[[45]](#footnote-45)» انجام‌شده است.

|  |  |  |
| --- | --- | --- |
| رویدادهای خواسته رسیدن به شهرت | رویدادهای خواسته یافتن همکار | زمان |
| کلیک منوی آپلود مقالات | کلیک منوی تمام مقالات (موفق 2 ثانیه) درخواست‌های (جستجوی همکار، گفتگو) | 1 |
| کلیک نظرات (موفق 3 ثانیه) درخواست‌های (اشتراک کار، جستجوی همکار) | کلیک فیلتر (موفق 6 ثانیه) نیازهای (گفتگو، جستجوی همکار) | 2 |
| کلیک ارسال (موفق 331 ثانیه) نیازهای اشتراک کار، مشهور شدن | کلیک نظرات (موفق 3 ثانیه) نیازهای (گفتگو، جستجوی همکار) | 3 |
| کلیک آپلود مقاله (موفق 4 ثانیه) نیازهای (مشهور شدن؛ اشتراک کار) | کلیک ارسال (موفق 87 ثانیه) نیازهای (گفتگو، جستجوی همکار) | 4 |

جدول (3-2): نمونه داده‌های آموزشی برای نرم‌افزار زمینه تصادفی مشروط

هر سطر این جدول شامل یک مجموعه کار است مثلاً در نقطه زمانی 4 ستون دوم نشان می‌دهد که دکمه ارسال[[46]](#footnote-46) زده‌شده و سیستم به این درخواست به‌درستی عمل کرده و این عمل 87 ثانیه طول کشیده و در مدت‌زمان کار کاربر با سیستم دو درخواست از سیستم داشته، خواسته سطح بالای جستجوی همکار و خواسته سطح پایین بحث و گفتگو[[47]](#footnote-47).

برای دادن ورودی به سیستم زمینه تصادفی مشروط برای هر نیاز سطح بالا و نیاز سطح پایین الگوهای ویژگی را با استفاده از یونی گرام و بایگرام ایجاد می‌کنیم.

این الگوهای ویژگی درواقع به‌عنوان داده آموزشی سیستم زمینه تصادفی مشروط وارد این سیستم می‌شوند تا سیستم بتواند عملکرد درست بر روی ورودی‌ها را از این طریق بیاموزد.

داده‌هایی که شامل عمل، نقطه زمانی و خواسته‌ها هستند ورودی سیستم زمینه تصادفی مشروط بوده و خروجی این سیستم مشابه جدول 3-1 روابط ایجادشده بین خواسته‌ها را نشان می‌دهد.

|  |  |
| --- | --- |
| بایگرام | یونی گرام |
| بایگرام 01: %X[0,0] | یونی گرام 01:%X[0,0] |
| بایگرام 02: %X[0,0]/%X[1,0] | یونی گرام 02:%X[-1,0]/ %X[0,0] |
| بایگرام 03: %X[-1,0]/%X[0,0] | یونی گرام 03%X[0,0]/%X[1,0] : |

جدول (3-3): نمونه الگوهای محاسبه‌شده برای خواسته‌های سطح پایین

حال باید این را در نظر داشته باشیم که نیازهای غیر عملکردی از ارتباط موجود بین خواسته‌های سطح پایین و بالا و بسته به ضریب اطمینانی که توسط سیستم تعیین‌شده به دست می‌آیند. پس از تشخیص نیازهای جدید باید برای هرکدام از آن‌ها مقادیر توابع آماری آزمون مان-ویتنی و آزمون پی محاسبه

می‌شود.

|  |  |  |  |
| --- | --- | --- | --- |
| تعداد نمونه | شاخص وزن | خواسته سطح پایین | خواسته سطح بالا |
| 311  184  39 | 0.582  0.345  0.073 | به اشتراک گذاشتن کار  بازنگری مقاله  بحث و گفتگو | به شهرت رسیدن |
| 266  187 | 0.587  0.413 | بحث و گفتگو  به اشتراک گذاشتن کار | جستجو برای همکار |

جدول (3-4): وزن به‌دست‌آمده برای خواسته‌های سطح بالا و سطح پایین

از طریق روابط بین خواسته‌ها به این نتیجه می‌رسیم که خواسته رسیدن به شهرت از دو طریق امکان‌پذیر است: 1- به اشتراک گذاشتن کار (با وزن 0.582) 2- دو رویداد تجدیدنظر کردن در کار به اشتراک گذاشته‌شده و بحث و گفتگو که وزن هردوی آن‌ها به‌عنوان رشته عملکرد محاسبه می‌شود (وزن = 0.345 و 0.073) اگرچه وزن حالت اول بیشتر است اما با بررسی عملکرد کاربران و نظر مهندس نرم‌افزار حالت دوم برای پیاده‌سازی انتخاب می‌شود.

در مورد نیاز جستجوی همکار از بررسی روابط به‌دست‌آمده توسط مدل زمینه تصادفی مشروط به رابطه جدیدی نرسیدیم اما وزن‌های به‌دست‌آمده نشان داد که کاربران بیشتر تمایل دارند از طریق بحث و گفتگو به نتیجه برسند بنابراین این مورد می‌تواند کمک کند که طراح سامانه در صفحه بارگزاری مقاله لینکی به صفحه قرار دادن نظرات ایجاد کند. در صفحه نظرات است که امکان گفتگوی کاربران فراهم‌شده.

مدل زمینه تصادفی مشروط برای تمام خواسته‌ها که به‌صورت ورودی داده می‌شود موارد گفته‌شده را محاسبه می‌کند و به این صورت می‌توانیم نیازهای غیر عملکردی را از طریق روابط به‌دست‌آمده استخراج‌کنیم.

فصل چهارم

تجزیه‌وتحلیل

1-4 تجزیه‌وتحلیل

این سمینار روشی را برای تجزیه‌وتحلیل و استخراج نیازهای غیر عملکردی از روابط موجود میان خواسته‌های انسان بر اساس شرایط و موقعیت موجود پیشنهاد می‌کند. خواسته‌ها را با سطوح انتزاعی متفاوت تفکیک می‌کنیم و با بررسی و انجام محاسبات آماری لازم روابط موجود بین آن‌ها را به دست می‌آوریم.

درواقع با تمرکز بر عوامل انسان‌محور در اهداف موجود تجزیه‌وتحلیل نیازهای غیر عملکردی از دیدگاه کاربران صورت می‎گیرد. این چارچوب می‎تواند نه‌تنها الزامات غیر عملکردی، بلکه الزامات عملکردی را نیز از طریق بررسی متخصصان حوزه استخراج کند. در مطالعات موردی ما، بسیاری از نیازهای غیر عملکردی تمرکز خود را بر قابلیت استفاده از سیستم جلب کردند. بااین‌حال، به‌منظور بهبود قابلیت استفاده، مقدار معینی از تغییر یا عملکرد جدید باید ایجاد یا اضافه شود و تمام این موارد را از طریق روابط موجود میان خواسته‌ها می‌توانیم به دست آوریم.

در توضیح متن بالا باید این را در نظر داشت که نیاز غیر عملکردی همیشه امنیت و سرعت کارکرد نرم‌افزار ما نیست، گاهی این نیاز را می‌توانیم یک تغییر کوچک در نظر بگیریم. تغییری مانند افزودن یک دکمه یا ایجاد ارتباط بین دو قسمت از سیستم نرم‌افزاری.

پس اینکه با استفاده از روش زمینه تصادفی مشروط حتماً باید به نیاز مشخصی برسیم برداشت نادرستی از این مفهوم است، درواقع نتیجه کار ما گاه می‌تواند بسیار جزئی باشد و یا نتیجه یک تغییر نگرش و دید نسبت به سیستم موجود باشد.

کارکرد چارچوب معرفی‌شده در این سمینار بیشتر درزمانی است که سیستم نرم‌افزاری پیاده‌سازی شده و هم‌زمان با استفاده کاربران از آن، نیازها و قابلیت‌های موجود توسعه داده می‌شود. حال این سوال مطرح می‌شود که اگر بخواهیم قبل از توسعه نرم‌افزار به این اطمینان برسیم که نیازسنجی کامل صورت گرفته چه راه‌حلی را باید به کار ببریم؟

فهرست منابع

[1] Peng Sun A multi-layered desires based framework to detect evolving nonfunctional

requirements of users, Iowa State University Capstones, Theses and Dissertations,2020

[2] S. Robertson and J. Robertson. Mastering the requirements process: Getting

requirements right. Addison-wesley, 2012

[3] L. Zhu and I. Gorton. Uml pro\_les for design decisions and non-functional

requirements. In Proceedings of the Second Workshop on Sharing and Reusing

Architectural Knowledge Architecture, Rationale, and Design intent, page 8. IEEE

Computer Society, 2007.

[4] C. Jensen and C. Potts. Experimental evaluation of a lightweight method for

augmenting requirements analysis. In Proceedings of the 1st ACM international

workshop on Empirical assessment of software engineering languages and technologies:

held in conjunction with the 22nd IEEE/ACM International Conference on Automated

Software Engineering (ASE) 2007, sspages 49{54. ACM, 2007.

[5] C. Tibermacine, R. Fleurquin, and S. Sadou. Nfrs-aware architectural

evolution of component-based software. In Proceedings of the 20th IEEE/ACM international

conference on Automated software engineering, pages 388{391. ACM, 2005.

[6] Y. Gazi, M. S. Umar, and M. Sadiq. Classi\_cation of nfrs for information

System International Journal of Computer Applications, 115(22), 2015.

[7] J. Eckhardt, A. Vogelsang, and D. M. Fern\_andez. Are" non-functional"

requirements really non-functional an investigation of non-functional

requirements in practice. In Software Engineering (ICSE), 2016 IEEE/ACM 38th

International Conference on, pages 832{842. IEEE, 2016

[8] J. Slankas and L. Williams. Automated extraction of non-functional

requirements inavailable documentation. In Natural Language Analysis in Software

Engineering

(NaturaLiSE), 2013 1st International Workshop on, pages 9{16. IEEE, 2013.

[9] A. van Lamsweerde. Goal-oriented requirements enginering: a roundtrip from

research to practice [enginering read engineering]. In Requirements Engineering

Conference, 2004. Proceedings. 12th IEEE International, pages 4{7. IEEE, 2004.

[10] J. Zou, L. Xu, W. Guo, M. Yan, D. Yang, and X. Zhang. Which non-functional

requirements do developers focus on? an empirical study on stack overow using

topic analysis. In Mining Software Repositories (MSR), 2015 IEEE/ACM 12th Working

Conference on, pages 446{449. IEEE, 2015.

[11] C. K. Chang, H.y. Jiang, H. Ming, and K. Oyama. Situ: A situation-theoretic

approach to context-aware service evolution. IEEE Transactions on Services

Computing, 2(3):261{275, 2009.

[12] I. Sommerville and P. Sawyer. Requirements engineering: a good practice guide.

John Wiley & Sons, Inc. 1997.

[13] S. Liaskos, S. McIlraith, and J. Mylopoulos. Representing and reasoning with

preference requirements using goals. Technical report, Technical report, Dept. of

Computer Science, University of Toronto, 2006.

[14] T. Keller. Contextual requirements elicitation: An overview. In Seminar in

Requirements Engineering, Department of Informatics, University of Zurich, 2011.

[15] J. Cleland-Huang, R. Settimi, X. Zou, and P. Solc. Automated classi\_cation

Of non-functional requirements. Requirements Engineering, 12(2):103{120, 2007

[16] S. S. Yau, H. Gong, D. Huang, W. Gao, and L. Zhu. Speci\_cation,

decomposition and agent synthesis for situation-aware service-based systems.

Journal of Systems and Software, 81(10):1663{1680, 2008.

[17] M. R. Endsley. Toward a theory of situation awareness in dynamic systems.

In Situational awareness, pages 9{42. Routledge, 2017.

[18] D. Lee and S. Helal. From activity recognition to situation recognition. In

international Conference on Smart Homes and Health Telematics, pages 245{251.

Springer, 2013.

[19] A. McCallum, D. Freitag, and F. C. Pereira. Maximum entropy markov

models for information extraction and segmentation. In Icml, volume 17, pages

591{598, 2000.

[20] J. La\_erty, A. McCallum, and F. C. Pereira. Conditional random \_elds:

Probabilistic models for segmenting and labeling sequence data. 2001.

[21] Y. Liu, J. Carbonell, P. Weigele, and V. Gopalakrishnan. Protein fold

recognition usingsegmentation conditional random \_elds (scrfs). Journal of

Computational Biology,

13(2):394{406, 2006.

[22] K. K. Gupta, B. Nath, and R. Kotagiri. Layered approach using conditional

random \_elds for intrusion detection. IEEE Transactions on dependable and secure

Computing, 7(1):35, 2010.

1. 1 Non Functional Requirement [↑](#footnote-ref-1)
2. situ [↑](#footnote-ref-2)
3. Conditional Random Field [↑](#footnote-ref-3)
4. Gazi et al [↑](#footnote-ref-4)
5. Afreen et al [↑](#footnote-ref-5)
6. Eckhardt et al [↑](#footnote-ref-6)
7. Natural Language Processing [↑](#footnote-ref-7)
8. Goal Oriented Requirements Engineering [↑](#footnote-ref-8)
9. Zou et al [↑](#footnote-ref-9)
10. Stack overflow [↑](#footnote-ref-10)
11. Requirements Engineering [↑](#footnote-ref-11)
12. Non Functional Requirement [↑](#footnote-ref-12)
13. Slankas et al [↑](#footnote-ref-13)
14. K-N N [↑](#footnote-ref-14)
15. Naive Bayes [↑](#footnote-ref-15)
16. Sequential Minimum Optimizer [↑](#footnote-ref-16)
17. Yau et al [↑](#footnote-ref-17)
18. Mica Endsley [↑](#footnote-ref-18)
19. Duckki Lee and Sumi Helal [↑](#footnote-ref-19)
20. t [↑](#footnote-ref-20)
21. Chang et al [↑](#footnote-ref-21)
22. (D, A, E)t [↑](#footnote-ref-22)
23. Mental Context [↑](#footnote-ref-23)
24. Behavioral Context [↑](#footnote-ref-24)
25. Environmental Context [↑](#footnote-ref-25)
26. Situation-Awareness [↑](#footnote-ref-26)
27. Part of speech [↑](#footnote-ref-27)
28. Named entity recognition [↑](#footnote-ref-28)
29. handwriting recognition [↑](#footnote-ref-29)
30. Hidden Markov Models [↑](#footnote-ref-30)
31. Maximum Entropy Markov Models [↑](#footnote-ref-31)
32. CRF [↑](#footnote-ref-32)
33. Wapiti [↑](#footnote-ref-33)
34. MALLEt [↑](#footnote-ref-34)
35. Unigram [↑](#footnote-ref-35)
36. Bigram [↑](#footnote-ref-36)
37. High confidence high and low level desires(hh) [↑](#footnote-ref-37)
38. ) Low confidence high level desires and high confidence low level desires (lh) [↑](#footnote-ref-38)
39. High confidence high level desires and low confidence low level desires(hl) [↑](#footnote-ref-39)
40. Low confidence high level desires and low confidence low lwvwl desires(ll) [↑](#footnote-ref-40)
41. Mann-Whitney U test [↑](#footnote-ref-41)
42. P values [↑](#footnote-ref-42)
43. Cooperative Research Environment [↑](#footnote-ref-43)
44. Look for collaborator [↑](#footnote-ref-44)
45. Build reputation [↑](#footnote-ref-45)
46. Submit [↑](#footnote-ref-46)
47. Discuss [↑](#footnote-ref-47)