





**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 ۱۴۰۰**

## چکیده

نیازهای غیر عملکردی<sup>۱</sup> تأثیر بسیار زیادی بر نحوه کارکرد یک نرم افزار دارند، درواقع می توانیم این نیازها را نیازهای کیفی در نظر بگیریم و با اطمینان بگوییم سیستم نرم افزاری که در زمان توسعه و پیدایش به این نیازها بی توجه بوده کارا نیست و با شکست مواجه خواهد شد. کارهایی که اخیراً در زمینه مهندسی نرم افزار و در مرحله نخست تولید نرم افزار یعنی تجزیه و تحلیل نیازمندی ها انجام شده نشان داده اند که ما می توانیم با ترکیب کردن اهداف مدنظر کاربران و نحوه تعامل آن ها با نرم افزار تا حدود زیادی به نیازهای کیفی لازم برسیم. دستیابی به این نیازها به سادگی صورت نمی گیرد و برای این کار باید به تمام خواسته های کاربران شامل خواسته های صریح و خواسته های ضمنی دست پیدا کرده هم چنین روابط متقابل بین کاربران، نیازهای به دست آمده، عوامل سیستم و سناریوها به طور دقیق بررسی شوند. در این سمینار که براساس مرجع [۱] نوشته شده است نحوه به دست آوردن نیازهای غیر عملکردی با استفاده از اهداف و خواسته های کاربران و روابط بین آن ها بررسی شده است؛ در این تحقیق از یک مدل آماری تحت چارچوب سیتو<sup>۲</sup> به نام زمینه تصادفی مشروط<sup>۳</sup> استفاده شده که توسط این مدل خواسته های به دست آمده کاربران با توجه به میزان اهمیت آن ها در سطوح مختلف دسته بندی می شوند، این خواسته ها و روابط میان آن ها نیازهای غیر عملکردی را نتیجه می دهند.

کلمات کلیدی: نیازهای غیر عملکردی، اهداف، روابط بین خواسته ها، مهندسی نرم افزار

---

<sup>1</sup> Non Functional Requirement

<sup>۲</sup> situ

<sup>۳</sup> Conditional Random Field

## فهرست مطالب

عنوان	صفحه
فصل اول: کلیات طرح تحقیق .....	۱
۱-۱ مقدمه .....	۲
۲-۱ سابقه و ضرورت انجام تحقیق .....	۳
فصل دوم: ادبیات و پیشینه تحقیق .....	۵
۱-۲ مهندسی الزامات .....	۶
۲-۲ استخراج الزامات غیر عملکردی .....	۶
۳-۲ چارچوب سیتو .....	۷
۴-۲ آگاهی از موقعیت (موقعیت آگاهی) .....	۹
۵-۲ مدل‌های یادگیری تحت نظارت متوالی .....	۱۰
فصل سوم: متدولوژی .....	۱۱
۱-۳ روش تحقیق .....	۱۲
۲-۳ جامعه و نمونه آماری .....	۱۶
۳-۳ ابزار اندازه‌گیری .....	۱۷
فصل چهارم: تجزیه و تحلیل .....	۲۲
۱-۴ تجزیه و تحلیل .....	Error! Bookmark not defined.
فهرست منابع .....	۲۴

## فهرست اشکال

عنوان	صفحه
شکل (۱-۳): فلوچارت نحوه به دست آوردن نیازهای غیر عملکردی .....	۱۴
شکل (۲-۳): رابطه سلسله مراتبی بین نیاز یافتن همکار و نیازهای سطح پایین آن .....	۱۸
شکل (۳-۳): رابطه بین خواسته رسیدن به شهرت و خواسته به اشتراک گذاشتن کار .....	۱۸

## فهرست جداول

عنوان	صفحه
جدول (۱-۳): روابط ایجاد شده بین نیازها از طریق سیستم زمینه تصادفی مشروط	۱۴
جدول (۲-۳): نمونه داده های آموزشی برای نرم افزار زمینه تصادفی مشروط	۱۹
جدول (۳-۳): نمونه الگوهای محاسبه شده برای خواسته های سطح پایین	۲۰
جدول (۴-۳): وزن بدست آمده برای خواسته های سطح بالا و سطح پایین	۲۰

# **فصل اول**

## **کلیات طرح تحقیق**

## ۱-۱ مقدمه

در مراحل طراحی و تولید نرم‌افزارها یکی از اصلی‌ترین و مهم‌ترین مرحله کار تشخیص نیازمندی‌ها است، تقریباً ۵۰ تا ۶۰ درصد از خطاهای موجود در توسعه نرم‌افزار ناشی از تشخیص نادرست است [۲] و این باعث می‌شود هزینه اصلاح نرم‌افزارها تا ۱۰۰ برابر افزایش پیدا کند [۲]

نیازمندی‌ها در واقع 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 طراحی شده این انتظار را دارد که در زمان اتصال به پایگاه داده تعداد رکوردهای موجود در پایگاه را به وی نشان دهد و این یک 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 به صورت 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 تجزیه و تحلیل نیازها به دست آوریم.

این سمینار بر پایه این نگرش استوار است که نیازمندی‌های مدنظر یک سیستم نرم‌افزاری خواسته‌های کاربران هستند که به شدت با اهدافی که برای پیاده‌سازی سیستم بیان شده 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 به تمامی خواسته‌ها و نیازهای کاربران از سیستم نرم‌افزاری است. درک و تجزیه و تحلیل درست این نیازها می‌تواند به تصمیمات طراحی بهتر [۳]، تغییرات مورد نیاز [۴] و تکامل معماری [۵] کمک کند.



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

## ۱-۲ سابقه و ضرورت انجام تحقیق

در یکی از طبقه‌بندی‌هایی که برای نیازهای غیر عملکردی بیان شده 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<sup>۱</sup> انجام شده است [۶]. در طبقه‌بندی دیگری توسط افرین و همکاران<sup>۲</sup>، قابلیت استفاده، عملکرد، قابلیت اطمینان، امنیت و قابلیت نگهداری را به عنوان نیازهای غیر عملکردی متداول نام برده‌اند [۲].

طبق مطالعه اخیر اکهارت و همکاران<sup>۳</sup>، ۵۳۰ دسته‌بندی برای نیازهای غیر عملکردی از ۱۱ مورد نیاز سنجی صنعتی استخراج شده است که ۷۵ درصد آن نیازهای عملکردی هستند یعنی حتماً باید مورد توجه قرار بگیرند و در هنگام توسعه نرم‌افزار کامپیوتری این نیازها برطرف شوند. [۷]

استخراج الزامات غیر عملکردی می‌تواند کاری سخت و زمان‌بر باشد. برخی از محققان با استفاده از تکنیک‌های پردازش زبان طبیعی<sup>۴</sup> بر اساس اسناد تولیدشده برای راهنمایی کاربر و اطلاعات به دست آمده در زمان تجزیه و تحلیل نیازمندی‌ها و داده‌هایی که از سیستم جمع‌آوری شده است به این پی برده‌اند که استخراج نیازهای غیر عملکردی کار بسیار سختی است. [۸] رویکردهای مبتنی بر پردازش زبان طبیعی اغلب فرض می‌کنند که مستندات مورد نیاز به وضوح بیان شده است درحالی‌که بسیاری از محصولات نرم‌افزاری که در دنیای واقعی استفاده می‌شوند به دلیل تکنیک‌های نیازسنجی ناکافی فاقد مستندات مورد نیاز 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.

---

<sup>۱</sup>Gazi et al

<sup>۲</sup>Afreen et al

<sup>۳</sup>Eckhardt et al

<sup>۴</sup>Natural Language Processing

یکی دیگر از روش‌های نیازسنجی، مهندسی الزامات هدف‌گرا<sup>۱</sup> است که اهداف را مورد توجه و بررسی قرار می‌دهد و نیازمندی‌های سیستم را با استفاده از اهداف تعیین شده برای آن در سطوح مختلف انتزاعی بیان می‌کند. [۹]

در یک فرایند هدف‌گرا، فعالیت‌های مهندسی الزامات با در نظر گرفتن روابط بین اهداف، عوامل و سناریوها بیان می‌شوند. [۹]

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

یکی دیگر از کارهایی که در این زمینه صورت گرفت و این نیازها را از دید توسعه‌دهندگان مورد مطالعه قرارداد، توسط ژو و همکاران<sup>۲</sup> [۱۰] انجام شد. آن‌ها پست‌ها و مباحث مربوط به یک وب‌سایت پرسش و پاسخ، به نام سرریز پشته<sup>۳</sup> را به عنوان ورودی متن در نظر گرفتند و از مدل موضوعی برای استخراج نیاز غیر عملکردی از روی این ورودی‌ها استفاده کردند. با این حال، محدودیت روش آن‌ها این بود که نیازهای کاربران نهایی را نمی‌توان به راحتی با نظرات مهندسان سیستم برطرف کرد و همچنین آماده‌سازی پست‌های مناسب که بتوان از روی آن‌ها و نظرات بیان شده برای آن‌ها تجزیه و تحلیل را انجام داد، به مدت زمان طولانی غیرقابل پیش‌بینی نیاز داشت. علاوه بر این، معنای واقعی از متن بحث را نمی‌توان با تکنیک مبتنی بر مدل موضوع نشان داد.

مطالعات انجام شده تاکنون به یک چارچوب توسعه نرم‌افزاری به نام سیتو منجر شده است که نیازها را از دید کاربرها بررسی می‌کند. [۱۱]

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

---

<sup>۱</sup>Goal Oriented Requirements Engineering

<sup>۲</sup>Zou et al

<sup>۳</sup>Stack overflow

## 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 در زمینه‌ی این موارد انجام شده را نیز بیان می‌کنیم:

## ۲-۱ مهندسی الزامات<sup>۱</sup>

مهندسی الزامات را می‌توان فرایندی برای کشف، مستندسازی و مدیریت الزامات سیستم‌های کامپیوتری دانست. [۱۲] هدف مهندسی الزامات ایجاد مجموعه‌ای از نیازمندی‌های سازگار، کامل و مرتبط است و آنچه را که مشتری واقعاً به آن احتیاج دارد باز می‌گرداند. [۱۲] فرایند سنتی استخراج الزامات یک فرایند یادگیری تعاملی بین مهندسان و کارفرمایان است [۱۳] که از طریق مصاحبه، مشاهده فعالیت‌های مشتریان و بازخورد عملکرد سیستم، می‌توان اطلاعات مربوط به نیازهای کاربر را به دست آورد [۱۴] 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 [۱۱]. 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.

## ۲-۲ استخراج الزامات غیر عملکردی<sup>۲</sup>

بسیاری از روش‌های اصلی برای استخراج الزامات یا نیازهای غیر عملکردی بر اساس تکنیک‌های مبتنی بر استخراج متن و پردازش زبان طبیعی است. [۹] تمامی آثار قبلی که در زمینه استخراج نیازمندی‌های سیستم نرم‌افزاری کار کرده‌اند پیشنهاد کرده‌اند که در اسناد مختلف تهیه شده 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 به عنوان شاخص انواع مختلف نیازها را شناسایی کرد این کلمات کلیدی برای طبقه‌بندی نیازهای غیر عملکردی

---

<sup>۱</sup>Requirements Engineering

<sup>۲</sup>Non Functional Requirement

استفاده می‌شود و می‌بایست از اسناد جمع‌آوری شده در مرحله تجزیه و تحلیل نیازمندی‌ها به دست بیایند.

سلانکاس و همکاران<sup>۱</sup> [۸] یک روش خودکار برای استخراج نیازهای غیر عملکردی از انواع اسناد نرم‌افزاری از جمله مشخصات مورد نیاز، توافقنامه‌های داده، راهنمای کاربر و غیره پیشنهاد کردند. جملات موجود در آن اسناد با بردارهای کلمه نشان داده شده و در ۱۴ دسته طبقه‌بندی شدند. آن‌ها از مدل‌های یادگیری ماشین مانند کی ان ان<sup>۲</sup>، نیو بایس<sup>۳</sup> و حداقل بهینه‌ساز متوالی<sup>۴</sup> برای طبقه‌بندی هر جمله جداگانه استفاده کردند و نتایج نشان داد که روش حداقل بهینه‌ساز متوالی در طبقه‌بندی جملات بر دیگر مدل‌ها برتری دارد.

## ۲-۳ چارچوب سیتو

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

در ادبیات رایانه، تعاریف مختلفی از موقعیت وجود دارد:

۱. یک موقعیت می‌تواند یک موقعیت اتمی، یک موقعیت ترکیبی منطقی یا یک موقعیت گذرا

باشد. یائو و همکاران<sup>۵</sup> [۱۶]

۲. درک عناصر موجود در محیط در یک زمان و مکان، درک معنای آن‌ها و پیش‌بینی وضعیت

آن‌ها در آینده است. میکا اندسلی<sup>۶</sup> [۱۷]

۳. یک موقعیت در زمان t سه‌گانه فعالیت‌های یک کاربر است، مجموعه‌ای از اقدامات و زمینه‌های

دستگاه‌ها در یک دوره زمانی است داکلی و سومی هلال<sup>۷</sup> [۱۸]

۴. وضعیت در زمان تی<sup>۸</sup>، یک سه‌گانه {دی، آ و ای} است. چانگ و همکاران<sup>۹</sup> [۱۱]

---

<sup>۱</sup>Slankas et al

<sup>۲</sup>K-N N

<sup>۳</sup>Naive Bayes

<sup>۴</sup>Sequential Minimum Optimizer

<sup>۵</sup>Yau et al

<sup>۶</sup>Mica Endsley

<sup>۷</sup>Duckki Lee and Sumi Helal

<sup>۸</sup>t

<sup>۹</sup>Chang et al

چارچوب سیتو از روش چانگ و همکاران استفاده می‌کند و وضعیت را به صورت سه گانه (دی، آ و ای)<sup>۱</sup> در یک لحظه به خصوص در نظر می‌گیرد.

زمینه ذهنی<sup>۲</sup> (دی) به حالات روانی انسان اشاره دارد که تاحد زیادی پنهان است، زمینه رفتاری<sup>۳</sup> (آ) چگونگی رفتار را مشخص می‌کند و زمینه محیطی<sup>۴</sup> (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. [۱۱] 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 به صورت 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. [۱۲] اما نیازمندی‌های موجود در برگ‌های این درخت اتمی هستند و نمی‌توان آن‌ها را تجزیه کرد. [۱۱] 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 تجزیه و تحلیل قرار می‌گیرد، درحالی‌که یک درخت خواسته از جنبه کاربر برای نظارت بر اهداف آن‌ها تجزیه و تحلیل می‌شود [۱۱]. چارچوب چندلایه‌ای که در این سمینار ارائه شده است مطابق با ساختار درخت خواسته‌ها

---

<sup>۱</sup>(D, A, E)<sub>t</sub>

<sup>۲</sup>Mental Context

<sup>۳</sup>Behavioral Context

<sup>۴</sup>Environmental Context

در چارچوب سیتو می باشد زیرا کاربر محور است و نیازهای کاربران را در یک ساختار سلسله مراتبی در نظر می گیرد.

## ۲-۴ آگاهی از موقعیت (موقعیت آگاهی)<sup>۱</sup>

با کار در زمینه نیازمندی های سیستم های نرم افزاری با دو اصطلاح به نام های آگاهی از موقعیت و آگاهی از شرایط آشنا می شویم که این دو مورد باهم تفاوت دارند. آگاهی از شرایط، انسان ها را کمتر در نظر گرفته و داده ها را در اولویت قرار می دهد. برعکس آگاهی از موقعیت، مستلزم ارائه داده های متنی به انسان ها است و کاربران را در اولویت قرار می دهد.

پس از استقرار یک برنامه نرم افزاری، داده هایی که از کاربران واقعی در زمان استفاده از سیستم به دست آمده و به روز هستند، نقش مهمی در تکامل سیستم ایفا می کنند. حالتی را در نظر بگیرید که دو فرد در یک زمینه رفتاری یا محیطی دارای حالات روانی کاملاً متفاوتی هستند پس موقعیت های متفاوتی تجربه خواهند کرد. و این حالات روانی و کل تأثیراتی که کاربران از محیط می گیرند آگاهی از موقعیت نامیده می شود.

در زمان نیازسنجی سیستم های نرم افزاری تجزیه و تحلیل دیدگاه کاربران از اهمیت ویژه ای برخوردار است به همین دلیل است که از مدل زمینه تصادفی مشروط برای استخراج الزامات استفاده می شود 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.

با این حال، باید به این مورد توجه کرد که این روش به دقت یادگیری مدل زمینه تصادفی متکی است و با ورودی داده های نادرست نتایج نادرستی به دست می آیند همچنین روش پیشنهادی بسیار متکی به داده های مشاهده ای است بنابراین خطرات مربوط به شرایطی که نادیده گرفته شده است می تواند قطعی نبودن ارزیابی الزامات را افزایش دهد.

---

<sup>۱</sup>Situation-Awareness

## ۲-۵ مدل‌های یادگیری تحت نظارت متوالی

در بسیاری از الگوریتم‌های طبقه‌بندی کلاسیک، داده‌های آموزش به‌طور مستقل از توزیع مشترک گرفته می‌شود. با این حال، در داده‌های متوالی، داده‌های نزدیک ارتباط تنگاتنگی باهم دارند [۱۸]؛ بنابراین، برای یادگیری و تجزیه روابط بین مشاهدات مختلف و برچسب‌های استنباط شده در یک دنباله مشاهده، یک مدل یادگیری مناسب مورد نیاز 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.

از نظر دقت در تبدیل گفتار به متن<sup>۱</sup> مشکلاتی که در این زمینه وجود دارد، نام‌گذاری یک موجودیت شناسایی شده<sup>۲</sup> و تشخیص دست خط<sup>۳</sup> بوده و ثابت شده است که زمینه تصادفی مشروط از مدل‌های مخفی مارکوف<sup>۴</sup> [۷] و حداکثر مدل‌های آنتروپی مارکوف<sup>۵</sup> [۱۹] بهتر عمل می‌کنند. این روش همچنین در تشخیص الگو و یادگیری ماشین [۲۰]، داده‌های توالی بیولوژیکی [۲۱] و تشخیص نفوذ [۲۲] استفاده شده 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 قادر است با استفاده از توابع ویژگی قابل استفاده به جای ماتریس احتمال، بر محدودیت‌های مدل‌های مخفی مارکوف غلبه کند.

---

<sup>۱</sup>Part of speech

<sup>۲</sup>Named entity recognition

<sup>۳</sup>handwriting recognition

<sup>۴</sup>Hidden Markov Models

<sup>۵</sup>Maximum Entropy Markov Models



# فصل سوم

## متدولوژی

### ۳-۱ روش تحقیق

در این سمینار برای دستیابی به اهداف غیر عملکردی از زمینه تصادفی مشروط استفاده شده است. زمینه تصادفی مشروط در واقع یک روش آماری تشخیص نیازمندی‌ها می‌باشد. نسخه زمینه تصادفی مشروط استفاده شده (زمینه تصادفی مشروط++)<sup>۱</sup> است زیرا با بررسی کردن دقت آن می‌بینیم که نسبت به سایر الگوریتم‌های زمینه تصادفی مشروط مانند اس جی دی، واپیتی<sup>۲</sup>، مالت<sup>۳</sup> و... دقت بالاتری دارد.

استخراج نیازهای غیر عملکردی در دو فاز صورت می‌گیرد:

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

۲. به دست آوردن نیازهای غیر عملکردی جدید با آنالیز و تجزیه و تحلیل روابط بین خواسته‌ها و در نظر گرفتن انتزاع‌های متفاوت.

مراحل کلی کار به صورت 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...) انجام شده است می‌توانیم اطلاعات به دست آمده را در قالب جدول در این پایگاه دانش ذخیره کنیم. مطابق با نیازهای به دست آمده 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 به دست آمده از مرحله قبل این کار انجام می‌شود. مدل زمینه تصادفی مشروط دو نوع الگوی متفاوت دارد: یونی گرام<sup>۴</sup> و بایگرام<sup>۵</sup> که یونی گرام روابط بین مشاهدات و خواسته‌های جاری را نشان می‌دهد در حالی که بایگرام رابطه بین مشاهدات و ترکیب خواسته‌های جاری و قبلی را شامل می‌شود.

---

<sup>۱</sup>CRF

<sup>۲</sup>Wapiti

<sup>۳</sup>MALLET

<sup>۴</sup>Unigram

<sup>۵</sup>Bigram

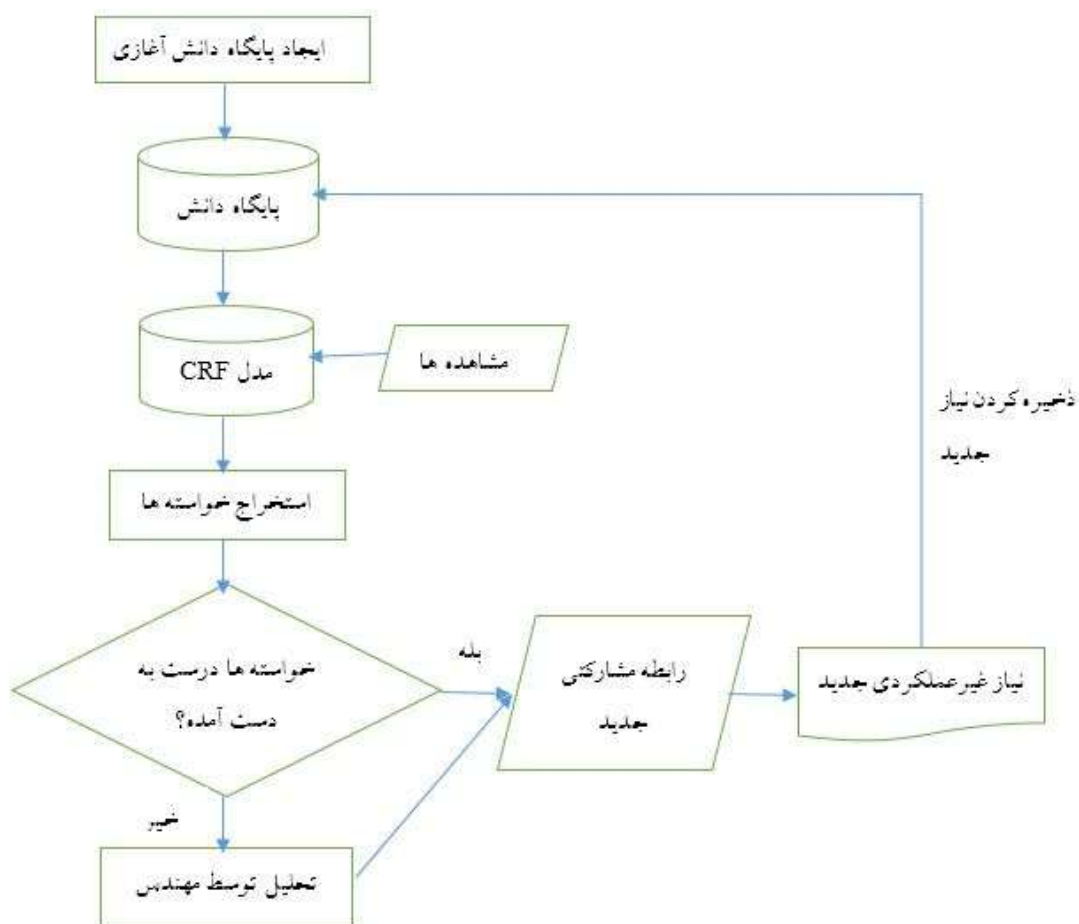
موردی که در این تحقیق بسیار مدنظر قرار گرفته وضعیت سیستم در مدت زمان خاصی است (منظور همان آگاهی از موقعیت است)، مثلاً فرض ما این است که کاربری در حال کار با یک سیستم نرم افزاری است و از سیستم درخواست مشاهده اطلاعات خود را دارد (با کلیک بر روی یک دکمه فرضی) کاربر از سیستم انتظار دارد که در مدت زمان کمی به درخواست او جواب بدهد حال فرض کنید سیستم نرم افزاری قادر به انجام درخواست نیست و این احتمال دارد به دلایل بسیاری اتفاق بیفتد مثلاً کلیک های مکرر کاربر امکان دارد باعث ایجاد وقفه در کار او شود و یا یک کار که در لحظه ای قبل انجام داده و بدون تمام کردن آن سیستم را بسته و دوباره از ابتدا شروع به کار کرده 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 انتخاب شده با استفاده از دو نوع الگوی ذکر شده (یونی گرام و بایگرام) تمام این سلسله رویدادها را در نظر می گیرد. مرحله سوم به دست آوردن خروجی مطلوب است که زمینه تصادفی مشروط برای تمام داده ها با استفاده از یک تابع وزن دار مقداری تحت عنوان اطمینان اکتسابی محاسبه می کند و این مقدار عددی مشخص می کند کدام رشته مشاهده ها امکان دارد یک خواسته در نظر گرفته شود. سپس با استفاده از توابع نرمال سازی به هر کدام از رشته ورودی ها مقدار احتمال مابین ۰ و ۱ تعلق می گیرد.

رشته ورودی های با بالاترین احتمال به عنوان نتیجه مناسب مدنظر قرار می گیرند و نتایج به دست آمده از این طریق نیازمندی های غیر عملکردی سیستم هستند.

در شکل ۳-۱ فلوچارت کلی نحوه استخراج نیازهای غیر عملکردی نشان داده شده است:

پس از تشخیص نیازمندی ها برای آن ها درخت نیازمندی ها را تشکیل می دهیم، در این درخت گره های پایینی زیرمجموعه گره های بالایی هستند و گره های پدر به گره های فرزندان تجزیه می شوند. نسبت به محل قرارگیری نیازمندی ها آن ها را به دودسته نیازهای سطح پایین (خواسته هایی هستند که به برگ درخت نزدیک تر هستند) و نیازهای سطح بالا (خواسته هایی هستند که به ریشه درخت نزدیک تر

هستند). تقسیم می‌کنیم همچنین برای هر نیازی بررسی می‌شود که این نیاز مقدار ضریب اطمینان بالایی دارد یا خیر؟



شکل (۳-۱): 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
خواسته سطح پایین	ضریب اطمینان زیاد	اچ ال	اچ اچ
	ضریب اطمینان کم	اچ ال	اچ اچ

جدول (۳-۱): روابط ایجاد شده بین نیازها از طریق سیستم زمینه تصادفی مشروط

این جدول چهار حالت دارد:

حالت اول: نیازهای سطح بالا و پایین با ضریب اطمینان بالا (اچ)

در این حالت نیازی که شناسایی شده 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 به دست آمده 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<sup>۲</sup>  
(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 (اچ ال)

در این شرایط خواسته سطح بالا ضریب اطمینان بالایی دارد اما فرزندان آن که نیازهای سطح پایین هستند ضریب اطمینان کمی دارند، از کم بودن این مقدار نتیجه می‌گیریم که این نیازها در پایگاه دانش

---

<sup>۱</sup>High confidence high and low level desires(hh)

<sup>۲</sup> Low confidence high level desires and high confidence low level desires (lh)

<sup>۳</sup>High confidence high level desires and low confidence low level desires(hl)

ندارند و این احتمال وجود دارد که زمینه تصادفی مشروط آن‌ها را به درستی 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 (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، پس از آن باید تلاش کنیم نیازهای جایگزین مناسب را به دست آوریم.

برخی از روش‌های آماری که برای تحلیل نتایج در این تحقیق استفاده شده است به صورت زیر است:

۱. آزمون مان - ویتنی:<sup>۱</sup> جزء آزمون‌های غیر پارامتری است و برای سنجش تفاوت میان نمونه‌ها به کار می‌رود. در این آزمون رتبه‌بندی روی می‌دهد و محاسبات بر روی رتبه‌ها انجام می‌گیرد.
۲. آزمون آماری مقدار پی:<sup>۲</sup> 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 ( $\leq 0,05$ ) تئوری تهی را به طور کامل رد می‌کند. مقدار پی بزرگ ( $> 0,05$ ) نمی‌تواند تئوری تهی را به طور کامل رد کند.

### ۳-۲ جامعه و نمونه آماری

برای انجام تحقیق و به دست آوردن داده‌ها در این سمینار از سیستم محیط تحقیقاتی تعاونی<sup>۳</sup> استفاده شده است. محیط تحقیقاتی تعاونی یک سیستم نرم‌افزاری متن‌باز است که مطابق با نیاز این تحقیق توسعه داده شده 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 ...

<sup>۱</sup>Low confidence high level desires and low confidence low lwwwl desires(II)

<sup>۲</sup>Mann-Whitney U test

<sup>۳</sup>P values

<sup>۴</sup>Cooperative Research Environment

شرکت‌کننده‌ها در این تحقیق از گروه‌های متفاوت علوم کامپیوتر و از سه دانشگاه متفاوت شامل دانشگاه ایالتی آیووا در ایالات متحده، دانشگاه نورث ایسترن چین و دانشگاه نیهون ژاپن انتخاب شده‌اند. در مجموع ۱۴۷ نفر در این تحقیق شرکت کرده‌اند که ۷۸٪ آن‌ها با علوم کامپیوتر آشنایی داشته‌اند میانگین آشنایی شرکت‌کننده‌ها با سیستم محیط تحقیقاتی تعاونی حدود ۳,۳ است که نشان‌دهنده آشنایی جزئی آن‌ها است. حدود ۸۵٪ شرکت‌کننده‌ها حداقل مدرک تحصیلی لیسانس داشته‌اند، سه زبان چینی، ژاپنی و انگلیسی به ترتیب بیشتر از سایر زبان‌ها استفاده شده است.

مدت زمان انجام این تحقیق ۶ ماه بوده، از ماه مارچ تا سپتامبر سال ۲۰۱۸، این داده‌ها زمانی جمع‌آوری شده‌اند که مدل زمینه تصادفی مشروط طراحی نشده بود بنابراین از یک متخصص نیز در این زمینه کمک گرفته شده و فرضیات مطرح شده توسط متخصص برای موارد آموزشی ذخیره شد. در مجموع تعداد ۱۱۹۶۰ سطر داده از ۹۴۹ مورد کار با سیستم جمع‌آوری شده است، منظور از کار با سیستم تمام فعالیت‌هایی است که از زمان ورود به سیستم تا زمان خروج از آن انجام شده 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 کار نکرده‌اند 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 ۴۳۸۸ سطر کاهش پیدا کرد که این تعداد از ۷۹۲ مورد کار با سیستم به دست آمده بود.

### ۳-۳ ابزار اندازه‌گیری

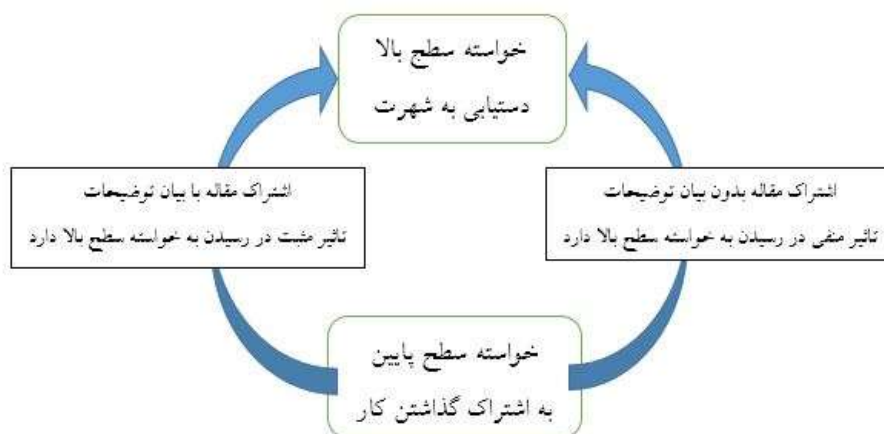
در این قسمت با بررسی یک مثال از روش حل مسئله ابزار اندازه‌گیری را کامل توضیح می‌دهیم: برای جمع‌آوری اطلاعات این تحقیق از سیستم تحت وبی به عنوان محیط تحقیقاتی تعاونی استفاده شده است محیطی که به منظور اشتراک مقالات علمی و بحث و تبادل نظر در این زمینه طراحی شده، در این محیط کاربر می‌تواند ثبت نام کند، اطلاعات خود را ثبت کند، مقاله یا مقالاتی را در این سامانه بارگذاری کند، راجب مطالب مختلف بحث و تبادل نظر کند و ... این امکان تعبیه شده 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، از آنجایی که 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.

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



شکل (۳-۲): رابطه سلسله مراتبی بین نیاز یافتن همکار و نیازهای سطح پایین آن



شکل (۳-۳): رابطه بین خواسته رسیدن به شهرت و خواسته به اشتراک گذاشتن کار

با توجه به شکل ۳-۱ 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» انجام شده و ستون سوم رشته کارهایی که برای رسیدن به خواسته سطح بالای «رسیدن به شهرت»<sup>۱</sup> انجام شده است.

زمان	رویدادهای خواسته یافتن همکار	رویدادهای خواسته رسیدن به شهرت
۱	کلیک منوی تمام مقالات (موفق ۲ ثانیه) درخواست‌های (جستجوی همکار، گفتگو)	کلیک منوی آپلود مقالات
۲	کلیک فیلتر (موفق ۶ ثانیه) نیازهای (گفتگو، جستجوی همکار)	کلیک نظرات (موفق ۳ ثانیه) درخواست‌های (اشتراک کار، جستجوی همکار)
۳	کلیک نظرات (موفق ۳ ثانیه) نیازهای (گفتگو، جستجوی همکار)	کلیک ارسال (موفق ۳۳۱ ثانیه) نیازهای اشتراک کار، مشهور شدن
۴	کلیک ارسال (موفق ۸۷ ثانیه) نیازهای (گفتگو، جستجوی همکار)	کلیک آپلود مقاله (موفق ۴ ثانیه) نیازهای (مشهور شدن؛ اشتراک کار)

جدول (۲-۳): نمونه داده‌های آموزشی برای نرم‌افزار زمینه تصادفی مشروط

هر سطر این جدول شامل یک مجموعه کار است مثلاً در نقطه زمانی ۴ ستون دوم نشان می‌دهد که دکمه ارسال<sup>۳</sup> زده شده و سیستم به این درخواست به درستی عمل کرده و این عمل ۸۷ ثانیه طول کشیده و در مدت زمان کار کاربر با سیستم دو درخواست از سیستم داشته، خواسته سطح بالای جستجوی همکار و خواسته سطح پایین بحث و گفتگو؛

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 به عنوان داده آموزشی سیستم زمینه تصادفی مشروط وارد این سیستم می‌شوند تا سیستم بتواند عملکرد درست بر روی ورودی‌ها را از این طریق بیاموزد.

<sup>۱</sup>Look for collaborator

<sup>۲</sup>Build reputation

<sup>۳</sup>Submit

<sup>۴</sup>Discuss

داده‌هایی که شامل عمل، نقطه زمانی و خواسته‌ها هستند ورودی سیستم زمینه تصادفی مشروط بوده و خروجی این سیستم مشابه جدول ۳-۱ روابط ایجادشده بین خواسته‌ها را نشان می‌دهد.

یونی گرام	بایگرام
%X[0,0] یونی گرام ۰۱:	%X[0,0] بایگرام ۰۱:
%X[-1,0]/%X[0,0] یونی گرام ۰۲:	%X[0,0]/%X[1,0] بایگرام ۰۲:
%X[0,0]/%X[1,0] یونی گرام ۰۳:	%X[-1,0]/%X[0,0] بایگرام ۰۳:

جدول (۳-۳): 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 تعیین شده 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
۳۱۱	۰,۵۸۲	به اشتراک گذاشتن کار	به شهرت رسیدن
۱۸۴	۰,۳۴۵	بازنگری مقاله	
۳۹	۰,۰۷۳	بحث و گفتگو	
۲۶۶	۰,۵۸۷	بحث و گفتگو	جستجو برای همکار
۱۸۷	۰,۴۱۳	به اشتراک گذاشتن کار	

جدول (۳-۴): وزن به دست آمده برای خواسته‌های سطح بالا و سطح پایین

از طریق روابط بین خواسته‌ها به این نتیجه می‌رسیم که خواسته رسیدن به شهرت از دو طریق امکان پذیر است: ۱- به اشتراک گذاشتن کار (با وزن ۰,۵۸۲) ۲- دو رویداد تجدیدنظر کردن در کار به اشتراک گذاشته شده و بحث و گفتگو که وزن هردوی آن‌ها به عنوان رشته عملکرد محاسبه می‌شود (وزن = ۰,۳۴۵ و ۰,۰۷۳) اگرچه وزن حالت اول بیشتر است اما با بررسی عملکرد کاربران و نظر مهندس نرم افزار حالت دوم برای پیاده سازی انتخاب می‌شود.

در مورد نیاز جستجوی همکار از بررسی روابط به دست آمده توسط مدل زمینه تصادفی مشروط به رابطه جدیدی نرسیدیم اما وزن های به دست آمده نشان داد که کاربران بیشتر تمایل دارند از طریق بحث و گفتگو به نتیجه برسند بنابراین این مورد می تواند 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 فراهم شده.

مدل زمینه تصادفی مشروط برای تمام خواسته ها که به صورت ورودی داده می شود موارد گفته شده را محاسبه می کند و به این صورت می توانیم نیازهای غیر عملکردی را از طریق روابط به دست آمده استخراج کنیم.

## فصل چہارم

## تجزیہ و تحلیل

## ۴-۱ تجزیه و تحلیل

این سمینار روشی را برای تجزیه و تحلیل 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 تجزیه و تحلیل نیازهای غیر عملکردی از دیدگاه کاربران صورت می‌گیرد. این چارچوب می‌تواند نه تنها 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. با این حال، به منظور 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 شروط 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 معرفی شده در این سمینار بیشتر در زمانی 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 profiles 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. Tibermachine, 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. Classification of nfrs for information System International Journal of Computer Applications, 115(22), 2015.
- [7] J. Eckhardt, A. Vogelsang, and D. M. Fernandez. 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 engineering: a roundtrip from research to practice [engineering 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. Settini, X. Zou, and P. Solc. Automated classification Of non-functional requirements. *Requirements Engineering*, 12(2):103{120, 2007

[16] S. S. Yau, H. Gong, D. Huang, W. Gao, and L. Zhu. Specification, 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. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[21] Y. Liu, J. Carbonell, P. Weigele, and V. Gopalakrishnan. Protein fold recognition using segmentation conditional random fields (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.