

گزارش توضیح‌پذیری سیستم‌های نرم‌افزاری: از آنالیز نیازمندی‌ها تا ارزیابی

سیستم [۱]

ملیکا محمدی گل

علیرضا سلطانی نشان

سودابه آشوری

a.soltani@iau-tnb.ac.ir

۱۸ تیر ۱۴۰۳

۱ مقدمه و بیان مسئله

دوران الگوریتمیک یا Algorithmic age منحصر به زمان کنونی و آینده جهان است، به گونه‌ای که الگوریتم‌های گوناگونی در زندگی انسان‌ها وارد شده‌اند. از ساده‌ترین عملیات گرفته تا پیچیده‌ترین تصمیم‌گیری‌ها را می‌توان پیاده‌سازی کرد. این الگوریتم‌ها حوزه‌های مختلف از قبیل اقتصاد، بهداشت، حمل و نقل و غیره را تحت تاثیر خود قرار داده است. از انتخاب کوتاه‌ترین مسیر رانندگی گرفته تا پیش‌بینی و تشخیص سرطان و طراحی پروتئین و داروهای مختلف.

این برگه به روشنی از قابلیت‌های این سیستم‌ها، که تصمیم‌گیری‌های مختلف را به گونه‌ای انجام می‌دهند که کاربر نمی‌داند که این پیش‌بینی چگونه رخ داده است یا این ماشین با چه داده‌هایی به چنین نتیجه‌ای رسیده است را Black box systems می‌نامد. زمانی که کاربر نمی‌تواند متوجه شود که پشت صحنه این پیش‌بینی‌ها به چه شکلی می‌باشد می‌گوییم که این سیستم‌ها فاقد شفافیت^۱ هستند. فاقد شفافیت نگرانی‌هایی را در رابطه با پارامترهایی مانند مسئولیت‌پذیری، انصاف و پیامدهای اخلاقی ایجاد می‌کند به ویژه در حوزه‌هایی که سیستم تصمیماتی را می‌گیرد که می‌تواند روی مردم و جامعه به طور کلی تاثیر گذار باشد.

بر همین اساس، این برگه بحث‌هایی را در مورد شفافیت و اخلاق سیستم‌های مدرن مطرح کرده است. درک چگونگی ادغام این نگرانی‌ها در سیستم‌ها و در نتیجه نحوه برخورد با آنها در طول مهندسی نرم‌افزار و مهندسی نیازمندی‌ها بسیار مهم است.

فهرست مطالب

۱	۱	مقدمه و بیان مسئله
۵	۲	رویکرد و راه حل
۵	۱.۲	توضیح پذیری چیست؟
۵	۲.۲	چالش های توضیح پذیری
۵	۱.۲.۲	پیچیدگی سیستم ها
۵	۲.۲.۲	طبیعت Black box
۵	۳.۲.۲	زمینه گرایی یا توضیح Subjectivity of Explanation
۶	۴.۲.۲	رابطه الکلنگی بین توضیح پذیری و عملکرد
۶	۵.۲.۲	سیستم های پویا و در حال تکامل
۷	۶.۲.۲	اعتبارسنجی و اعتماد
۷	۳	سابقه دانشی و مدل پیشنهادی
۷	۱.۳	تعاریف یا Definitions
۷	۲.۳	مدل ها یا Models
۸	۱.۲.۳	مدل های مفهومی یا Conceptual models
۸	۲.۲.۳	مدل های کیفی یا Quality models
۹	۳.۲.۳	مدل های مرجع یا Reference models
۱۱	۳.۳	راهنمای شناختی یا Catalogues
۱۲	۴	بررسی رسالت مقاله
۱۲	۱.۴	سوال های پژوهشی
۱۲	۲.۴	استراتژی جست و جو در SLR
۱۲	۱.۲.۴	جست و جوی دستی
۱۲	۲.۲.۴	جست و جوی گلوله برفی برای تجمیع و تکمیل نتایج جست و جو
۱۴	۳.۴	نتایج: فرآورده های پژوهشی
۱۴	۴.۴	راه حل این فرآورده ها
۱۴	۵.۴	جنبه هایی که در این طرح باید توضیح داده شوند
۱۴	۶.۴	نتیجه راهنمای شناختی (دانشی)
۱۵	۷.۴	تجزیه و تحلیل نیازمندی ها
۱۵	۸.۴	اهداف
۱۶	۹.۴	محدودیت ها
۱۶	۱۰.۴	استراتژی پیاده سازی
۱۶	۱۱.۴	بررسی و دو مرحله برای پیاده سازی توضیح پذیری
۱۶	۵	روال استخراج اطلاعات
۱۶	۱.۵	نتیجه استخراج اطلاعات
۱۷	۲.۵	فرم ارائه
۱۷	۳.۵	استفاده از ماژول توضیحات

۱۷	۶ ارزیابی
۱۷	۱.۶ سطوح ارزیابی برای توضیح پذیری
۱۷	۱.۱.۶ ارزیابی در سطح سیستم
۱۷	۲.۱.۶ ارزیابی در سطح توضیح
۱۸	۲.۶ روش‌های ارزیابی
۱۸	۳.۶ بررسی بازخورد کاربران نهایی
۱۸	۴.۶ پرسشنامه
۱۸	۵.۶ آزمون‌های A/B
۱۸	۶.۶ مطالعات موردی
۱۸	۷.۶ مصاحبه
۱۹	۷ معیارها
۱۹	۸ کلیات و مباحث
۱۹	۱.۸ محدودیت‌ها و تهدید اعتبار پژوهش
۱۹	۱.۱.۸ انجام یک مرور سیستماتیک و فرآیند کدگذاری
۲۰	۲.۱.۸ کارگاه‌ها
۲۰	۹ نتیجه‌گیری نهایی و ایده‌ها

فهرست تصاویر

۱۰	مدل مرجع برای پشتیبانی از توسعه سیستم‌های توضیح‌پذیر	۱
۱۰	یک مدل مفهومی که تاثیر توضیح‌پذیری را در ابعاد کیفی مختلف نشان می‌دهد.	۲
۱۳	بررسی ساختار SLR انجام شده در این مقاله	۳
۱۵	راهنمای دانشی در جهت توضیح‌پذیری و تاثیر آن در جنبه‌های کیفی دیگر	۴

۲ رویکرد و راه حل

رویکرد و روش‌شناسی‌ای که این مقاله در مورد آن صحبت می‌کند توضیح‌پذیری در سیستم‌های نرم‌افزاری و حتی مدل‌های هوش مصنوعی است تا بتواند ضعف عدم شفافیت سیستم‌ها را رفع کند. در این مقاله، بنا به ضرورت توضیح‌پذیری سیستم‌های نرم‌افزاری، به صورت روشی مناسب، سعی بر آن دارد تا توضیح‌پذیری را بیان کند و به صورت شفاف به عنوان یک نیازمندی غیرعملیاتی ابعاد مختلف آن را روشن سازد. همچنین در مورد رابطه بین جنبه‌های کیفی نیز صحبت می‌کند.

۱.۲ توضیح‌پذیری چیست؟

توضیح‌پذیری یک روش مفید است تا از نگرانی‌های اخلاقی نرم‌افزارها و مدل‌ها بکاهد. به معنای قابلیت شرح نرم‌افزار و سیستم است. وقتی یک سیستم یا مدل هوش مصنوعی توضیح‌پذیر است، به این معناست که عملکرد و تصمیمات آن قابل تفسیر و توجیه است. به عبارت دیگر، می‌توان به راحتی فهمید که یک سیستم به چه شکلی کار می‌کند و چگونه به تصمیمات خود رسیده است. توضیح‌پذیری یک ویژگی بسیار مهم در سیستم‌های نرم‌افزاری است که موجب افزایش اعتماد به آن می‌شود و ارزش‌های اخلاقی و قانونی را در رابطه با سیستم تعریف خواهد کرد. امروزه به مسئله توضیح‌پذیری سیستم‌ها بسیار اهمیت داده می‌شود و یکی از مهم‌ترین نیازمندی‌های غیرعملیاتی^۲ محسوب می‌شود. در حالی که به کاربران این اجازه را می‌دهد که خودشان بتوانند انتخاب کنند که از این سیستم استفاده کنند یا از آن دوری کنند چرا که بر روی رابطه قابلیت اعتماد و اتکای سیستم بسیار تاثیرگذار می‌باشد. نکته: با توجه به قدرت هوش مصنوعی در تمام حوزه‌های زندگی بشر، توضیح‌پذیری به عنوان یکی از مهم‌ترین پایه‌های اعتماد در نیازمندی‌های نرم‌افزار می‌باشد.

۲.۲ چالش‌های توضیح‌پذیری

دلایل زیر نشان‌دهنده آن است که جمع‌آوری و استخراج داده، مذاکره و اعتبارسنجی در فرایند توضیح‌پذیری با چالش‌هایی رو به رو می‌باشد:

۱.۲.۲ پیچیدگی سیستم‌ها

در سیستم‌هایی که مبنی بر هوش مصنوعی و فرایند یادگیری ماشین هستند با وجود الگوریتم‌های مختلف که وظیفه تصمیم‌گیری را در سیستم دارند، سطح پیچیدگی بسیار بالا می‌باشد. درک و توضیح این سیستم‌ها با فرایندهایشان برای کاربران مختلف به مفهوم ساده، بسیار سخت و غیرقابل درک می‌باشد.

۲.۲.۲ طبعیت Black box

از نظر کاربران، بسیاری از الگوریتم‌ها به شکل جادویی عمل می‌کنند، بدان معنا که فرایندهای داخلی این الگوریتم‌ها کاملاً به صورت مات می‌باشد و توسط انسان بدون دانش قبلی به راحتی قابل درک نیست.

۳.۲.۲ زمینه‌گرایی توضیح یا Subjectivity of Explanation

زمینه‌گرایی توضیح به معنای نسبی بودن یا وابستگی توضیحات به نگرش و دیدگاه فردی است. در حالت کلی تفسیر هر چیزی توسط ذینفعان می‌تواند کاملاً متفاوت از نظر معنا و دیدگاه باشد. مذاکره برای به اجماع رسیدن در سطح و نوع توضیح مورد نیاز می‌تواند چالش برانگیز باشد، به ویژه زمانی که با دیدگاه‌ها و علایق گوناگون سروکار داریم.

^۲Non-functional

عدم درک مشترک^۳

یکی دیگر از دشواری‌ها، ارتباطات مناسب در مهندسی نیازمندی است. ذینفعان بیرونی و تیم توسعه ممکن است ناخواسته از یکسری کلمات متفاوت با مفهوم یکسان استفاده کنند که در نهایت باعث ایجاد سوءتفاهم و نقص فهم مشترک بین افراد شود که در نهایت چالشی برای ارتباط با یکدیگر ایجاد می‌کند. لازمه کارآمدی ارتباطات درک مشترک از مفاهیم می‌باشد که ریسک دوباره کاری و نارضایتی ذینفعان را کاهش می‌دهد.

رویکرد درک مشترک بین افراد

مهندسان نرم‌افزار می‌توانند مجموعه‌ای از فرآورده‌ها را ایجاد کنند که باعث ایجاد درک و فهم مشترک در پروژه‌های نرم‌افزاری می‌شود و بارها قابل استفاده مجدد و اصلاح خواهند بود تا فرآورده‌ها، محصولی از مذاکره با زبانی مشترک بین افراد باشد.

فرآورده‌ها

فرآورده‌ها هر گونه اسناد متنی و اشکال گرافیکی هستند که به دور از کدها و محصولاتی نرم‌افزاری، ابزاری برای مذاکره بین تمام افراد حاضر (چه ذینفعان چه مهندسان مختلف) می‌باشند. محتوای فرآورده‌ها معمولاً اشکال، متن‌ها، مدل‌های بصری، فهرست‌ها، چارت‌ها، چهارچوب‌ها و مدل‌های کیفیت می‌باشد. این فرآورده‌ها در شکلهای ساختار پروژه بسیار کارآمد هستند به گونه‌ای که در فرایندهای مهندسی نیازمندی از قبیل، مدل‌های مفهومی^۴ کاتالوگ دانش^۵ و مدل‌های مرجع^۶ کاربرد متعددی دارند.

۴.۲.۲ رابطه الکلنگی بین توضیح‌پذیری و عملکرد

با افزایش توضیح‌پذیری ممکن است کارایی سیستم کاهش یابد. در نتیجه لازم است بین این نیازمندی با سایر نیازمندی‌ها تعادل ایجاد شود. راهکار ایجاد تعادل می‌تواند موارد زیر باشد:

۱. کاهش پیچیدگی توضیحات: به جای آنکه توضیحات کاملی در مورد عملکرد الگوریتم‌های هوش مصنوعی به ازای هر مورد فراهم شود، سیستم می‌تواند بسیار ساده با ارائه خلاصه‌ای مفید، فاکتورهای مهم و اساسی دلیل انتخاب موارد به عنوان توصیه کاربر را مشخص کند.

۲. استفاده از متدهای فنی در مهندسی نرم‌افزار مانند فرایندهای کش کردن انتخاب‌های کاربر (براساس کلیک‌های مختلف روی محصولات یا مدت زمانی که روی محصول مورد نظر کاربر مطالعه داشته) محاسبات از پیش تعیین شده‌ای در مورد چرایی انتخاب محصول به عنوان توصیه را مشخص کند.

انتخاب استراتژی مناسب برای حفظ توضیح‌پذیری به همراه سرعت و کارایی بالا در عملکرد سیستم توصیه‌گرا، یک تریدآفی است که وظیفه آن بر عهده تیم توسعه، طراح و معماری نرم‌افزار می‌باشد.

۵.۲.۲ سیستم‌های پویا و در حال تکامل

یکی از چالش‌های مهم توضیح‌پذیری، سیستم‌هایی است که در طول زمان دچار تغییرات کلی به ویژه در نیازمندی‌ها می‌شوند. اینکه توضیحات متناسب با تغییر سیستم‌ها به روز شود چالشی مهم است.

^۳ Lack of shared understanding

^۴ Conceptual models

^۵ Knowledge catalogues

^۶ Reference models

اعتبار بخشیدن به توضیحات ارائه شده توسط یک سیستم می تواند دشوار باشد، به ویژه زمانی که آنها شامل فرآیندها یا داده های پیچیده باشند. ایجاد اعتماد در این توضیحات مستلزم روش های اعتبارسنجی قوی و شفافیت در فرآیند تولید توضیح است. همچنین اشاره می کند که مهندسی نیازمندی فرایند ساده برای شناسایی و مشخص کردن نیازمندی ها نیست، بلکه فرایندی جهت حمایت از ارتباطات کارآمد این نیازمندی ها بین ذینفعان مختلف می باشد. مهم ترین مشکل در زمینه توضیح پذیری، کمبود دانش ساخت یافته می باشد. اگر چنین دانشی به وجود بیاید به مهندسی نرم افزار کمک می کند فاکتورهایی را طی روند پیاده سازی سیستم های توضیح پذیر در نظر بگیرد تا به درک مشترک کمک کند.

۳ سابقه دانشی و مدل پیشنهادی

برای اجماع فهم مشترک بر مسائل مختلف این حوزه از مهندسی نرم افزار، خواننده نیاز دارد که با مفاهیم زیر به صورت کلی آشنا باشد تا بتواند:

۱. از دانشی فراتر از زمینه های خود استفاده کنند و از این دانش برای رفع نیازهای یک پروژه خاص (جاری یا جدید) استفاده کنند.
 ۲. دستیابی به درکی مشترک که منجر به ارتباطات بهتر و تعریف نیازمندی های سیستم به شکل «درست» می شود.
- در این مقاله توضیح پذیری به صورت چهار فرآورده^۷ به صورت روشمند و منطبق با چرخه عمر نرم افزار تبیین شده است. این چهار سطح عبارتند از:

۱.۳ تعاریف یا Definitions

تعاریف در مهندسی نرم افزار و مهندسی نیازمندی، راهنمایی تقریبی برای مهندسان نرم افزار در مورد دامنه، عناصر و هر چیز دیگری را ارائه می دهند. به این صورت که یکی از مهم ترین مراحل تسهیل ارتباطات برای یک موضوع یا یک مفهوم می باشند. وقتی در مورد تعریف جنبه های کیفی^۸ صحبت می شود در حقیقت منظور همان راهنمایی ها برای مهندسان نرم افزار است که به آنها در فهمیدن روند مهندسی نیازمندی به خصوص تضمین کیفیت^۹ کمک می کند. عدم اجماع حاضرین بر سر مفاهیم و تعاریف می تواند سبب ایجاد نتیجه نامناسبی در مشخصات و یکپارچگی نیازمندی های غلط شود. برای مثال وقتی در جلسات در مورد Usability صحبت می شود، برخی از افراد توسعه دهنده منظور را در رابطه با استفاده و بهره وری بلند مدت^{۱۰} می دانند و برخی از افراد منظور را در استفاده آسان محصول^{۱۱} می دانند.

۲.۳ مدل ها یا Models

یک مدل در بالاترین سطح تجرید در مورد کارایی سیستم تمرکز دارد که بتواند تمام جنبه های سیستم را به سادگی و به دور از جزئیات نمایش دهد. هیچ وقت یک مدل به تنهایی، تمام سیستم را تشریح نمی کند. مدل ها در طیف گسترده ای از جنبه های مختلف یک سیستم استفاده می شوند. مدل ها می توانند برای اهداف توسعه نرم افزار و توسعه کسب و کار استفاده شوند. از نظر نرم افزاری می توانند نقش مهمی در تعریف ساختار نرم افزار یا پیکربندی ها داشته باشند و از سوی دیگر برای توصیف و بهینه سازی نگرانی های سازمانی مانند فرآیندها و حوزه های تجاری می توانند بسیار مفید باشند. از انواع مدل ها می توان به موارد زیر اشاره کرد:

- Artifact^۷
- Quality aspects^۸
- Quality assurance^۹
- Long-term efficiency^{۱۰}
- Ease of Use (EoU)^{۱۱}

۱.۲.۳ مدل‌های مفهومی یا Conceptual models

مدل‌هایی هستند که برای تعریف و توصیف یک مفهوم استفاده می‌شوند. افراد را قادر می‌سازند که بتوانند ساختار و ویژگی‌های یک جنبه کیفی مشخص را در طول فرآیند تحلیل نیازمندی درک کنند. دانش مورد نیاز برای توسعه مدل‌های مفهومی معمولاً از Literature، تجارب قبلی مشابه و حوزه تخصصی استخراج می‌شوند. در این مدل سه مفهوم عمده کلاس‌های ذینفعان، ابعاد استخراج و طیف کیفیت مورد استفاده قرار گرفته است:

کلاس‌های ذینفعان

طبقه‌بندی جنبه‌های کیفیت که تحت تأثیر توضیح‌پذیری هستند، بر اساس طبقات ذینفعان (کاربران، توسعه دهندگان، گروه‌های تحت تأثیر، تنظیم کنندگان) صورت می‌گیرد. درک توضیح‌پذیری برای هر طبقه، بسته به نیازها و مشخصه‌های خاص آن می‌باشد.

ابعاد استخراج

شش بعد برای شناسایی توضیح‌پذیری و استخراج آن مورد توجه قرار می‌گیرد:

۱. نیازها و انتظارات کاربران نهایی^{۱۲}.

۲. ارزش‌ها و اصول فرهنگ سازمانی که در تعیین استراتژی و طراحی سیستم نقش و تأثیر دارد.

۳. اهداف و مأموریت‌های سازمان

۴. قوانین و هنجارها و استانداردهای قانونی و نظاری مرتبط با سیستم

۵. جنبه‌های دامنه، محدوده و موضوعی که سیستم روی آن اعمال می‌شود.

۶. محدودیت‌های عملی و غیرفنی پروژه مانند منابع مالی، زمان، فناوری و نیرو انسانی که در تصمیم‌گیری‌های مربوط به عملیات سیستم تأثیر گذار می‌باشد.

۲.۲.۳ مدل‌های کیفی یا Quality models

مدل‌هایی که صفات کیفی و ویژگی‌های نیازمندی نرم‌افزار را تعریف و مشخص می‌کنند. این مدل‌ها معمولاً روش‌های سیستماتیکی را برای ارزیابی و اطمینان کیفی نیازمندی‌ها بر اساس استاندارد معتبر بکار می‌گیرند. این مدل‌ها به ذینفعان کمک می‌کنند تا درک کنند که لازمه بالا بودن کیفیت خدمات چیست و دستورالعمل‌هایی را برای ایجاد، تجزیه و تحلیل و اعتبارسنجی نیازمندی‌ها ارائه می‌دهد.

۱. استاندارد ISO/IEC 25010 (SQuaRE) [۹]: یک استاندارد جامع برای نیازمندی‌ها و ارزیابی کیفیت نرم‌افزار است. ویژگی‌های مورد بررسی آن از قبیل، عملکرد، قابلیت استفاده، کارایی، قابلیت نگهداری و غیره می‌باشند.

۲. مدل Quality Function Deployment (QFD) [۵]: رویکردی مشتری محور است که مهم‌ترین وظیفه آن ترجمه نیازهای مشتری به نیازمندی‌های محصول خاص است. این مدل به مهندسان مخصوصاً مهندسان نیازمندی کمک می‌کند تا بتوانند تطابق نیازمندی‌های پیاده‌سازی شده را با انتظارات و ترجیحات مشتریان بررسی کنند.

۳. McCall's Quality Model [۳]: این مدل توسط جان.دی. مک‌کال مطرح شده است و ۱۱ عامل کیفی را که در سه دسته: عملکرد محصول، بازبینی محصول و جا به جایی و انتقال محصول قرار دارند، را شناسایی می‌کند. این عوامل شامل قابلیت اعتماد، قابلیت استفاده، کارایی و ویژگی‌های مشابه با استاندارد SQuaRE هستند.

^{۱۲}End users

۴. IEEE 730 Standard [۶]: این استاندارد فرآیندهای اطمینان کیفیت را برای پروژه‌های توسعه نرم‌افزار از جمله مهندسی نیازمندی‌ها را تعریف می‌کند. فعالیت‌های مرتبط با برنامه‌ریزی کیفیت، اطمینان کیفیت و کنترل کیفیت را در طول چرخه عمر توسعه نرم‌افزار شامل می‌شود.

۵. CMMI (Capability Maturity Model Integration) [۲]: یک چهارچوب برای بهبود فرآیندهای مرتبط با توسعه و نگهداری سیستم‌های نرم‌افزاری را ارائه می‌دهد. فعالیت‌های این چهارچوب شامل روش‌های مرتبط با مدیریت نیازمندی‌ها می‌باشد و اهمیت مدیریت و اطمینان از کیفیت نیازمندی‌ها را در طول چرخه توسعه تأکید می‌کند. در این استاندارد تمام فرآیندهای مختلف توسعه نرم‌افزار از پایه‌ای‌ترین سطح تا بالاترین سطح ارزیابی می‌شوند و امتیازی برای هر سطح متناسب با میزان تسلط و بهره‌وری به آنها اختصاص می‌یابد. به کمک این امتیازات و نمرات مشخص می‌شود که هر سطح به چه میزانی رشد و بهبود داشته است.

منظور از Blueprint در حوزه توسعه نرم‌افزار و چرخه نیازمندی‌ها

معمولاً جزئیات مشخصی را در مورد ساختارها، طراحی بخش‌ها و فانکشنالیتی یک سیستم نرم‌افزاری یا اپلیکیشن بیان می‌کند. این Blueprint به برنامه نویسان و توسعه‌دهندگان کمک می‌کند تا از آن به عنوان راهنما در حوزه فهم مناسب از اینکه چه چیزی باید پیاده‌سازی شود، استفاده می‌کنند. در مهندسی نیازمندی‌ها هم ممکن است مانند همین مفهوم به کار رود با این تفاوت که نیازمندی‌های functional و Non-functional در کنار یکدیگر مطرح می‌شوند تا معماری و ساختار نرم‌افزار را به وسیله مدل‌های داده محور طراحی و توسعه دهند.

۳.۲.۳ مدل‌های مرجع یا Reference models

مدلی که شامل حداقل مقدار از مجموعه‌ای از مفاهیم، بدیهیات و روابط در یک دامنه مسئله خاص می‌باشد و به هیچ یک از استانداردها، فناوری‌ها، کدهای نوشته شده، اجراها یا سایر جزئیات وابستگی ندارد. مدل‌های مرجع می‌توانند به صورت Blueprint در مهندسی نرم‌افزار استفاده شوند تا بتوانند زیرساخت نرم‌افزاری را ارائه دهند. مدل مرجع شامل عوامل مرتبطی است که برای توسعه سیستم‌های توضیح‌پذیر در مراحل مختلف چرخه عمر نرم‌افزار باید در نظر گرفته شود. مدل مرجع، به مهندسان نرم‌افزار در تجزیه و تحلیل، پیاده‌سازی عملیات، و ارزیابی نیازها برای سیستم‌های توضیح‌پذیر کمک می‌کند. این مدل، یک چارچوب مرجع از عوامل اصلی و نکات مرتبط را در اختیار توسعه‌دهنده می‌گذارد که هنگام تعریف و استخراج توضیح‌پذیری از تحلیل نیازمندی‌ها تا مرحله طراحی (عملیاتی سازی نیازمندی‌های استخراج شده) و ارزیابی (اندازه‌گیری این که آیا نیازمندی‌ها برآورده شده‌اند) باید لحاظ شود. نام‌های مختلفی را به مدل‌های مرجع اختصاص دادند مانند نام‌های زیر:

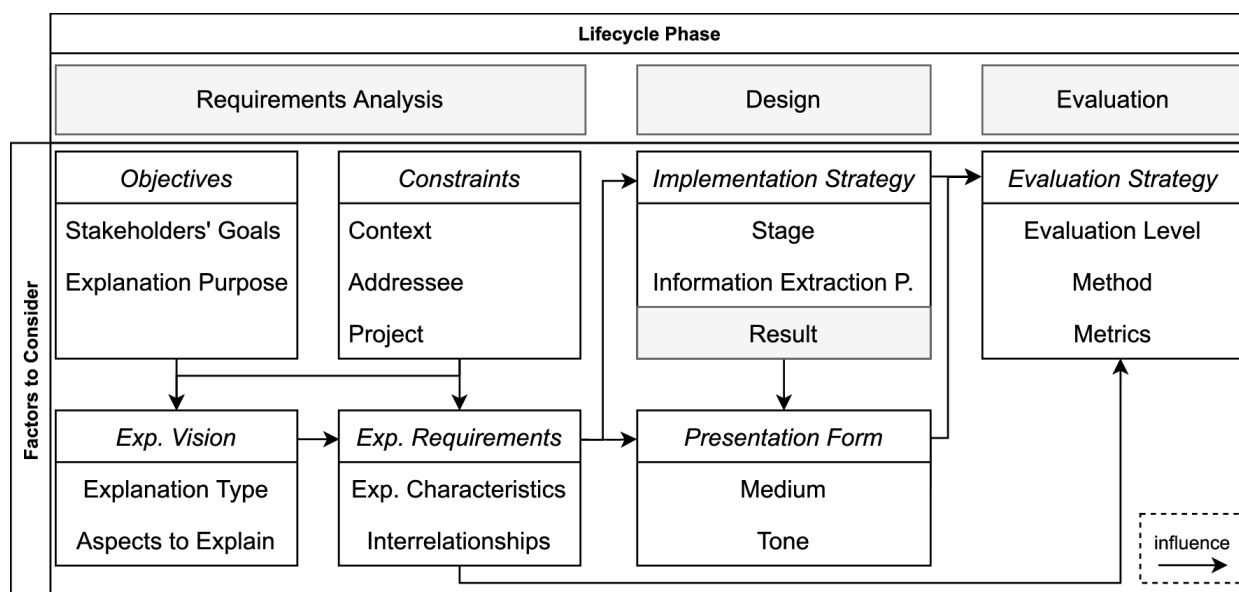
• Universal models

• Generic models

• Model patterns

برای نمونه، جهت درک بهتر پروتکل‌های شبکه و نحوه ارسال و دریافت داده‌ها، مدلی هفت لایه نام OSI^{۱۳} را معرفی کردند که در بالاترین سطح تجرید می‌توان تعاریف هر لایه به همراه وظایف آن‌ها را به بهترین شکل آموزش داد. مدل مرجع OSI به وسیله مهندسان شبکه برای توصیف معماری‌های شبکه مورد استفاده قرار می‌گیرد تا بتوانند متناسب با پروتکل‌ها برنامه‌های مورد نیاز خود را توسعه دهند.

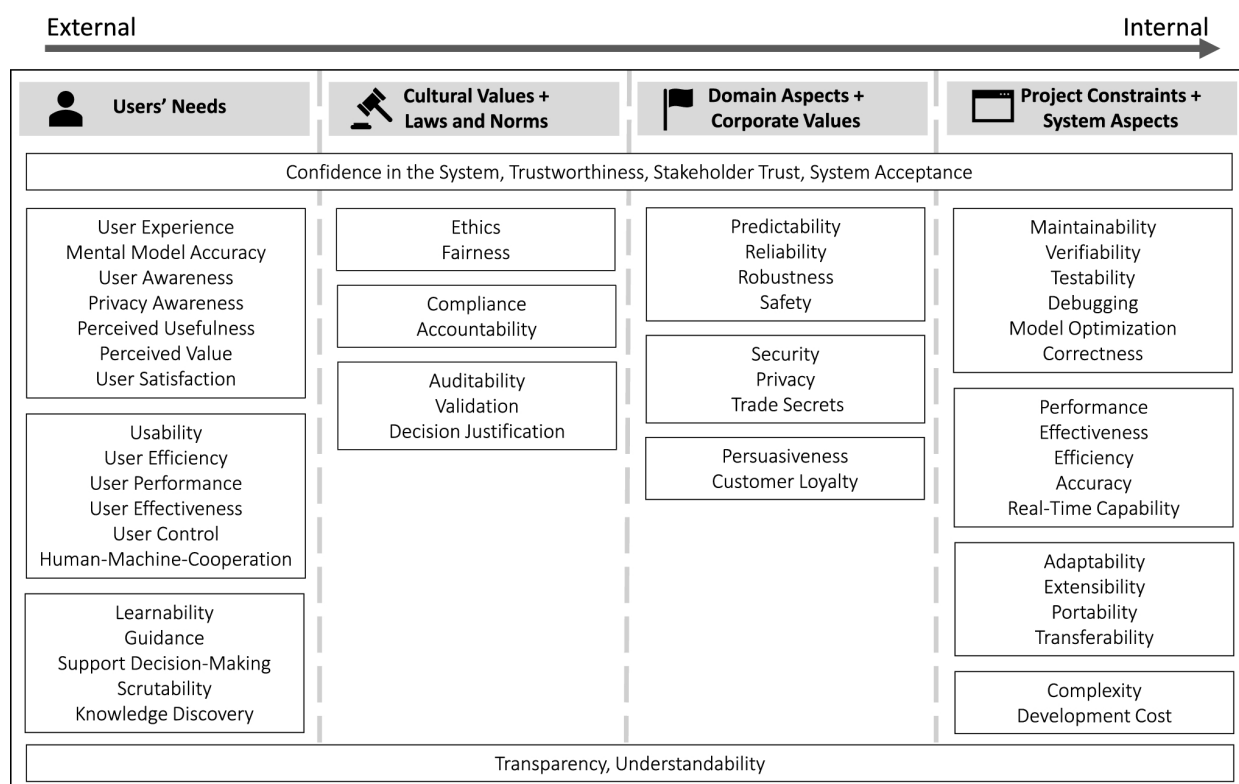
^{۱۳} Open System Interconnection



شکل ۱: مدل مرجع برای پشتیبانی از توسعه سیستم‌های توضیح‌پذیر

مهم‌ترین دلیل استفاده از مدل‌های مرجع را در زیر به دو شکل بیان کرده‌ایم:

۱. یک چهارچوب یا نمونه‌ای با سطح بالای تجرید، جهت درک کامل روابط میان موجودیت‌ها در یک محیط یا دامنه است (مانند شبکه‌های کامپیوتری یا سیستم‌های تبیین‌پذیر).
۲. جهت استانداردسازی یا توصیف فرایندهای توسعه. ذات این نوع سطح از تجرید به مهندسان انعطاف‌پذیری را اعطا می‌کند که می‌توانند در موقعیت‌های مختلف به راحتی سازگار شوند.



شکل ۲: یک مدل مفهومی که تاثیر توضیح‌پذیری را در ابعاد کیفی مختلف نشان می‌دهد.

۳.۳ راهنمای شناختی یا Catalogues

کاتالوگ دانش مجموعه‌ای سازمان‌دهی شده از منابع دانشی است که درون یک سازمان وجود دارد. این منابع می‌توانند شامل انواع دانش‌ها مانند اسناد، گزارش‌ها، روش‌های توسعه و بهترین رویکردهای حل مسئله، مواد آموزشی و موارد دیگر باشند. هدف اصلی از کاتالوگ‌ها تسهیل در توسعه، به اشتراک‌گذاری و استفاده مجدد از منابع دانش در یک سازمان در پروژه‌های مشابه می‌باشد. بعضی از محققان کاتالوگی را برای دامنه مشخصی مبتنی بر فرضیه چهارچوب‌های NFR توسعه داده‌اند به گونه‌ای که نتیجه این توسعه می‌تواند به تریدآف "چگونه یک یا چند NFR در یک سیستم رابطه و تعامل دارند و چگونه می‌توانند با یکدیگر همزستی داشته باشند" بپردازد. از نمونه‌های این کاتالوگ‌ها می‌توان به موارد زیر اشاره کرد:

- Serrano and Serrano یک کاتالوگ به طور خاص برای دامنه محاسباتی فراگیر و سیار ایجاد کرده‌اند [۷].
 - Torres and Martins پیشنهاد استفاده از کاتالوگ‌های NFR را در ساخت برنامه‌های میان‌افزاری RFID برای کاهش چالش‌های استخراج داده NFR در سیستم‌های مستقل، را مطرح کرده‌اند [۸].
- تمام این نمونه‌ها سعی در این داشتند که کاتالوگ‌ها ابزاری برای کاهش و حذف خطاهای احتمالی در شناسایی FRها و NFRها باشند.

۴ بررسی رسالت مقاله

به طور کلی این مقاله از مرور ادبیات سیستماتیک^{۱۴} استفاده کرده است. تعریف توضیح پذیری که از این قسمت به دست می آید، نقطه شروع ایجاد یک مدل مهندسی است که تأثیر توضیح پذیری را در کنار سایر ابعاد کیفیت نشان می دهد که مورد پذیرش و درک مشترک باشد. سپس فرآورده های مختلفی همچون راهنمای شناختی، مدل مفهومی و مدل مرجع را بر اساس این هسته مفهومی استخراج و تدوین می کند:

۱.۴ سوال های پژوهشی

- RQ1: تعریف مناسب از توضیح پذیری برای رسیدن به فهم مشترک در مهندسی نیازمندی ها و مهندسی نرم افزار چیست؟
- RQ2: حوزه های متأثر از توضیح پذیری در پس زمینه سیستمی چیست؟ چه حوزه های کیفی با توجه به زمینه سیستم (دنیای مسأله) از توضیح پذیری متأثر می شود؟
- RQ3: چگونه توضیح پذیری بر سایر حوزه های کیفی تأثیر می گذارد؟
- RQ4: چگونه می توان به متخصصان نرم افزار کمک کرد تا بتوانند فاکتورهای حائز اهمیت را در تحلیل، عملیاتی کردن و ارزیابی نیازمندی ها برای سیستم های توضیح پذیر مشخص کرد.

راه حل

راه حل اصلی مرور ادبیات سیستماتیک است. برای ارزیابی و تکمیل یافته ها، از یک روش کیفی دیگر نیز استفاده شده است: دو کارگاه با متخصصان برگزار شد.

۲.۴ استراتژی جست و جو در SLR

۱.۲.۴ جست و جوی دستی

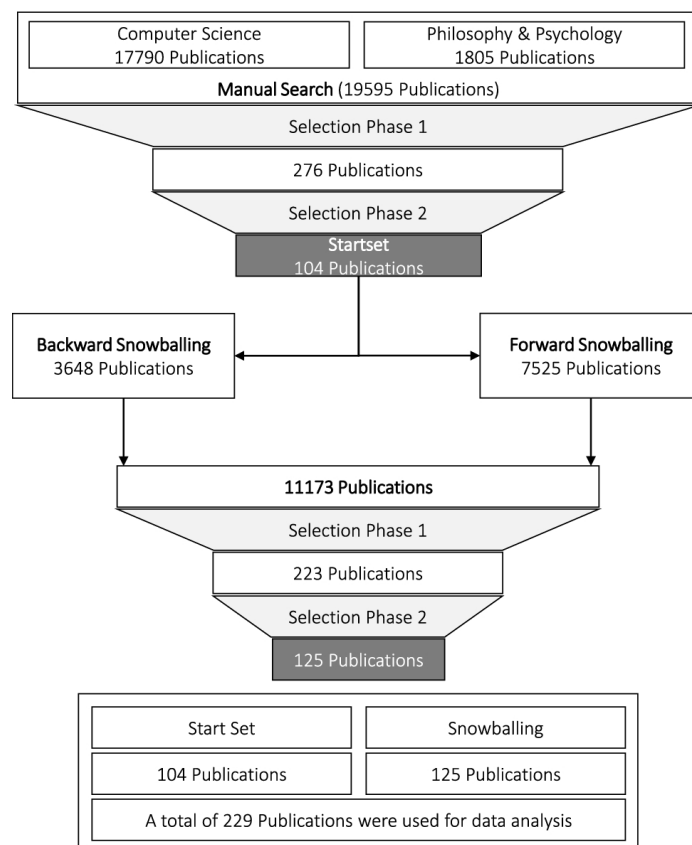
در این پژوهش ۲۲۹ مقاله ۳۶ سال اخیر مورد بررسی قرار گرفته است. این مقالات بین رشته ای و مرتبط با مهندسی نیازمندی ها می باشند. مقدار $k = 87.0$ نشان دهنده توافق کامل، روی این مقالات بر اساس جست و جو انجام شده می باشد. از open coding برای آنالیز داده ها جهت تعریف توضیح پذیری و رابطه آن با سایر جنبه های کیفی، استفاده کرده است.

۲.۲.۴ جست و جوی گلوله برفی برای جمع و تکمیل نتایج جست و جو

گردآوری نتایج به صورت پیش رونده و پس رونده بود. این روش بر اساس تئوری GT^{۱۵} است. یک مرور ادبیات هیچگاه کامل نمی شود، بلکه به اشباع می رسد. این اشباع زمانی اتفاق می افتد که هیچ مفهوم یا دسته بندی جدیدی از داده ها به دست نیاید. از این اصل برای پایان دادن به روش گلوله برفی استفاده شده است. یک تکرار بیشتر در مقالات نداشته اند. در تکرار دوم به هیچ دیدگاه یا مفهوم جدیدی نرسیده اند.

^{۱۴} Systematic Literature Review (SLR)

^{۱۵} Grounded Theory



شکل ۳: بررسی ساختار SLR انجام شده در این مقاله

معیارهایی که محققان بر اساس آن‌ها مجموعه‌ای از مقالات را در نظر گرفتند:

۱. اطلاعاتی که به صورت کامل یا بخشی از سوالات پژوهشی محققان را در بر می‌گرفت.
۲. مقالات در بازه زمانی ژانویه ۱۹۸۴ تا مارس ۲۰۲۰ بوده است. (۳۶ سال) سال شروع، اولین باری بوده که کار جدی روی توضیح‌پذیری انجام شده است. سابقه ۳۶ سال، نشان‌دهنده قدمت و اهمیت موضوع توضیح‌پذیری است.
۳. مقالات شامل پژوهشی، مروری، کنفرانسی و کارگاهی بوده است.

معیارهایی که بر اساس آن‌ها مقاله‌ها را رد کردند:

۱. مقالات غیر انگلیسی زبان
 ۲. مقالاتی که از تکنیک‌های الگوریتمی استفاده کرده بودند اما درباره پس‌زمینه نظری توضیح‌پذیری بحثی نکرده بودند.
- انتخاب مقاله یک فرایند دو فازی داشته است:
۱. انتخاب مقاله بر مبنای عنوان، چکیده و کلمات کلیدی (در این فاز مقالاتی که از الگوریتم استفاده می‌کردند، قابل شناسایی نبودند).
 ۲. انتخاب مقاله بر مبنای کل متن

اعتبارسنجی داده‌های جمع‌آوری شده با دو کارگاه بوده که در آن متخصصان و خبره‌های رشته‌های مزبور حضور داشتند. هدف این کارگاه‌ها این بوده که بررسی کنند آیا تمام داده‌هایی که نسبت به دامنه جمع‌آوری شده، درست هستند یا نه؟ نتیجه این بررسی باعث تولید یک دانشی شده که در قالب مدل مفهومی به صورت شکل در آمده است. این دانش در ابتدا استخراج شده و سپس تبدیل به اطلاعات ارزشمندی در قالب راهنمای دانش آماده‌سازی شده است.

۳.۴ نتایج: فرآورده‌های پژوهشی

در مدل مفهومی و کاتالوگ دانش، ارزیابی ذهنی در دسته‌بندی و طبقه‌بندی جنبه‌های کیفی به ابعاد مختلف، دخالت داشت:

۴.۴ راه‌حل این فرآورده‌ها

۱. بررسی داخلی: ریشه‌های این طبقه‌بندی بر مفاهیم معتبر موجود در ادبیات قرار داده شده است. این طبقه‌بندی، با کارشناسان مقایسه و بحث شد.
 ۲. بررسی خارجی: کارگاه‌ها با کارشناسان برگزار شد. در این مسیر نیز نتایج ادبیات با دانش کارشناسی مقایسه شد.
- کدگذاری و دسته‌بندی برای مدل مرجع نیز به صورت ارزیابی ذهنی بوده است. برای کاهش این تهدید، میزان انحراف پژوهشگر را با انجام تحلیل (کدگذاری و دسته‌بندی) به صورت مستقل کاهش دادیم. چون هدف از برگزاری کارگاه‌ها اعتبارسنجی یافته‌ها نبود، از نتایج بر پایه کارهای قبلی برای دسته‌بندی مدل مرجع استفاده شد. با این حال، درستی و واقع‌گرایی مدل مرجع ارزیابی نشده و نیاز به بررسی‌های بیشتری برای اعتبارسنجی این مدل وجود دارد.

۵.۴ جنبه‌هایی که در این طرح باید توضیح داده شوند

- عملکرد سیستم به طور کلی
- فرآیندهای استدلال سیستم
- منطق درونی سیستم
- درونی بودن مدل سیستم
- قصد و مفهوم سیستم (نتیجه و عملکرد مورد انتظار)
- رفتار سیستم و تعامل آن با جهان خارجی
- تصمیم‌ها و معیارهایی که در تعیین راهبردها و عملکرد سیستم مورد استفاده قرار گرفته است.
- شاخص‌های اندازه‌گیری و ارزیابی عملکرد سیستم
- دانش سیستم درباره کاربران و جهان خارجی (مطالعه ترجیحات کاربران، دانش سیستم درباره جهان و عوامل محیطی مرتبط).

۶.۴ نتیجه راهنمای شناختی (دانشی)

در این راهنما دانشی، تأثیر توضیح‌پذیری از دید متخصصان (شرکت‌کنندگان در کارگاه‌ها) و نیز با توجه به مقاله‌های مورد بررسی (روش SLR) بر ۵۷ نیازمندی غیر عملیاتی، به شرح جدول، فهرست شده است.

Quality Aspect	Literature	Expert	Quality Aspect	Literature	Expert	Quality Aspect	Literature	Expert
Accountability	+	+	Knowledge Discovery	+	+	Support Decision Making	+	+
Accuracy	+ -	+	Learnability	+	+	System Acceptance	+	+
Adaptability		-	Maintainability		+ -	Testability	+	
Auditability	+	+	Mental Model Accuracy	+	+	Trade Secrets	-	-
Complexity		-	Model Optimization	+	+	Transferability	+	
Compliance	+	+	Perceived Usefulness	+	+	Transparency	+	+
Confidence in the System	+ -	+ -	Perceived Value	+	+	Trustworthiness	+	+
Correctness	+	+	Performance	+ -	-	Understandability	+ -	+
Customer Loyalty	+	+	Persuasiveness	+	+	Usability	+ -	+ -
Debugging	+	+	Portability		+ -	User Awareness	+	+
Decision Justification	+	+	Predictability	+		User Control	+	+
Development Cost	-	-	Privacy	+ -	-	User Effectiveness	+ -	+
Effectiveness	+		Privacy Awareness	+		User Efficiency	+ -	
Efficiency	-		Real-Time Capability		-	User Experience	+ -	+ -
Ethics	+	+	Reliability	+	+	User Performance	+	
Extensibility		-	Robustness	+	+	User Satisfaction	+	
Fairness	+	+	Safety	+	+ -	Stakeholder Trust	+ -	+
Guidance	+	+	Scrutability	+		Validation	+	+
Human-Machine Cooperation	+	+	Security	+ -	-	Verifiability	+	+

+ positively influenced by explainability - negatively influenced by explainability

شکل ۴: راهنمای دانشی در جهت توضیح‌پذیری و تاثیر آن در جنبه‌های کیفی دیگر

۷.۴ تجزیه و تحلیل نیازمندی‌ها

تجزیه و تحلیل نیازمندی‌های جمع‌آوری شده به منظور درک و مستندسازی آن‌ها می‌باشد. مواردی که باید درباره آن‌ها اطلاعات جمع‌آوری و تحلیل کرد، عبارت است از: زمینه استفاده، جنبه‌های سیستم توضیح‌پذیر و مصرف‌کننده توضیح (سیستم نهایی)، ابعاد کیفی دیگر به دو صورت روی چگونگی شناسایی توضیح‌پذیری در سیستم شناسایی، تاثیر دارند؛

۱. چه اهدافی با توضیح‌پذیری مشارکت دارند؟^{۱۶}

۲. محدودیت‌هایی که در سیستم مزبور وجود دارد^{۱۷}.

اهداف و محدودیت‌ها به عنوان پایه‌ای برای تصمیمات سطح بالاتر، مانند تعریف یک دیدگاه توضیح‌پذیری، عمل می‌کنند. دیدگاه توضیح‌پذیری تعریفی سطح بالا از توضیح‌پذیری است و شامل ملاحظات مانند نوع توضیحی که ارائه می‌شود و جنبه‌های مهم برای توضیح است. علاوه بر این، تمام این ملاحظات (اهداف، محدودیت‌ها، دیدگاه توضیح‌پذیری) در نهایت در نیازمندی‌های توضیح‌پذیری جمع‌آوری می‌شوند.

۸.۴ اهداف

شامل اهداف نهایی ذینفعان و دلایل نیاز به توضیح‌پذیری در یک سیستم است. این خواسته‌ها، دلایل و اهداف، متعاقباً به وسیله هر یک از ابعاد کیفیتی که قبلاً ذکر شد (مانند ارزش‌های فرهنگی، جنبه‌های دامنه و...) تحت تاثیر قرار می‌گیرند. توضیح‌پذیری می‌تواند ابزار دستیابی به جنبه‌های دیگر کیفی باشد که باید به آن جنبه تصریح شود. ممکن است در یک پروژه خاص ذینفعان بخواهند جنبه‌های دیگر کیفیت مانند بهبود تجربه کاربری یا نیاز به توسعه یک سیستم در هماهنگی با ارزش‌های اخلاقی از طریق توضیح‌پذیری تحقق یابد. به طور کلی، همه یا هر کدام از ۵۷ جنبه کیفیتی که در مدل شکل شماره ۴ لیست شده‌اند، در هر ترکیبی می‌توانند به عنوان یک هدف عمل کنند.

^{۱۶} مانند شفافیت بیشتر یا قابلیت استفاده در یک سیستم

^{۱۷} مانند زمان‌بندی پروژه یا محدودیت‌های مالی

۹.۴ محدودیت‌ها

محدودیت‌ها عواملی هستند که به طور مستقیم یا غیرمستقیم تصمیم‌گیری‌های طراحی را تحت تأثیر قرار می‌دهند یا محدود می‌کنند. محدودیت‌ها با عواملی مانند زمینه^{۱۸} گیرنده و شرایط پروژه مرتبط هستند. این عوامل تأثیر قابل توجهی بر طراحی سیستم می‌گذارند. همه این تأثیرات باید در نظر گرفته شوند تا یک دیدگاه از سیستمی که قرار است توسعه یابد تعریف شود و آن را به نیازمندی‌ها تبدیل کند. محدودیت‌ها باعث اعمال و شکل گرفتن تصمیمات طراحی خاص می‌شوند.

۱۰.۴ استراتژی پیاده‌سازی

استراتژی پیاده‌سازی به این معنا است که چگونه توضیحات در سیستم پیاده‌سازی می‌شود. این بخش شامل توابع، ماژول‌ها (از نظر راه‌حل‌های الگوریتمی) و عناصر رابط کاربری است که باید در سیستم پیاده‌سازی شود تا توضیحات ارائه شود.

- توابع می‌توانند بسته به آنچه باید توضیح داده شود و پیچیدگی مدل زیربنایی، از نظر پیچیدگی متفاوت باشند.
- ماژول‌ها به عنوان بخش مجزا از سیستم و یک موجودیت جداگانه عمل می‌کنند. (به عنوان مثال، یک دستیار مجازی).
- در نهایت، عوامل رابط کاربری بیشتر مربوط به انتخاب‌های طراحی مربوط به این است که توضیحات چگونه در رابط کاربری ارائه خواهد شد.

۱۱.۴ بررسی و دو مرحله برای پیاده‌سازی توضیح‌پذیری

سیستم‌های توضیح‌پذیر را می‌توان در دو مرحله بررسی کرد:

۱. مرحله پس‌ازمان: توضیح سیستم فعلی، پس از طراحی و پیاده‌سازی
۲. مرحله پیش از زمان: قبل از طراحی و پیاده‌سازی سیستم، توضیح‌پذیری را در آن و نیازمندی‌های آن اعمال می‌کنیم و سپس سیستم مورد نظر را توسعه می‌دهیم.

۵ روال استخراج اطلاعات

این که چگونه اطلاعات مورد نیاز برای ایجاد توضیحات استخراج می‌شود. استخراج اطلاعات برای سیستم‌های مبتنی بر هوش مصنوعی اغلب باید توسط یک ماژول اضافی انجام می‌شود.

در سیستم‌های سنتی این سوال مطرح است که آیا دسترسی به کد سیستم برای به دست آوردن اطلاعات مورد نیاز برای توضیح آن ضروری است؟ با دسترسی کد می‌توان آن را تحلیل کرده، به الگوریتم رسید و اطلاعات داخلی را که برای ساخت توضیحات استفاده می‌شود، به دست آورد. در واقعیت، دسترسی به کد سیستم ضروری نیست، چون می‌توان اطلاعات مورد نیاز برای ایجاد توضیحات را از طریق تجزیه و تحلیل‌های خارجی جمع‌آوری کرد. برای این کار می‌توان از روش اختلال محلی استفاده کرد. در این روش، ورودی را دست کاری می‌کنیم تا چگونگی تغییرات خروجی را ببینیم.

۱.۵ نتیجه استخراج اطلاعات

اطلاعات استخراج شده، معمولاً دارای یک معنای خاص است که به آن نتیجه می‌گویند. انتخاب دقیق معنای یک توضیح ممکن است حیاتی باشد، به توجه به اینکه چه کسی توضیح را دریافت می‌کند و در چه زمینه‌ای تولید می‌شود. این اتفاق ممکن است رخ دهد زیرا معنای یک توضیح معمولاً کسانی را که می‌توانند آن را درک کنند، تحت تأثیر قرار می‌دهد.

انواع مختلف نتایج:

^{۱۸} زمینه، تعامل فرد (برای مثال پزشک)، سیستم (سیستم تشخیص سرطان توضیح‌پذیر)، وظیفه (پشتیبانی از تفسیر آزمایش‌های تصویری تومورهای سرطانی) و محیط (بیمارستان) است.

۱. وابستگی ویژگی: اطلاعات استخراج شده می‌تواند به وابستگی ورودی و خروجی در یک ویژگی اشاره کند.

۲. مثال‌ها: ممکن است مثال‌های نماینده از تصمیمات مشابه باشند.

۳. پیوندهای به توضیح: به عنوان مثال، روشن کردن یک ویژگی خاص می‌تواند برای پیاده‌سازی توضیحات متضاد استفاده شود و توضیحات علیتی می‌توانند با ارائه مثال‌ها منتقل شوند.

۲.۵ فرم ارائه

توضیحات می‌توانند به صورت‌های مختلف متنی، عددی، تصویری و شنیداری باشند. هرکدام از این فرمت‌ها دارای زیر-کلاس‌های دیگری است. به عنوان مثال، فرمت‌های متنی می‌توانند به زبان طبیعی و یا به صورت قوانین (مثلاً دستورات if-else) باشند. توضیحات باید با استراتژی‌های طراحی خاص رابط کاربری مانند توضیحات شنیداری، آیکون‌ها و انیمیشن‌ها (مثلاً برای برجسته کردن رویدادی جدید مانند یک رخداد) در صفحه‌نمایش کامپیوتر باشند و به طور احتمالی با متن‌های مختصر ارائه شوند. علاوه بر این، لحن ارتباطی باید غیررسمی باشد و نه رسمی.

۳.۵ استفاده از ماژول توضیحات

می‌توان برای ارائه توضیحات در سیستم‌های مبتنی بر هوش مصنوعی، یک ماژول به سیستم موجود اضافه کرد. این روش از نظر مرحله پسا زمان خواهد بود. وقتی می‌توان از ماژول توضیحات استفاده کرد که سیستم مورد نظر، به وسیله شبکه عصبی عمیق (DNN) پشتیبانی شود. بنابراین، روش استخراج اطلاعات ماژول باید با DNN ها سازگار باشد

۶ ارزیابی

ارتباطات پلی بین اهداف کلی مشتری به یک سو و معیارهای موافقت شده برای اندازه‌گیری به سوی دیگر می‌سازند. به همین ترتیب، برای تحلیل تأثیر توضیحات، روش‌ها و معیارهای ارزیابی باید تعریف شوند. به طور خاص، معیارها باید کمک کنند تا متوجه شویم آیا راه‌حل‌های فنی انتخابی به تأمین نیازمندی‌های تعریف‌شده کمک می‌کنند؟ همچنین به ارزیابی نیز کمک می‌کنند؟ آیا اجرای انتخاب شده برای قابلیت توضیح مناسب است؟ یا نیاز به بهبود دارد؟

۱.۶ سطوح ارزیابی برای توضیح‌پذیری

حداقل دو سطح ارزیابی برای توضیح‌پذیری می‌توان در نظر گرفت:

۱.۱.۶ ارزیابی در سطح سیستم

توضیح‌پذیری جنبه فعال‌سازی دارد و با آن می‌توان به NFRهای دیگر دست پیدا کرد. پس می‌توانیم توضیح‌پذیری را با اندازه‌گیری میزان مشارکت آن در دستیابی به جنبه‌های کیفیت دیگر ارزیابی کنیم. مثلاً می‌توان از مشارکت توضیح‌پذیری و استفاده‌پذیری^{۱۹} و یا عملکرد، به ارزیابی آن پرداخت.

۲.۱.۶ ارزیابی در سطح توضیح

می‌توانیم توضیح‌پذیری را با واریسی توضیحات تولید شده نیز ارزیابی کنیم. روش‌های مختلف برای ارزیابی توضیحات سیستم‌های نرم‌افزاری در سطح توضیح پیشنهاد شده است. با این حال، این روش‌ها استاندارد ندارد و انتخاب روش‌های ارزیابی و معیارهای آن به اهداف از پیش تعیین شده، وابسته است.

^{۱۹} به عنوان مثال کلمه Usability تأثیر توضیحات بر استفاده‌پذیری سیستم می‌تواند به صورت کمی (از طریق نمرات آزمون‌های استفاده‌پذیری) یا به صورت کیفی (از طریق ارزیابی درک کاربران از استفاده‌پذیری سیستم) ارزیابی شود.

۲.۶ روش‌های ارزیابی

محققان در این مقاله روش‌هایی را برای ارزیابی شناسایی کرده‌اند عبارتند از:

- مطالعات کاربری به طور کلی
- پرسشنامه‌ها
- آزمون‌های A/B
- مطالعات موردی
- مصاحبه‌ها

۳.۶ بررسی بازخورد کاربران نهایی

بررسی بازخوردهای کاربران نهایی: مهم‌ترین روش‌های استفاده شده برای ارزیابی توضیحات، مطالعات کاربری هستند. ارزیابی‌های ذهنی، روش متداول‌تری برای ارزیابی هستند. فعالیت‌ها و روش‌های متمرکز بر روی کاربر، اغلب برای توسعه سیستم‌های توضیح پذیر توصیه می‌شوند. در این تحقیق، محققان در SLR خود یافته‌اند که مهم‌ترین روش‌های ارزیابی بر روی بازخوردهای کاربران نهایی تمرکز دارد.

۴.۶ پرسشنامه

در پرسشنامه‌ها، شرکت‌کنندگان مطالعه ممکن است سوال شوند که آیا یک جنبه خاص از سیستم را پس از دریافت توضیحات بهتر درک کرده‌اند یا خیر. به عنوان مثال: "از توضیحات، من نحوه عملکرد [نرم‌افزار، الگوریتم، ابزار] را می‌فهمم."

۵.۶ آزمون‌های A/B

در این روش، دو یا چند نسخه یا نوع مختلف از یک عنصر (مثلاً توضیحات) به صورت تصادفی به گروه‌های مختلف از کاربران ارائه می‌شود. یک گروه (گروه A) نسخه اصلی یا قدیمی را دریافت می‌کند و دیگر گروه (گروه B) نسخه جدید یا تغییر یافته را دریافت می‌کند. سپس به هر گروه پرسیده می‌شود که کدام نسخه را ترجیح می‌دهند یا چگونگی تاثیر آن را ارزیابی می‌کنند. این روش به تحلیل تاثیر تغییرات و انجام مقایسه بین دو یا چند شرایط یا نسخه مختلف برای انتخاب بهترین گزینه برای مدیریت و بهبود سیستم کمک می‌کند.

۶.۶ مطالعات موردی

یک روش تحقیق تجربی که بر روی مطالعه یک مورد یا پدیده خاص (مانند یک فرد، یک گروه، یک محیط، یا یک سازمان) در محیط یا متن خود تمرکز دارد. در مطالعات موردی از ترکیب چندین تکنیک جمع‌آوری داده استفاده می‌کنند تا به درک بهتر پیچیدگی موارد فردی کمک کند و دقت داده‌ها و استنتاج‌های حاصل را افزایش دهد. یک مطالعه موردی ممکن است شامل ارزیابی سیستم‌های توضیح‌پذیر در یک محیط خاص و استخراج اطلاعات دقیق در مورد تاثیر آنها باشد.

۷.۶ مصاحبه

یک روش تحقیق کیفی است که در آن جمعیت مورد نظر (یعنی افرادی که مورد مصاحبه قرار می‌گیرند) توسط پژوهشگر پرسش می‌شوند.

۷ معیارها

ابزارهای سنجش متنوعی برای توضیح‌پذیری وجود دارد. سازگاری، پذیرفته‌شدن، واقع‌گرایی و متقاعدسازی می‌توانند برای سنجش توضیحات به کار روند. قابلیت فهم، ارتباط، طول، به‌موقع بودن، کامل بودن، و سودمندی نیز از معیارهایی است که در پژوهش‌های پیشین بسیار مطرح شده است. به‌موقع بودن توضیح، یک محدودیت است. یک توضیح وقتی کمک‌کننده یا مرتبط است که نه تنها جامع باشد، بلکه در لحظه مناسب بیان شود تا در تصمیم‌گیری کمک کند. مثلاً در هنگام مسیریابی، توضیحی که دلایل تغییر مسیر را شرح می‌دهد باید به موقع فهمیده شود تا مسافر بتواند به موقع تصمیم بگیرد. در این سناریو، حد فاصل زمانی که کاربر توضیحات را دریافت می‌کند تا زمان انجام اقدام، می‌تواند یک معیار در نظر گرفته شود. مثلاً این معیار می‌تواند برای ارزیابی این استفاده شود که آیا کاربر توضیح تغییر مسیر را در زمان مناسب به دست آورده یا آیا هنوز فاصله‌هایی در انتقال اطلاعات وجود دارد. سرانجام، یک مطالعه کاربری می‌تواند به ارزیابی نگرش شرکت‌کنندگان نسبت به تجربه کاربری کمک کند.

۸ کلیات و مباحث

اولین کاری که در این پژوهش مورد بررسی قرار گرفت، تعریف توضیح‌پذیری برای مهندسان نرم‌افزار است. این تعریف مشخص می‌کند هنگام مدیریت نیازمندی‌ها و عملکرد مناسب برای سیستم‌های توضیح‌پذیر چه چیزی باید در نظر گرفته شود:

- جنبه‌هایی که باید توضیح داده شوند.

- زمینه‌ها، توضیح‌دهندگان و مخاطبان

علاوه بر این، آگاهی از این متغیرها فرآیند توسعه نرم‌افزار را تسهیل و حمایت می‌کند و نیازمندی‌های توضیح‌پذیری را مشخص می‌کند. از آن جا که توضیح‌پذیری به عنوان یک جنبه ارتباطی بین سیستم‌ها و انسان‌ها دیده می‌شود، بسته به اینکه چگونه در عمل اتفاق می‌افتد، می‌تواند روابط را تقویت یا آن‌ها را آسیب بزند. از طرف دیگر، انتخاب‌های طراحی نادرست در خصوص توضیح‌پذیری می‌تواند مشکلات زیر را به وجود آورد:

۱. ارائه اطلاعات ناکافی یا انتخاب ارائه نادرست: بر روابط با کاربر مثلاً مشکلات تجربه کاربری تأثیرات منفی بگذارد.

۲. تأثیر نابجا در جنبه‌های کیفی ضروری برای شرکت‌ها: مثلاً برای تصویر برند و وفاداری مشتری آسیب جدی داشته باشد.

۳. ایجاد تأثیرات منفی برای پروژه یا سیستم: مثلاً هزینه‌های توسعه را افزایش دهد یا موجودیت سیستم را مختل کند.

۱۰.۸ محدودیت‌ها و تهدید اعتبار پژوهش

اساس این پژوهش، تحلیل داده‌های کیفی است. بنابراین، احتمال وابستگی نتایج به زمان تحلیل و احتمال وجود اختلافات زیادی در نتایج وجود دارد. بنابراین، از یک رویکرد چند روشی استفاده شده تا نتایج ارائه شده از جامعیت و اعتبار بالاتری برخوردار باشند. این روش‌ها عبارتند از:

۱۰.۱.۸ انجام یک مرور سیستماتیک و فرآیند کدگذاری

در این قسمت معیارهایی وجود دارد که می‌تواند پیش از شروع مرور ادبیات، برای رسیدن به یک سطح مناسب از فهم مشترک تدوین شده باشد تا تمام تهدیدهای اعتبار این پژوهش را کاهش دهد:

۱. معیارهای اضافه و کاهش برای جلوگیری از تصمیمات ذهنی موضوعی در فرآیند انتخاب.

۲. معیار دوره انتشار، که قابل ارزیابی است.

۳. معیارهای وابسته به محتوای مقالات که به موضوع هستند.

۴. برای کاهش میزان تعصب پژوهشگر، تحلیل به صورت مستقل انجام شد.

۵. در هر دو مرحله بررسی ادبیات و فرآیند کدگذاری، در صورت عدم توافق، تصمیم در مورد اضافه یا حذف (برای یک مقاله) یا اختصاص کد (برای داده‌های استخراج شده) به صورت جمعی تصمیم گرفته شده است.

یکی از محدودیت‌های دیگر مرور فرآیند پژوهش این است که تنها دوره زمانی تا مارس ۲۰۲۰ را پوشش می‌دهد. توضیح‌پذیری یک زمینه تحقیقاتی به سرعت در حال تکامل است، تضمینی برای به‌روز بودن نتایج محققان وجود ندارد. برای کاهش این مشکل، هنگام ساخت مدل مرجع از تحقیقات جدیدتر به عنوان راهنمایی استفاده شده است. در این زمینه، هیچ مشکلی برای به اجماع رسیدن نتایج محققان وجود نداشته است. علاوه بر این، دوره زمانی مرور پژوهش در حال حاضر شامل ۳۶ سال است، در طی آن مشاهده کردیم که مفاهیم مشابه ثابت در ادبیات باقی می‌مانند.

۲.۱.۸ کارگاه‌ها

برای کارگاه‌ها چندین تهدید به اعتبار وجود دارد:

۱. برخی از تکالیف داده شده به شرکت‌کنندگان بر اساس نتایج بررسی ادبیات پژوهش بوده است که ممکن است افراد را به طور ضمنی تحت تاثیر قرار دهد.

(آ) راه حل: تمام اطلاعات اضافی از تکالیف حذف شد. همچنین، کارشناسان می‌توانستند از تجربیات خود استفاده کنند و محدودیتی نداشتند.

۲. کارگاه‌ها به صورت آنلاین برگزار شد که می‌تواند یک عامل محدود کننده باشد. در اینجا، هدف از تکالیف این بوده است که کارشناسان را با موضوع آشنا کند تا کارگاه‌ها به طور متمرکز در میان فرمت آنلاین برگزار شوند.

(آ) راه حل: از مک‌های زیاد و وظایف واضح استفاده شد و این کمک کرد تا شرکت‌کنندگان در طول کارگاه بهره‌وری داشته باشند.

۳. زمان اختصاص داده شده برای هر کارگاه کوتاه بود. چهار ساعت برای هر کدام.

(آ) راه حل: فقط سه وظیفه برای هر کارگاه در نظر گرفته شد. محدود کردن کارها باعث شد کارگاهی لغو نشود (اشاره به کتاب کار آهسته کال نیوپورت [۴]).

۹ نتیجه‌گیری نهایی و ایده‌ها

توضیح‌پذیری یک وسیله مناسب برای دستیابی به جنبه‌های کیفی ضروری در یک سیستم، مانند شفافیت، پاسخگویی و اعتماد است. با جانمایی این جنبه‌ها در سیستم‌ها، به ابزارها و روش‌هایی برای تحلیل، پیاده‌سازی و اعتبارسنجی نیازمندی‌های مرتبط احساس نیاز می‌شود. به همین دلیل، باید موافقت راجع به درک توضیح‌پذیری به طور کلی داشته باشیم. معنی آن، اثرات آن، دسته‌بندی آن. علاوه بر این، تأثیر توضیح‌پذیری با جنبه‌های کیفی مهم‌تری مانند اخلاق، حریم خصوصی و اعتماد همچنین باید مورد تحقیق قرار گیرد.

تعریف توضیح‌پذیری ارائه شده در این پژوهش، به فرآیند ارتباط و انطباق انتظارات هنگام اشاره به توضیح‌پذیری کمک می‌کند. مدل مفهومی به متخصصان کمک می‌کند تا دسته‌بندی‌های توضیح‌پذیری را درک کنند. کاتالوگ دانش به شناسایی تفاوت‌ها بین توضیح‌پذیری و سایر کیفیت‌های مهم کمک می‌کند. مدل مرجع پیشنهادی برای توضیح‌پذیری به مهندسين نرم‌افزار کمک می‌کند تا جنبه‌های مرتبط و تاثیرگذار برای تجزیه و تحلیل نیازها، طراحی و ارزیابی سیستم‌های توضیح‌پذیر را درک کنند. به عنوان پیشنهاد پژوهش، با پشتیبانی از این آرتیفکت‌ها، می‌توان به استراتژی‌های طراحی و راه‌حل‌های سطح اجرا دست پیدا کرد که به نتایج مثبت برای همه ذینفعان منجر می‌شود.

- [1] Chazette, Larissa, Brunotte, Wasja, and Speith, Timo. Explainable software systems: from requirements analysis to system evaluation. *Requirements Engineering*, 27(4):457–487, 2022.
- [2] cmmiinstitutue. Official website.
- [3] for Geeks, Geeks. McCall’s quality model, 2024.
- [4] Newport, Cal. Slow productivity.
- [5] One, Quality. Quality function deployment (qfd).
- [6] SA, IEEE. Ieee standard for software quality assurance process. IEEE 730-2014.
- [7] Serrano, Milene and Serrano, Maurício. Ubiquitous, pervasive and mobile computing: A reusable-models-based non-functional catalogue. In *ER@ BR*. Citeseer, 2013.
- [8] Torres, Renato Cristiano and Martins, Luiz Eduardo Galvão. Nfr catalogues for rfid middleware. *Journal of Computer Science and Technology*, 14(02):102–108, 2014.
- [9] www.iso.org. Iso/iec 25010:2011.