فصلنامه علمی- پژوهشی
فنّاوری اطلاعات و ارتباطات ایران

سال دوم، شماره‌های۳و۴، بهار و تابستان ۱۳۸۹
صص: ۷۰- ۸۹

ایجاد نیمه خودکار مشاپ های سازمانی با استفاده از توصیفات معنایی

شهریار محمدی[*] **علی خلیلی**[**]

* استادیار، دانشکده مهندسی صنایع، دانشگاه خواجه نصیرالدین طوسی

** دانشجوی کارشناسی ارشد، دانشکده مهندسی صنایع، دانشگاه خواجه نصیرالدین طوسی

چکیده

مشاپ ها[1] ، نسل بعدی برنامه های کاربردی تحت وب هستند. یک مشاپ ، یک برنامه کاربردی مبتنی بر وب سبک وزن است که از ترکیب اطلاعات و یا قابلیت های دو یا چند منبع موجود، برای تحویل یک تجربه یکپارچه و جدید به کاربر بوجود می آید. مشاپ ها یک کلاس جدید از تکنیک های یکپارچه سازی را در محیط های سازمانی برای پیاده سازی برنامه های کاربردی موقعیتی معرفی می کنند (برنامه هایی که برای پاسخ به یک مساله فوری،گذرا و مشخص در سازمان توسعه داده می شوند). در یک محیط سازمانی پویا، پیچیده و رقابتی، پیش بینی و ایجاد همه برنامه های کاربردی ترکیبی که در آینده مورد استفاده قرار خواهند گرفت، امری غیر ممکن است. مشاپ های سازمانی به عنوان یک راه حل ساده و سریع، به افراد و تیم های کوچک در سازمان که دانش کمی در زمینه برنامه نویسی وب دارند، کمک می کنند تا با ترکیب و استفاده مجدد از منابع جاری سازمان و منابع منتشر شده روی اینترنت، برنامه های ترکیبی دلخواه خود را برای پاسخ به نیازهای زودگذرشان ایجاد کنند. در حال حاضر، ابزارهای ویرایشگر زیادی برای تسهیل فرایند ایجاد مشاپ های سازمانی ارائه شده اند. این ابزارها، با ایجاد یک واسط کاربری بصری، ایجاد مشاپ های وب را تا حدود زیادی آسان می سازند اما هنوز نیازمند این هستند که کاربر نهایی، تجربه هایی در زمینه تکنولوژی های وب، امنیت اطلاعات و ساختار داده ای مولفه های تشکیل دهنده مشاپ داشته باشد. علاوه بر این، مشاپ ایجاد شده توسط این ابزارها، وابستگی شدیدی به مولفه های تشکیل دهنده آن دارد. بنابر این ایجاد تغییر در ساختار مشاپ یا جایگزینی یک مولفه با مولفه دیگر، کاری پیچیده و زمانبر می باشد. این مساله در مشاپ های سازمانی که در آنها منابع داخل سازمان با منابع جهانی منتشر شده روی اینترنت، برای حل یک مساله موقعیتی، توسط یک کارگر دانش ترکیب می شوند، پررنگ تر می باشد. در این مقاله تلاش می شود تا با ترکیب سه تکنولوژی معماری سازمانی، وب معنایی و وب ۲,۰، راه حلی برای ایجاد نیمه خودکار مشاپ های سازمانی ارائه شود. علاوه بر این، در نظر داریم که یک مدل حاشیه گذاری معنایی برای سرویس های تشکیل دهنده مشاپ های سازمانی ارائه دهیم که توسط آن بتوانیم توصیفات معنایی لازم برای سرویس ها و نیز سیاست های امنیتی سازمان را در ایجاد مشاپ های سازمانی در نظر بگیریم. خروجی تحقیق، یک ابزار ویرایشگر مشاپ است که با پیاده سازی مدل پیشنهادی، فرایند ایجاد مشاپ های سازمانی را تسهیل می بخشد.

کلیدواژگان: مشاپ های سازمانی، معماری سرویس گرا، وب ۲,۰، وب معنایی، برنامه های کاربردی موقعیتی، برنامه های ترکیبی.

۱- مقدمه

در سال‌های اخیر، ظهور تکنولوژی مشاپ‌های وب که یکپارچه‌سازی موردی اطلاعات و سرویس ها را از چندین منبع مختلف امکان پذیر می سازد، روشی جدید را برای توسعه برنامه‌های کاربردی موقعیتی در سازمان‌ها به ارمغان آورده است.

1. Mashups

* نویسنده عهده‌دار مکاتبات (smohammadi40@yahoo.com)

Archive of SID
www.SID.ir

برنامه های کاربردی موقعیتی، برنامه هایی زودگذر و غیر قابل پیش بینی در داخل سازمان هستند که تعداد کاربران هدف محدود و با نیازهای مشخص دارند. شرکت های مطرح تحقیق کننده در بازار مانند فورستر[1] [۱] ، گارتنر[2] [۲] و یا اکونومیک اینتلیجنس یونیت[3][۳]، رشد سریع این الگوی توسعه را در سازمان ها در سالهای آینده پیش بینی کرده اند و آن را "مشاپ های سازمانی"[4] نامیده اند. مشاپ های سازمانی، کارگران دانش[5] را قادر می سازند تا وظایف معمولشان در سازمان را که عموما شامل دسترسی، تحلیل و یکپارچه سازی اطلاعات از منابع مختلف است، بطور موثرتری انجام دهند [۴]. علاوه بر این، مشاپ های سازمانی، با ترکیب منابع داخل سازمان با منابع جهانی منتشر شده روی اینترنت، دید جدیدی را در مورد موقعیتی که در یک حوزه مشخص رخ داده است ایجاد می کنند. مزیت اصلی مشاپ های سازمانی این است که توسط کاربر نهایی خود توسعه داده می شوند نه یک برنامه نویس حرفه ای وب. کاربر نهایی، یک آگاهی موقعیتی دارد و نزدیکترین شخص به مساله ای است که قرار است توسط مشاپ سازمانی حل شود. بنابر این راه حل آن را بهتر از سایرین می داند و می تواند با ایجاد یک راه حل سفارشی، رضایتمندی بیشتری را هم بر ای خود و هم برای سازمان بوجود بیاورد.

در حال حاضر، ابزارهای متعددی توسط شرکت های مختلف نرم افزاری، برای تسهیل فرایند ایجاد مشاپ های وب ارائه شده اند (مانند مایکروسافت پاپفلای[6]، [7]یاهو پایپس، آی بی ام کد ویکی[8]،گوگل مشاپ ادیتور[9]). این ابزارها، با ایجاد یک واسط کاربری بصری، ایجاد مشاپ های وب را تا حدود زیادی آسان می سازند اما هنوز نیازمند این هستند که کاربر نهایی، تجربه هایی در زمینه تکنولوژی های وب، امنیت اطلاعات و ساختار داده ای مولفه های تشکیل دهنده مشاپ داشته باشد. علاوه بر این، مشاپ ایجاد شده توسط این ابزارها، وابستگی شدیدی به مولفه های تشکیل دهنده آن دارد. بنابر این ایجاد تغییر در ساختار مشاپ یا جایگزینی یک مولفه با مولفه دیگر، کاری پیچیده و زمانبر می باشد.

1. Forester
2. Gartner
3. Economic Intelligence Unit
4. Enterprise Mashups
5. Knowledge Worker
6. http://www.popfly.com
7. http://pipes.yahoo.com/pipes/
8. http://services.alphaworks.ibm.com/qedwiki/
9. http://code.google.com/gme/

این مساله در مشاپ های سازمانی که در آنها منابع داخل سازمان با منابع جهانی منتشر شده روی اینترنت، برای حل یک مساله موقعیتی، توسط یک کارگر دانش ترکیب می شوند، پررنگ تر می باشد. زیرا:

- **زمان**، یک عامل مهم در سازمان بشمار می آید. مشاپ های سازمانی برای بهره گیری از فرصت های گذرای ایجاد شده در سازمان، توسعه داده می شوند و توسعه سریع آنها می تواند به سودآوری سازمان کمک کند. سادگی کاربری مشاپ و انعطاف پذیری ساختارمشاپ، می توانند تا حد زیادی، زمان توسعه آن را کاهش دهند. سادگی کاربری، این امکان را فراهم می آورد که کاربر نهایی با کمترین دانش برنامه نویسی وب، مشاپ های دلخواه خود را ایجاد کند. انعطاف پذیری ساختار مشاپ، امکان استفاده مجدد و مشارکت افراد مختلف در ایجاد مشاپ را فراهم می آورد.
- **امنیت**، یک عامل مهم در سازمان بشمار می آید. عدم آگاهی از مسائل امنیتی در هنگام ایجاد مشاپ سازمانی، ممکن است باعث فاش شدن اطلاعات حساس داخل سازمان شود و ریسک های زیادی را برای سازمان بوجود بیاورد.

یافتن راه حلی برای نیمه خودکار سازی فرایند ایجاد مشاپ های سازمانی[10] ، می تواند منجر به صرفه جویی در زمان و همچنین ایجاد مشاپ های ایمن تر با قابلیت استفاده مجدد بیشتر شود. این کار، راه را به سوی تمرکز روی نوآوری در ایجاد مشاپ های با ارزش افزوده به جای هدر دادن وقت در جزئیات پیاده سازی می گشاید. سوال اصلی این تحقیق، یافتن روشی است برای تسهیل فرایند ایجاد مشاپ های سازمانی.

ساختار این مقاله بدین صورت است که در بخش ۲، مفاهیم پایه مورد استفاده در تحقیق شرح داده می شوند. سپس در بخش ۳، به تعریف مشاپ های سازمانی پرداخته، کاربردهای آنها در سازمان، به همراه چالش های بکارگیری آنها مورد بحث قرار می گیرند. در بخش ۴، مدل پیشنهادی خود برای ایجاد نیمه خودکار مشاپ های سازمانی را ارائه می دهیم. در بخش ۵، به ارزیابی و امکان سنجی مدل پیشنهادی می پردازیم و در نهایت در بخش ۶ با جمع نتیجه گیری و شرح تحقیقات آتی، مقاله را به پایان می بریم.

10. Semi-automatic Creation of Enterprise Mashups

Archive of SID

www.SID.ir

۲- پیش زمینه

۲-۱ معماری سرویس گرا

معماری سرویس گرا، رهیافتی است برای ساخت سیستم های توزیع شده که کارکردهای نرم افزاری را در قالب سرویس ارائه می کنند. این سرویس ها هم توسط دیگر نرم افزارها قابل فراخوانی هستند و هم برای ساخت سرویس های جدید مورد استفاده قرار می گیرند، این رهیافت برای یکپارچه سازی فناوری ها در محیطی که انواع مختلفی از سکو های نرم افزاری و سخت افزاری وجود دارند ایده آل است.

معماری سرویس گرا، شیوه توسعه نرم افزار را از ساخت "محصول محور"[1] سنتی به ترکیب سرویس "مصرف کننده محور"[2] تغییر داده است [۶]. با پشتیبانی از تکنولوژی های ترکیب معماری سرویس گرا مانند BPEL [۷] و WSCI [۸]، فرایند کسب و کار جدید، کاربرد یا راه حل را می توان بطور نسبتا سریع و با هزینه کم، توسط ترکیب سرویس های توزیع شده در محیط های نا همگن ایجاد کرد. با این حال، ابزارها و زبانهای ترکیب موجود، عموما برای توسعه دهندگان حرفه ای طراحی شده اند تا مشکلات کسب و کار را در محیط فناوری اطلاعات پیچیده سازمان حل کنند.

در حال حاضر، سه مساله مهم در مورد تکنولوژی های ترکیب معماری سرویس گرا وجود دارند [۶]:

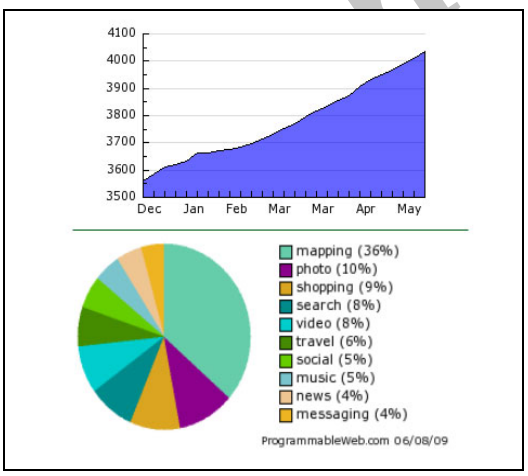
۱. این تکنولوژی ها شامل یک سربار[3] نیازمندی های نسبتا زیاد در رابطه با مهارت توسعه دهنده و زیر ساخت پشتیبان ، می باشند.
۲. این تکنولوژی ها نمی توانند از سفارشی سازی ترکیب بصورت پویا[4] و سریع پشتیبانی کنند.
۳. این تکنولوژی ها نمی توانند به خوبی از ترکیب برنامه های کاربردی جاری و سنتی روی وب که واسط سرویس ندارند، پشتیبانی کنند.

این مسائل، مانعی برای پذیرش سریع و گسترده معماری سرویس گرا شده اند. وب ۲.۰، با معرفی مشاپ های وب بر پایه مفاهیم ترکیب در معماری سرویس گرا ، راه حلی سریع ، سبک وزن و کاربر پسند را برای رفع این مسائل مطرح می کند.

۲-۲ وب ۲,۰

وب ۲,۰ یک برچسب است بر روی هزاران کاربردی که محتوای تولید شده توسط کاربر را استخراج و مورد استفاده مجدد قرار می دهند، از تعاملات مشارکتی و اجتماعی روی وب پشتیبانی می کنند، و از برهمکنش های جذاب برای کاربر[5] که مبتنی بر Ajax[6] هستند بهره می برند [۹]. این اصطلاح برای اولین بار در سال ۲۰۰۵ و توسط اورلی[7] در کنفرانسی به همین نام مطرح شد [۱۰]. وب ۲,۰ ، یک انقلاب کسب و کار در صنعت کامپیوتر بشمار می آید که در نتیجه حرکت به سمت اینترنت به عنوان یک سکو بوجود آمده است و تلاشی است برای شناخت قوانین موفقیت روی این سکو جدید. این ایده که کاربران در تولید محتوا نقش محوری را ایفا می کنند، جنبه اصلی وب ۲,۰ بشمار می آید [۱۱]. کاربردهای وب ۲,۰، کاربران غیرمتخصص را قادر ساخته است که در وب مشارکت داشته باشند و به تحقق وب قابل خواندن و نوشتن[8] کمک کرده است [۹].

مشاپ های وب، یکی از کاربرد های مهم وب ۲,۰ هستند که امروزه در بین افرادی که به ایجاد و فراهم آوردن سرویس های مفید روی وب علاقمند هستند، رواج یافته است. اصطلاح مشاپ[9] از موسیقی پاپ سر منشاء می گیرد و به عمل تولید یک آهنگ جدید (با سبک جدید) از ترکیب دو یا چند قطعه موسیقی موجود (عموما در سبک های مختلف) اطلاق می شود. در زمینه اینترنت، یک مشاپ به یک برنامه کاربردی تحت وب گفته می شود که داده های دو یا چند منبع را ترکیب کرده و بصورت یک ابزار یکپارچه ساده در می آورد [۱۲]. از این رو به مشاپ ها، برنامه های وب مرکب نیز گفته می شود[10].



شکل ۱: رشد مشاپ ها و دسته بندی انواع آنها

1. product-centric
2. consumer-centric
3. Overhead
4. On the fly
5. Engaging User Interactions
6. Asynchronous JavaScript and XML
7. O'Reilly
8. Read-Write Web
9. Mashup یا mash-up
10. Web Application Hybrid یا Composite Web Application

www.SID.ir
hive of SID

در حال حاضر، مشاپ ها به عنوان یک سکو قدرتمند برای توسعه برنامه های کاربردی تحت وب مطرح شده اند که چندین مجموعه جریان داده را بصورت یک تجربه کاربری یکپارچه ترکیب می کنند. وب سایت وب قابل برنامه ریزی[1] یک منبع مهم برای فهرست کردن مشاپ های تولید شده توسط کاربران و نشان دادن روند توسعه آنها روی اینترنت به شمار می آید. بر اساس گزارش این سایت، تعداد مشاپ های تولید شده روی اینترنت از مرز ۴۰۰۰ گذشته است و با نرخ رشد پایدار حدود ۳ مشاپ در روز، به گسترش خود ادامه می دهد. همانطور که در شکل ۱ نشان داده شده است، در حال حاضر بیشتر مشاپ های ایجاد شده توسط کاربران غیر تجاری هستند و حالت سرگرمی دارند. با این وجود، با بلوغ تکنولوژی پیش بینی شده است که مشاپ ها در سال ۲۰۱۰ به یک مدل غالب در بازار تبدیل خواهند شد [۱۳] و مشاپ های سازمانی در سال ۲۰۱۳ به حدود ۷۰۰ میلیون دلار از سهم بازار نرم افزار دست خواهند یافت(پیش بینی فورستر[2] [۱]).

توسعه بر مبنای قابلیت استفاده مجدد که در وب ۲,۰ مطرح شد، یک نوع اقتصاد جدید در برنامه های کاربردی وب بوجود آورد اما غیاب توصیفات داده ای معنایی در وب ۲,۰ ، محدودیت هایی را در ساخت این برنامه ها با پیچیدگی بیشتر بوجود آورده است. وب ۲,۰ از نظر داده ای بسیار غنی است اما از کمبود توصیفات معنایی و مشکل یکپارچه سازی در ابعاد گسترده رنج می برد [۱۴]. وب معنایی که به وب ۳,۰ نیز مشهور شده است، سعی دارد تا با فراهم آوردن یک زیرساخت معنایی بر روی وب ۲,۰، این مشکلات را برطرف سازد.

۲-۳ وب معنایی

در سالهای اخیر، وب معنایی با نام های وب داده ها[3] [۱۷] و یا یا وب ۳,۰ ، سعی در ورود به دنیای واقعی برنامه های کاربردی تحت وب را داشته است [۱۸]. وب معنایی با اینکه بسیاری از مشکلات یکپارچه سازی را حل می کند اما از کمبود کاربران و داده ها رنج می برد. همین مساله باعث پذیرش کند آن تا به امروز بوده است [۱۴]. این مساله توسط طبیعت کاربر محور وب ۲,۰ حل شده است. بنابر این ترکیب این دو تکنولوژی می تواند منجر به فراهم آمدن یک وب غنی داده ها شود که هم توسط انسان و هم توسط ماشین قابل بهره برداری است. غنی بودن در اینجا هم به غنی بودن داده ها اشاره دارد و هم به غنی بودن تعاملات کاربران.

۳- مشاپ های سازمانی

در ادبیات موضوع، تعاریف متعددی از مشاپ‌های سازمانی یا بطور کلی تر، مشاپ ها، ارائه شده است. در این مقاله، سعی شده است که علاوه بر در نظر گرفتن تعاریف علمی مطرح شده در کنفرانس ها و نشریات علمی بین‌المللی، تعاریف تجاری مطرح شده در صنعت نیز که جنبه کاربردی دارند، مورد بررسی قرار گیرند. با توجه به دیدگاه فنی [۶] و [۲۰] و [۲۱]، تجاری [۲۶-۲۲]، کاربردی [۲۷-۲۹]، توسعه دهندگان نرم افزار[۳۰-۳۲] و جامعه کاربران ([۳۳])، ۱۶ تعریف را مورد تحلیل قرار دادیم. نتیجه تحلیل این بود که تفاوت عمده ای میان تعاریف استفاده شده دانشگاهی و صنعتی وجود ندارد. در بیشتر تعاریف، به شیوه ترکیب منبع گرا، به عنوان خصوصیت اصلی مشاپ های سازمانی اشاره شده است. تفاوت اصلی تعاریف، روی نوع منابع ترکیب شده است. با مرور ادبیات موضوع و جمع بندی تعاریف موجود، تعریف زیر را به عنوان مبنای کار این مقاله ارائه می دهیم: "یک مشاپ سازمانی، یک سرویس وب است که منابع موجود در داخل سازمان را با منابع موجود در خارج سازمان، ترکیب می کند.این منابع می توانند محتوا، داده یا قابلیت نرم افزاری باشند که توسط کاربران نهایی سازمانی برای ایجاد برنامه های کاربردی اطلاعات محور یا برنامه های کاربردی موقعیتی با یکدیگر ترکیب می شوند."

۳-۱ کاربردهای مشاپ های سازمانی

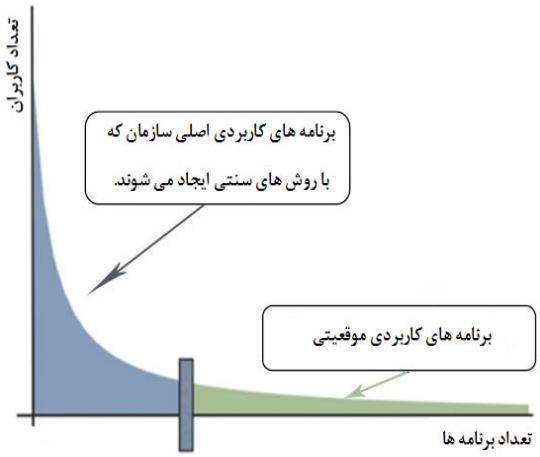
کاربرد اصلی مشاپ های سازمانی ، پیاده سازی برنامه های کاربردی موقعیتی در سازمان می باشد. این نوع از برنامه های کاربردی، عموما مشخصات زیر را دارند [۳۴]:

- کاربران هدف آنها محدود (تعداد کم) می باشد.
- گذرا هستند و طول عمر کوتاهی دارند.
- نیاز به یک راه حل آنی[4] و نسبتا خوب دارند.

1. ProgrammableWeb.com
2. Forrester
3. Web of Data
4. Just-in-time

Archive of SID
www.SID.ir

- توسط کاربران نهایی خود توسعه داده می شوند.



شکل ۳: جایگاه پروژه های موقعیتی در سازمان

برنامه‌های کاربردی موقعیتی، طیف گسترده ای از برنامه‌های کاربردی سازمان را در بر می‌گیرند که استراتژیک نبوده و در راستای ماموریت اصلی سازمان نیستند (شکل ۳). به این برنامه ها اصطلاحا دم دراز[1] برنامه های سازمان گفته می شود. جدول ۱ مقایسه ای را میان برنامه های کاربردی سنتی و برنامه های کاربردی موقعیتی در سازمان نشان می دهد. علاوه بر این، مشاپ های سازمانی این امکان را فراهم می آوردند که هوشمندی کسب و کار توسط فناوری اطلاعات پیاده سازی شود. همه اطلاعات مهم در مورد یک نقش را می توان در کنار هم و در یک صفحه نمایش داد. این کار، حس منطقی بیشتری را در مقابل داده های تنها بوجود می آورد. داده ها یا برنامه های داخل سازمان وقتی در کنار داده ها یا برنامه های دیگر قرار می گیرند، ممکن است حاوی اطلاعات بیشتری باشند. بنابر این ارزش تفسیر داده ها وقتی بصورت یکپارچه در می آیند بالاتر می رود [۳۵].

جدول ۱: برنامه های کاربردی سنتی در مقایسه با برنامه های کاربردی موقعیتی

	برنامه های کاربردی سنتی	برنامه های کاربردی موقعیتی
کاربران هدف	گسترده- یک گروه عمومی	محدود- یک فرد مشخص یا یک تیم کوچک
نحوه اداره و مدیریت	متمرکز و رسمی	غیر متمرکز و غیر رسمی
طول عمر	طولانی- چند ماه یا سال	کوتاه مدت- چند روز یا هفته
فازهای توسعه	به خوبی تعریف شده در راستای توافقات زمانی انجام شده	هیچ فاز یا زمانبندی مشخصی برای توسعه تعریف نمی شود و روی یک راه حل موردی و نسبتا خوب تمرکز دارد

۳-۲ چالش های مطرح در مشاپ های سازمانی

با اینکه مشاپ های سازمانی سرعت و انعطاف پذیری قابل توجه ای را در تحویل سرویس های ارزشمند برای کاربران سازمان به ارمغان می آورند، اما چالش ها و موانع موجود در آنها، سد راه بسیاری از شرکت ها برای بکارگیری آنها شده است. آگاهی از این چالش ها در بکارگیری آنها برای توسعه برنامه های موقعیتی در سازمانها بسیار مهم است. چالش های اصلی مطرح در مشاپ های سازمانی عبارتند از:

- **قابلیت پاسخگویی**

قابلیت پاسخگویی[2] بنا بر تعریف ارائه شده در [۳۶] عبارت است از "خاصیتی که توسط آن، ارتباط یک منبع واحد با یک شیء یا عمل، می تواند برای یک شخص ثالث اثبات گردد".

مساله قابلیت پاسخگویی در مشاپ ها بطور گسترده ای نادیده گرفته شده است و در حال حاضر ایجاد کننده مشاپ و دارنده اصلی سرویس یا محتوای منبع، هیچ تضمینی را در این زمینه به کاربران نمی دهند. این مساله وقتی حاد می شود که بخواهیم مشاپ های تراکنشی در محیط سازمانی ایجاد کنیم و سرویسی برای پرداخت در مشاپ وارد شود .

در [۳۷] تحقیق گسترده ای روی این مفهوم برای محیط مشاپ ها صورت گرفته است و قابلیت پاسخگویی در مشاپ ها ، نتیجه برقراری دو شرط زیر دانسته شده است:

الف. اطمینان[3] : احراز هویت[4] تمامی مولفه های شرکت کننده در سرویس مشاپ و توافق بر سر مساله قابلیت پاسخگویی.

ب. عدم انکار[5] : جبران خسارت با آشکار سازی کامل عملیات انجام شده توسط سرویس مشاپ.

1. Long tail
2. Accountability
3. Trust
5. Non-Repudiation

Archive of SID
www.SID.ir

راه حل پیشنهادی ما [۳۸] برای حل این مساله ،استفاده از PKI[1] به همراه سرویس های توصیفی منطقا سلسله مراتبی می باشد. بدین صورت که هر سرویس وب مشاپ که می خواهد قابلیت پاسخگویی داشته باشد بایستی یک سرویس وب عمومی پذیرفته شده (سرویس توصیفی یا Meta Web Service) را به سرویس های خود اضافه کند و اطلاعاتی را در مورد خود (از قبیل امضای دیجیتالی، جریان کار داخلی، اطلاعات مدیریتی مثل دقت، هزینه استفاده و غیره) به آن انتقال دهد. وظیفه این سرویس توصیفی، نظارت بر جریان داخلی مشاپ بدون دخالت در عملیات داخلی آن می باشد. در صورت بروز مشکل در مشاپ، از طریق این سرویس توصیفی می توان منشاء مشکل را شناسایی نموده و جبران خسارت کرد.

- **امنیت**

یکی از چالش های اصلی در سرویس های مشاپ، مساله امنیت است. مشاپ های سازمانی به دلایل زیر ،آسیب پذیری زیادی از نظر امنیتی دارند:

۱. غالبا توسط کاربران سازمانی ایجاد می شوند که دانش زیادی در زمینه امنیت ندارند.
۲. می توانند با دیگران به اشتراک گذاشته شوند که ممکن است خارج دیواره آتش[2] باشند.
۳. می توانند از ترکیب منابع مختلف و ناهمگن ایجاد شوند که ممکن است خارج دیواره آتش باشند و یا پروتکل های برقراری ارتباط متفاوتی داشته باشند.

به همین دلایل، بسیاری از سازمانها در بکارگیری مشاپ های سازمانی، بین امنیت و عملکرد یکی را بر می گزینند.

یکی از مزایای استفاده از مشاپ ها، سهولت استفاده از آنها می باشد. بدین منظور، اغلب مشاپ ها برای ایجاد یک واسط کاربری پویا و تعاملی از تکنولوژی های سمت مشتری[3] و بویژه تکنولوژی Ajax برای برقراری ارتباط بصورت غیر همگام میان سرویس دهنده و مشتری استفاده می کنند. بکارگیری تکنولوژی های سمت مشتری، از یکسو باعث افزایش سهولت کاربری می شود و از سوی دیگر آسیب پذیری امنیتی مشاپ ها را بالا می برد. در همه مرورگرهای معروف، سیاستی به نام هم منشاء بودن[4] پیاده سازی شده است که در واقع اساس امنیت وب می باشد. مرورگرهای وب، این سیاست را به عنوان یک مکانیزم محافظتی برای ایزوله کردن برنامه های کاربردی وب (که از دامنه های مختلف می آیند)، از یکدیگر بکار می برند [۳۹]. این سیاست در تضاد کامل با معماری مشاپ ها قرار دارد که قصد دارند سرویس ها و داده های مختلف را از دامنه های مختلف گرد آوری کرده و مورد استفاده قرار دهند. بدین ترتیب، بکارگیری یک راه حل ایمن برای ارتباط های بین دامنه ای[5] که امنیت سمت مشتری را نیز نقض نکند برای سرویس های مشاپ ضروری به نظر می رسد.

- **کیفیت و دقت داده ها**

از آنجا که در ایجاد مشاپ های سازمانی، کاربران نقش کلیدی را ایفا می کنند و حجم زیادی از اطلاعات بوجود آمده از منابعی است که این کاربران از داخل یا خارج سازمان تهیه کرده اند، دانستن منشاء اصلی داده ها و کیفیت و دقت آنها یکی از چالش های مهم در مشاپ های سازمانی بشمار می آید. محیط های سازمانی روی داده ها حساس هستند و اتکا روی داده های با کیفیت پایین یا ناصحیح، می تواند خسارات جبران ناپذیری را به سازمان تحمیل کند. ظهور سیستم های نگهدارنده سابقه[6] و نشانگرهای منشاء[7] در آینده نزدیک می توانند به شناخت بیشتر ارزش داده های مورد استفاده در مشاپ های سازمانی کمک کنند [۴۰].

۴- مدل پیشنهادی برای مشاپ های سازمانی معنایی

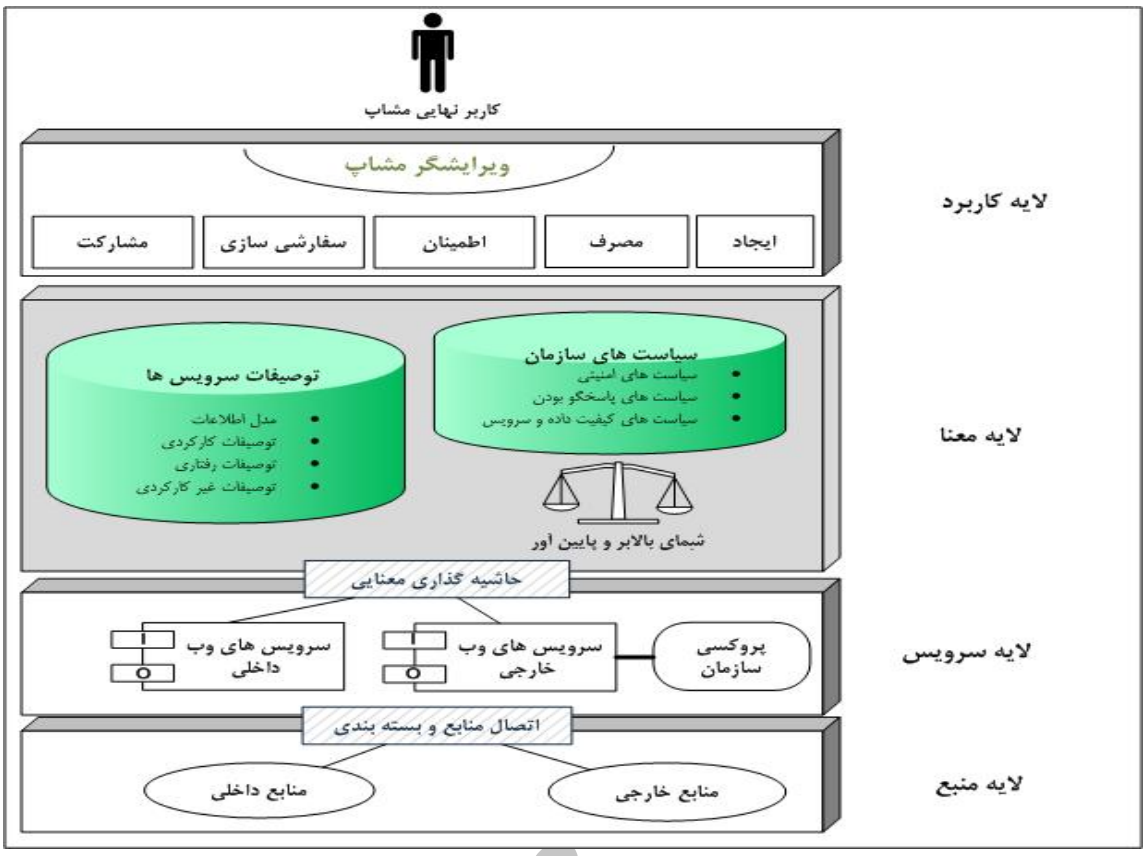
مدل پیشنهادی در این مقاله از یک معماری چهار لایه برای ایجاد نیمه خودکار مشاپ های سازمانی بهره می برد. همانطور که در شکل ۴ نشان داده شده است چهار لایه اصلی عبارتند از لایه منبع، لایه سرویس، لایه معنا و لایه کاربرد. ارتباط بین این لایه ها از طریق مولفه های اتصالی در نظر گرفته شده بین آنها صورت می گیرد. در ادامه به شرح دقیق هر لایه می پردازیم:

- **لایه منبع**

سبک ترکیب بر مبنای منابع، مشخصه اصلی مشاپ ها می باشد. در واقع، منابع، بلوک های اصلی سازنده مشاپ های سازمانی هستند [۴۱]. یک منبع می تواند محتوا، داده یا قابلیت کاربردی باشد. منابع می توانند در داخل یا خارج سازمان باشند.

1. Public Key Infrastructure
2. Firewall
3. Client-Side
4. Same-Origin Policy
5. Cross-Domain
6. Reputation System
7. Provenance Indicators

Archive of SID
www.SID.ir



شکل ۴: مدل پیشنهادی برای مشاپ های سازمانی معنایی

منابع می توانند از طریق سیستم های سازمانی مرکزی (مثل سیستم برنامه ریزی منابع[1]، سیستم مدیریت روابط مشتریان[2]، سیستم مدیریت زنجیره تامین[3] و سیستم مدیریت محتوای سازمان[4])، برنامه های کاربردی خط کسب و کار[5]، وب و یا حتی منابع اطلاعاتی شخصی(مثل صفحه گسترده ها[6]) تامین شوند [۴۲]. منابع مورد استفاده در مشاپ های سازمانی معمولا در فرمت نهایی XML هستند و به آنها feed گفته می شود.

- **لایه سرویس**

سرویس های وب، مولفه های نرم افزاری با اتصال کم هستند که روی وب منتشر و فراخوانی می شوند و از طریق پروتکل های استاندارد وب، قابل دسترسی هستند [۴۳]. سرویس های وب، یک مدل همکاری نو را پیشنهاد می کنند که مجزا از طبیعت ویژه هر پیاده سازی خاص است [۴۴]. در مدل پیشنهادی ما و در پس زمینه مشاپ های سازمانی، یک سرویس به عنوان یک لایه تجرد[7] بر روی منابع عمل می کند که آنها را غنی سازی می کند تا بتوان از آنها به عنوان دارایی های قابل ترکیب استفاده نمود. هر سرویس ، یک ورودی و یک خروجی دارد که امکان وصل کردن آنها به هم را فراهم می کند.

در حال حاضر، سه نوع معماری برای پیاده سازی سرویس های وب شرکت کننده در مشاپ های سازمانی موجود است:

- **معماری سبک RPC**

RPC[8] فراخوانی رویه از راه دور است. هدف RPC این است که به برنامه های کاربردی مختلف اجازه دهد تا مستقل از تفاوت ها و اهدافشان، از طریق شبکه و با استفاده از یک روش استاندارد با یکدیگر ارتباط برقرار کنند [۴۵]. هر سرویس وب RPC، یک پایانه[9] دارد که تعدادی عملیات روی آن قرار می گیرند که از طریق متد POST پروتکل HTTP قابل انجام می باشند. این نوع از سرویس ها برای تعامل با مشتریان از پروتکل های اضافی روی HTTP استفاده می کنند که آنها را سنگین وزن می سازد.

1. ERP
2. CRM
3. SCM
4. CMS
5. Line-of-business applications
6. Spreadsheets
7. Abstraction
8. Remote Procedure Call
9. Endpoint

Archive
www.SID.ir

پروتکل XML-RPC یک مثال روشن از معماری RPC است. این پروتکل از یک ساختار XML استاندارد برای درخواست ها و پاسخ ها از سرویس وب استفاده می کند. با ظهور پروتکل SOAP [۴۶]، این پروتکل دیگر کمتر مورد استفاده قرار می گیرد. پروتکل SOAP هم مانند XML-RPC یک پروتکل مبتنی بر XML است. با این حال، بسیاری از نواقص XML-RPC مانند فقدان نوع داده تعریف شده توسط کاربر ، پشتیبانی بهتر از مجموعه کاراکترها و امنیت بدوی آن را با استفاده از زبان توصیف سرویس(WSDL[1]) و داده های شمای XML (XSD[2]) (XSD[2]) برطرف می سازد [۴۵].

- **معماری سبک REST**

REST مخفف Representational State Transfer یا تبدیل حالت نمایشی می باشد. REST یک الگوی عملیات منابع است که به عنوان یک استاندارد غیر رسمی برای طراحی سرویس در برنامه های کاربردی وب ۲,۰ رواج پیدا کرده است. بر خلاف سبک سنتی RPC که از اشیاء راه دور، فراخوانی متد از راه دور و قابلیت کپسوله شده استفاده می کند، REST تنها با ساختار داده ها و وحالت آنها سروکار دارد. سادگی REST به همراه تطابق ذاتی آن با HTTP، آن را به انتخاب اصلی کاربران برای ایجاد مشاپ های وب تبدیل کرده است. سرویس های مبتنی بر REST که RESTful نیز نامیده می شوند از یک پایانه منحصربفرد برای هر منبع به همراه متدهای طبیعی HTTP بهره می برند و چون از هیچ پروتکل اضافی برای تعامل با مشتریان استفاده نمی کنند بسیار سبک وزن تر از سرویس های مبتنی بر RPC هستند.

اصل مرکزی طراحی مبتنی بر REST، مجموعه ای از عملیات تبدیل حالت است که در هر سیستم ذخیره و بازیابی اطلاعات وجود دارد. این عملیات عبارتند از : "ایجاد"، "خواندن"، "بهنگام سازی" و "حذف". به این مجموعه عملیات بطور مختصر CRUD[3] می گویند.

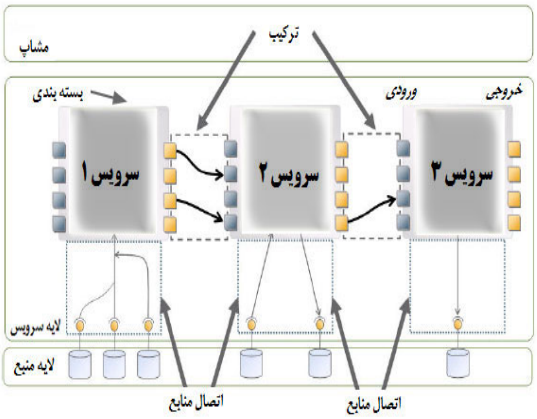
جامعه وب ۲,۰، یک نگاشت غیر رسمی از عملیات CRUD را روی دستورهای فراهم شده توسط HTTP یا همان متدهای HTTP انجام داده است که در آن، "ایجاد" معادل متد POST، "خواندن" معادل متد GET، "بهنگام سازی" معادل متد PUT و"حذف" معادل متد DELETE از پروتکل HTTP قرار داده شده است. این متدهای HTTP ، عملیات CRUD درخواست شده روی یک منبع را که توسط آدرس پایانه آن تعریف شده است، مشخص می کنند [۱۴].

- **معماری ترکیبی REST-RPC**

از دیدگاه طراحی، هیچ طراحی با قصد اولیه شروع به طراحی یک سرویس ترکیبی REST-RPC نمی کند. اما، عموما به علت دانش کم در مورد تئوری REST ، سرویس های وب زیادی با این نوع معماری در روی وب وجود دارد [۴۵]. این نوع از سرویس های وب جایی بین سرویس های مبتنی بر REST و سرویس های RPC قرار می گیرند. بیشتر این سرویس ها از متد GET پروتکل HTTP به همراه یک متد دیگر تعریف شده توسط کاربر که غیر از "خواندن" است استفاده می کنند.

از آنجا که در مدل پیشنهادی ما، ترکیب در لایه سرویس صورت می گیرد به مکانیزمی نیاز داریم تا منابع را بصورت سرویس مورد استفاده قرار دهیم. برای دستیابی به این هدف از مولفه "اتصال منابع و بسته بندی" بهره می گیریم. **اتصال منابع**[4]، عبارت است از مرتبط ساختن بیش از یک منبع به یکدیگر. **بسته بندی**[5]، قرار دادن یک واسط سرویس شامل داده ها و ساختارهای ورودی و خروجی روی منابع بهم متصل شده می باشد. بسته بندی، امکان ترکیب منابع در سطح سرویس را فراهم می آورد.

از آنجا که در این مقاله تمرکز ما روی سرویس های RESTful است، با اتصال منابع و بسته بندی آنها، قصد داریم که سرویس های RESTful روی منابع جاری سازمان ایجاد کنیم.



شکل ۵: اتصال منابع و بسته بندی

1. Web Services Descriptor Language
2. XML Schema Data
3. Create-Read-Update-Delete
4. Piping
5. Wrapping

Archive of SID

www.SID.ir

پروکسی سازمان، یک مولفه مهم دیگر در لایه سرویس است که مسئول اعمال سیاست های سازمان و اطمینان از اجرای ایمن سرویس های وب خارجی می باشد.

همانطور که در بخش ۳ اشاره شد، امنیت یک مساله مهم در مشاپ های سازمانی بشمار می آید. مشاپ های سازمانی منابع ناهمگن متعددی را که ممکن است خارج از دیواره آتش سازمان باشند با هم ترکیب می کنند و اغلب توسط کاربران نهایی سازمانی ایجاد می شوند که دانش کمی در زمینه امنیت دارند. این امر آنها را در مقابل حملات امنیتی، بسیار آسیب پذیر می کند. بویژه در مدل پیشنهادی ما که از یکپارچه سازی سمت مرورگر استفاده می کند، مشاپ تولید شده باید بطریقی مدل امنیتی هم منشا بودن را در مرورگر وب دور بزند تا بتواند اطلاعات سرویس شخص ثالث را بدست بیاورد. یک سرویس نا شناخته می تواند کد متخاصمی[1] را به برنامه کاربردی تزریق کند که منجر به حملات متعددی مثل XSS، CSRF و DoS شود و داده های سازمانی یا اطلاعات حساس کاربران را به سرقت ببرد [۴۷].

بنابر این تعاملات همه سرویس های وب خارجی باید از طریق پروکسی سازمان انجام شود که سرویس ها را از کدهای متخاصم پاک کرده و اجرای ایمن سرویس ها را تحت سیاست های سازمان تضمین می کند.

- **لایه معنا**

برای نیمه خودکار سازی فرایند ایجاد مشاپ های سازمانی به یک لایه شامل توصیفات معنایی بر روی لایه سرویس نیاز داریم. در حال حاضر چارچوب های[2] زیادی برای اضافه کردن توصیفات معنایی بر روی سرویس های وب وجود دارند. بیشتر این چارچوب ها برای مدل سازی سرویس ها از شیوه بالا به پایین[3] استفاده می کنند و برای سرویس های با سبک RPC و بویژه سرویس های مبتنی بر SOAP پیشنهاد شده اند (مانند WSMO [۴۸]، OWL-S [۴۹] و SWSF [۵۰]). این چارچوب ها برای مشاپ های وب که به شیوه پایین به بالا[4] ایجاد می شوند، مناسب نیستند [۵۱]. از طرف دیگر بر خلاف سرویس های مبتنی بر SOAP که یک سند WSDL برای تعریف عملیات خود دارند، سرویس های RESTful هیچ استانداردی را برای این کار در نظر نمی گیرند (WADL[5] توسط شرکت سان[6] به عنوان یک گزینه پیشنهاد شده است که نا به امروز مورد پذیرش واقع نشده است [۵۲]). علاوه بر این ها، از آنجا که عمومیت یافتن مشاپ های وب، بیشتر به دلیل استفاده از سرویس های وب RESTful می باشد، در مدل پیشنهادی خود، به چارچوبی نیاز داریم که از این نوع سرویس ها به خوبی پشتیبانی کند.

برای اضافه کردن توصیفات معنایی به سرویس های وب، به **حاشیه گذاری معنایی**[7] نیاز داریم. حاشیه گذاری معنایی مثل یک چسب بین لایه سرویس و لایه معنا عمل می کند. ما در مدل پیشنهادی خود از مدل حاشیه گذاری SAWSDL[8] [۵۳] برای حاشیه گذاری معنایی سرویس های مشارکت کننده در مشاپ سازمانی استفاده می کنیم. دلیل انتخاب SAWSDL ، سبک وزن بودن آن و مستقل بودن آن از هر تکنولوژی خاص معنایی می باشد (بدین معنی که هیچ نوع، فرم یا زبانی را برای توصیفات معنایی تعریف نمی کند [۵۱]). علاوه بر اینها، به علت اینکه افزونه هایی را روی وب سرویس های جاری اضافه می کند، برای مدلسازی پایین به بالای مشاپ های وب مناسب است.

SAWSDL یک لایه توسعه ساده تعریف می کند که با بهره گیری از سه خصوصیت توسعه دهنده، به مولفه های WSDL اجازه می دهد تا بصورت معنایی حاشیه گذاری شوند [۵۱] [۵۳]:

- modelReference برای اشاره به مفاهیم معنایی که یک مولفه WSDL را توصیف می کنند.
- liftingSchemaMapping و loweringSchemaMapping برای مشخص کردن نگاشت بین داده های XML و مدل اطلاعاتی معنایی.

مشابه SAWSDL ، پیشنهادهای زیادی برای حاشیه گذاری معنایی سرویس های وب RESTful وجود دارند. این پیشنهادها را می توان به دو دسته تقسیم کرد:

۱. پیشنهادهایی که یک فایل توصیفگر سرویس مانند WADL که قابل خواندن توسط ماشین باشد، در نظر می گیرند و تلاش می کنند که آن را به طریقی مشابه WSDL حاشیه گذاری کنند. به عنوان مثال می توان به "Semantic Bridge for Web Services" که توسط BBN Technologies[9] پیاده سازی شده است اشاره کرد.

1. Malicious Code
2. Framework
3. Top-Down
4. Bottom-Up

5. Web Application Description Language
6. Sun Microsystems
7. Semantic Annotation
8. Semantic Annotation for WSDL and XML Schema
9. http://www.bbn.com/technology/data_indexing_and_mining/asio_sbws

Archive of SID
www.SID.ir

۲. پیشنهادهایی که هیچ فایل توصیفگر قابل خواندن برای ماشین را برای سرویس در نظر نمی گیرند و تلاش می کنند که صفحات وب قابل فهم برای انسان ها را که توصیفگر سرویس هستند، با استفاده از Microformats[1] حاشیه گذاری کنند. به عنوان مثال می-توان به "SA-REST"[۵۴] و "HRESTS & MICROWSMO" [۵۵] که توسط مرکز Knoesis[2] پیاده سازی شده است اشاره کرد.

از آنجا که بیشتر سرویس های وب RESTful جاری، فایل توصیفگر قابل خواندن توسط ماشین ندارند و فقط شامل صفحات وب توصیفگر قابل فهم توسط انسان هستند، گزینه دوم عملی تر می باشد.

Microformat ها، مجموعه ای از فرمت های داده ای ساده و باز هستند که بر روی استانداردهای موجود پذیرفته شده روی وب ساخته می شوند. آنها اول برای فهم توسط انسان ساخته شده اند و سپس برای خوانده شدن توسط ماشین. Microformat ها سازگاری بهتری با طبیعت وب ۲,۰ دارند که کاربر انسانی را به عنوان محور در نظر می گیرد. از طرف دیگر، به علت این که قابل خواندن توسط ماشین نیز هستند می توان آنها را در وب ۳,۰ یا همان وب معنایی نیز جای داد.

پیشنهاد ما در این مقاله، چیزی مابین دو گزینه مطرح شده در بالا می باشد. روش پیشنهادی ما از یک عملیات توصیفگر در میان عملیات جاری سرویس بهره می گیرد. عملیات توصیفگر[3]، تصویر کلی سرویس وب و نیز عملیات های آن را به همراه جزئیات فراخوانی هر عملیات، توصیف می کند. عملیات توصیفگر از مدل داده ای RDF که در فرمت داده ای JSON[4] قرار گرفته است، استفاده استفاده می کند و دسترسی به داده ها از طریق متد GET پروتکل HTTP صورت می گیرد.

JSON [۵۶] که مخفف نشانه گذاری اشیاء در جاوااسکریپت[5] است، است، یک فرمت تبادل داده سبک وزن می باشد. نوشتن و خواندن JSON برای انسان ها راحت است. از سوی دیگر، تولید و تجزیه JSON برای ماشین نیز راحت است. با توجه به این ویژگی ها، ما از JSON برای توصیف سرویس های وب RESTful استفاده می کنیم. فاکتور دیگر انتخاب JSON طبیعت سمت مشتری بودن[6] آن است.

1. http://microformats.org/
2. http://knoesis.org
3. Descriptor Operation
4. JavaScript Object Notation
5. JavaScript
6. Client-based

از آنجا که JSON یک زیر مجموعه از زبان سمت مرورگر جاوااسکریپت است، می تواند به سرعت در سمت مشتری ارزیابی شود. این ویژگی در کارایی مدل پیشنهادی ما که از یکپارچه سازی سمت مرورگر استفاده می کند بسیار موثر است.

در مدل پیشنهادی خود، از دو روش برای قرار دادن RDF در JSON استفاده می کنیم. روش اول، استفاده از RDF/JSON [۵۷] برای مرتب کردن[7] مستقیم RDF در JSON می باشد. این روش مشابه استفاده از RDFa [۵۸] در "SA-REST" می باشد. در RDF/JSON یک سه تایی شامل فاعل(S)، مسند(P) و مفعول(O) به صورت زیر کد گذاری می شود:

{ "S" : { "P" : [O] } }

روش دوم، استفاده از یک فرمت JSON ساده به همراه یک مبدل برای تبدیل آن به RDF است. این روش برای انسان ها بسیار خواناتر است و مشابه hRESTS [۵۵] در "MicroWSMO " می باشد.

قرار دادن حاشیه گذاری معنایی به عنوان یک عملیات در سرویس مزایای زیادی را در پی دارد. این کار یک طبیعت پویا به توصیفات معنایی می بخشد. توصیفات معنایی می توانند به صورت پویا و بدون اینکه از قبل برای آنها تدارکی شده باشد، ایجاد شوند. همچنین با تغییر در عملیات سرویس، این توصیفات نیز بصورت پویا تغییر می کنند. علاوه براین، می توان برای دسترسی به این توصیفات از مکانیزم های امنیتی موجود در سرویس بهره برد.

جدای از مکانیزم حاشیه گذاری متفاوت ارائه شده در روش ما، مدل سرویس همانند مدل MicroWSMO می باشد که در جدول ۲ تعریف شده است. تنها تفاوت موجود، تعریف دو خصوصیت اضافی به نامهای "hasInputType" و "hasOutputType" می باشد. مقدار این خصوصیت ها می تواند ""، "SET" و "INDIVIDUAL" باشد.

اگر یک عملیات، مجموعه ای از منابع را به عنوان ورودی بگیرد یا به عنوان خروجی برگرداند، مقدار این خصوصیات "SET" می شود. اگر یک عملیات، تنها یک منبع را به عنوان ورودی بگیرد یا به عنوان خروجی برگرداند، مقدار این خصوصیات "INDIVIDUAL " می شود. اگر یک عملیات، هیچ ورودی یا خروجی نداشته باشد مقدار این خصوصیات "" می شود. این خصوصیات در مدل پیشنهادی ما،

7. Serialize

Archive of SID

www.SID.ir

جدول ۲: مدل سرویس بصورت RDF/N3

```
1 @prefix semcem: <http://www.semcem.com#> .
2 @prefix hr: <http://www.wsmo.org/ns/hrests#> .
3 @prefix wsl: <http://www.wsmo.org/ns/wsmo-lite#> .
4 @prefix sawsdl: <http://www.w3.org/ns/sawsdl#> .
5 @prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
6 @prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
7 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
8
9 # service model classes and properties
10 wsl:Service a rdfs:Class .
11 wsl:hasOperation a rdf:Property ;
12 rdfs:domain wsl:Service ;
13 rdfs:range wsl:Operation .
14 wsl:Operation a rdfs:Class .
15 wsl:hasInputMessage a rdf:Property ;
16 rdfs:domain wsl:Operation ;
17 rdfs:range wsl:Message .
18 wsl:hasOutputMessage a rdf:Property ;
19 rdfs:domain wsl:Operation ;
20 rdfs:range wsl:Message .
21 wsl:Message a rdfs:Class .
22
23 # hr properties
24 hr:as Address a rdf:Property ;
25 rdfs:domain wsl:Operation ;
26 rdfs:range hr:URITemplate .
27 hr:hasMethod a rdf:Property ;
28 rdfs:domain wsl:Operation ;
29 rdfs:range xsd:string .
30
31 # semcem properties
32 hr:hasInputType a rdf:Property ;
33 rdfs:domain wsl:Operation ;
34 rdfs:range xsd:string .
35 hr:hasOutputType a rdf:Property ;
36 rdfs:domain wsl:Operation ;
37 rdfs:range xsd:string .
38
39 # SAWSDL properties
40 sawsdl:modelReference a rdf:Property .
41 sawsdl:liftingSchemaMapping a rdf:Property .
42 sawsdl:loweringSchemaMapping a rdf:Property .
43
44 # a datatype for URI templates
45 hr:URITemplate a rdfs:Datatype .
```

الگوی فراخوانی عملیات را از سرویس های مختلف به هم وصل شده تعریف می کنند.

توصیفات معنایی در مدل پیشنهادی ما به سه دسته تقسیم می شوند:

- **توصیفات سرویس ها**

برای پشتیبانی از خودکارسازی مراحل مختلف مورد نیاز برای ایجاد مشاپ های سازمانی، باید چهار جنبه معنایی اصلی در سرویس را در نظر بگیریم:

الف. *مدل اطلاعات* (یک هستان شناسی دامنه[1]) که داده ها را نمایش می دهد (بویژه در پیغام های ورودی و خروجی).

1. Domain Ontology

ب. *توصیفات کارکردی* که بوسیله طبقه بندی کارکردها[2] یا پیش شرایط و اثرات[3]، مشخص می کند که سرویس چه کاری را انجام می دهد.

ج. *توصیفات رفتاری* که ترتیب فراخوانی عملیات در فراخوانی سرویس را تعریف می کنند.

د. *توصیفات غیر کارکردی* که سیاست های سرویس یا سایر جزئیات مربوط به پیاده سازی یا محیط احرایی سرویس را نشان می دهند.

راه حل های ارائه شده موجود مانند WSMO-Lite [۵۱] را می توان برای پوشش دادن این جنبه ها مورد استفاده قرار داد. WSMO-Lite، یک هستان شناسی سبک وزن را بر روی سرویس های وب حاشیه گذاری شده توسط مدل SAWSDL یا MicroWSMO فراهم می آورد.

- **شمای بالابر و پایین آور**

یکی از مهمترین موانع در ایجاد مشاپ های سازمانی، تنظیم کردن ورودی ها و خروجی های سرویس هایی است که با هم ترکیب می شوند. این عمل، به میانجیگری داده ها[4] معروف است. از آنجا که سرویس های وب RESTful هیچ راهی برای مشخص کردن فرمت و نوع داده های ورودی و خروجی سرویس ارائه نمی دهند، این عمل باید بصورت دستی توسط ایجاد کننده مشاپ انجام شود.

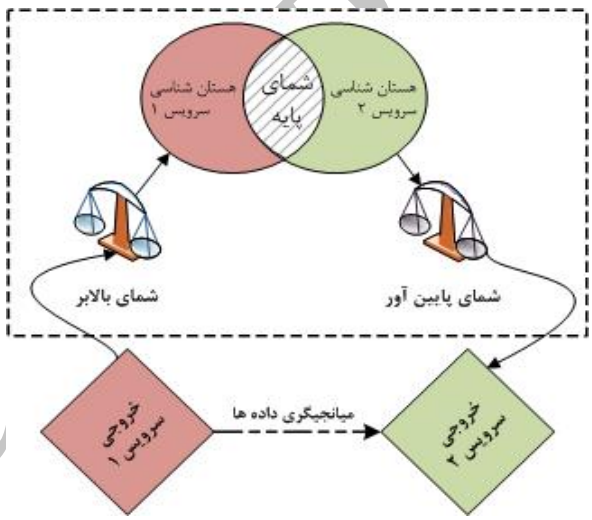
میانجیگری داده ها در SAWSDL وSA-REST توسط تعریف یک هستان شناسی برای نمایش پیغام های ورودی و خروجی و همچنین تعریف یک شمای بالابر و پایین آور[5] انجام می شود. کاری که شمای بالابر و پایین آور انجام می دهند، ترجمه ساختار داده های ورودی و خروجی سرویس به ساختار یک هستان شناسی مرجع به نام شمای پایه[6] می باشد.استفاده از این شمای پایه ضروری است زیرا مجبور کردن همه افراد روی وب به استفاده از یک ساختار داده مشخص در ورودی ها و خروجی های سرویسشان امری غیر ممکن است [۵۴].

از آنجا که ساختار داده ها در شمای پایه نسبت به خروجی نیمه ساختار یافته سرویس، در سطح تجرد[7] بالاتری قرار دارد، ترجمه از خروجی سرویس به شمای پایه را " بالابردن" و ترجمه از شمای

2. Functionality Classification
3. Preconditions and Effects
4. Data Mediation
5. Lifting and Lowering Schema
6. Grounding Schema
7. Abstraction

Archive of SID
www.SID.ir

پایه به ورودی سرویس را "پایین آوردن" می نامند. شماهای بالابر و پایین آور برای سرویس های وب در مدل پیشنهادی ما توسط XSLT [۵۹] یا XQuery [۶۰] نوشته می شوند. تکنولوژی XSLT[1] توسط کنسرسیوم جهانی وب با هدف اولیه تبدیل یک سند XML به نوع دیگر، طراحی شده است . XSLT ، دارای قابلیت های بمراتب بیشتری بمنظور تبدیل یک سند XML به HTML و سایر فرمت های مبتنی بر متن است و آن را زبانی بمنظور تبدیل ساختار یک سند XML در نظر می گیرند. XQuery هم همانند XSLT یک زبان پرس و جو[2] و تبدیل ساختار اسناد XML است.



شکل ۶: شماهای بالابر و پایین آور

در شکل ۶، نمایشی از شماهای پایین آور وبالابر، آورده شده است. حالت ایده آل وقتی است که هر دو سرویس از یک هستان شناسی واحد برای بالابردن و پایین آوردن استفاده می کنند (شمای پایه برابر). اما در دنیای واقعی، توافق روی یک هستان شناسی مرجع و بطور عمومی پذیرفته شده امری غیر ممکن است و توسعه هستان شناسی های مرجع، در حال پیشرفت می باشد. فراهم کننده های سرویس معمولا ترجیح می دهند که از هستان شناسی های خاص خودشان استفاده کنند. بنابر این، در مدل پیشنهادی خود به مکانیزمی برای یافتن شباهت ها میان دو هستان شناسی و تنظیم کردن[3] آنها نیاز داریم. حالت دیگری که ممکن است رخ دهد وقتی است که وقتی دو سرویس از دو زیر مجموعه مختلف (اما دارای اشتراک) از یک هستان شناسی واحد استفاده می کنند.

1. eXtensible Stylesheet Language Transformation
2. Query
3. Alignment

ما از یک الگوریتم ساده برای تطابق دو هستان شناسی استفاده می کنیم که دو modelReference را به عنوان ورودی می گیرد و درجه شباهت میان آنها را مشخص می کند. اگر دو هستان شناسی همسان نبودند، یک فایل XSLT تولید می کند که خروجی سرویس اول را بر مبنای ورودی سرویس دوم تنظیم می کند. این فایل بر روی سرویس دهنده مشاپ ذخیره می شود تا در مواجهات آینده مورد استفاده قرار بگیرد.

- **سیاست های سازمان ۲**

مشاپ های سازمانی ایجاد می شوند تا با ترکیب منابع داخل سازمان با منابع جهانی به حل یک مشکل موقعیتی بپردازند و برای سازمان سود آوری داشته باشند. اما این مساله مانند یک شمشیر دو لبه است! مشاپ های سازمانی ممکن است یک تهدید برای منابع داخل سازمان محسوب شوند اگر به درستی توسعه پیدا نکنند. همانطور که در فصل ۳ اشاره شد، چالش های زیادی در بکارگیری مشاپ های سازمانی وجود دارند.این چالش ها تا زمان بلوغ تکنولوژی مشاپ های وب ادامه خواهند داشت.

سیاست های سازمان بطور روشن تعریف می کنند که در ایجاد یک مشاپ سازمانی چه چیزهایی را می توان انجام داد و چه چیزهایی را نمی توان انجام داد در واقع، سیاست های سازمان، بایدها و نبایدها در ایجاد مشاپ سازمانی را برای تضمین اجرای ایمن و درست آن بیان می کنند. این سیاست ها را می توان به سه دسته تقسیم کرد:

الف. ***سیاست های امنیتی***: این سیاست ها به رویه ها و سیاست های زمان طراحی بر می گردد و عموما با تکنولوژی مورد استفاده یا ابزارهای امنیتی کاری ندارند.این سیاست ها سعی دارند تا با تعریف قوانین و رویه هایی، از تهدید های امنیتی جلوگیری کنند [۶۱]. برای ایجاد یک سیاست امنیتی، سازمان باید موارد کاربردی[4] کاربردی[4] ای را تعریف کند و امتحان کند که چگونه اطلاعات و رفتار سیستم های تحت تاثیر آن باید مورد محافظت قرار گیرند. بیشتر سازمان ها راهنماهایی را برای امنیت در نظر می گیرند مانند اینکه داده ها بدون کد گذاری نمی توانند به خارج از دیواره آتش سازمان منتقل شوند یا اینکه هر گونه استفاده از مشاپ باید به واحد فناوری اطلاعات گزارش داده شود و یا در بعضی موارد استفاده از برخی مشاپ ها اصلا اجازه داده نمی شود.

بطور کلی، سیاست های امنیتی، سطح امنیت مورد نیاز برای یک مشاپ مورد نظر را تعریف می کنند.

4. Use case

Archive

www.SID.ir

ب. ***سیاست های پاسخگو بودن***: قابلیت پاسخگویی[1] به این اشاره دارد که یک موجودیت، تعهدی برای اجرای یک اختیار و یا برآورده ساختن یک مسئولیت دارد. عدم انکار یک تراکنش توسط سرویس دهنده و سرویس گیرنده، جزئی از قابلیت پاسخگویی بشمار می آید. در یک سرویس مشاپ، مساله قابلیت پاسخگویی بسیار پیچیده است. زیرا ممکن است تعداد زیادی سرویس دهنده ضمنی وجود داشته باشند و محتوای ارائه شده توسط مشاپ لزوما توسط منشاء آن محتوا ارائه نشده باشد. علاوه بر این، محتوای منشا ممکن است در طول اجرای مشاپ تغییر کند [۳۷].

سیاست های پاسخگو بودن، سطح پاسخگویی مورد نیاز برای یک مشاپ مورد نظر را تعریف می کنند.

ج. ***سیاست های کیفیت داده و سرویس***: یکی از مشکلات قدیمی در تلاش های یکپارچه سازی سازمانی، دقت اطلاعات جمع آوری شده از منابع خارجی است. مشاپ های وب این مشکل را تشدید می کنند [۴۰]. سمت و سویی که وب در پیش گرفته است تا کاربران را به مشارکت در ایجاد داده ها تشویق کند باعث تقویت این مشکل شده است. اضافه کردن توصیفات معنایی به سرویس هایی وب می تواند غنای داده ها و سرویس های مورد استفاده در سرویس مشاپ را بالاتر ببرد.

سیاست های کیفیت داده و سرویس ، سطح دقت مورد نیاز برای یک مشاپ مورد نظر را تعریف می کنند.

برای تعریف سیاست های سازمان نیاز به یک چارچوب سیاست مانند Rein [۶۲] داریم. Rein یک چارچوب سیاست برای وب است که از تکنولوژی های وب معنایی استفاده می کند. Rein چندین هستان شناسی برای سناریوهای سیاستی مختلف به همراه مکانیزم هایی برای استدلال روی آنها فراهم می آورد که می توان آنها را برای کارکردن با زبان های سیاستی[2] مختلف و دانش دامنه ای[3] در RDF-S و OWL تنظیم کرد.

- **لایه کاربرد**

توصیفات معنایی، بسیاری از مراحل ایجاد یک مشاپ سازمانی را تسهیل می بخشند. در مدل پیشنهادی ما، یکپارچه سازی بطور منطقی در لایه معنا انجام می شود. این امر، خصوصیات متفاوت و ناهمگن سرویس های مختلف مورد نیاز برای ایجاد یک مشاپ سازمانی را از دید کاربر پنهان می سازد و در نتیجه ایجاد آن را تا حد زیادی خودکار می کند. در زمینه مشاپ های سازمانی، وظایف زیادی وجود دارند که می توان آنها را با استفاده از توصیفات معنایی خودکار کرد:

- اکتشاف: یافتن سرویس های وب مناسب بر اساس اهداف مورد نظر کاربر.
- ترکیب: اتصال چند سرویس وب به یکدیگر برای دستیابی به کارکرد درخواست شده توسط کاربر.
- میانجیگری داده ها: تنظیم ورودی ها و خروجی های سرویس های مختلف برای قابل ترکیب کردن آنها.
- میانجیگری فرآیند: حل مسائل مربوط به ناهمگن بودن فرایندهای عمومی درخواست کننده سرویس و سرویس وب.
- اداره کردن خطا ها: اداره کردن پیغام های خطا یا دیگر رویدادهای استثنایی[4] برگردانده شده توسط سرویس وب و تولید پیغام های خطای مرتبط در پاسخ به خطاهای زمان اجرا.

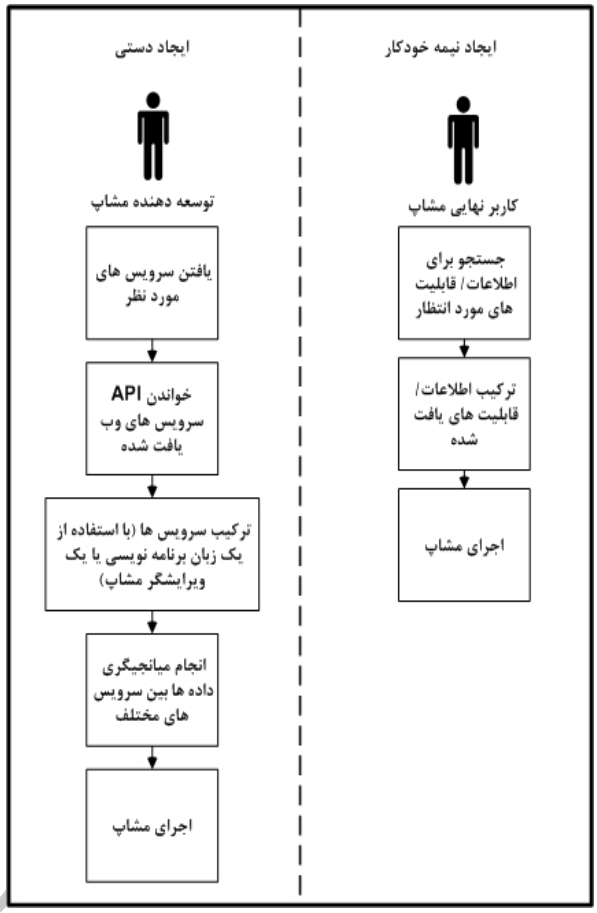
از دیدگاه مشتری و با توجه به تحقیقاتی که در بازار مشاپ های سازمانی صورت گرفته است، الگوهایی کلیدی برای اجرای موفق مشاپ های سازمانی استخراج شده اند. شرکت JackBe[5] که یکی از پیشتازان حوزه مشاپ های سازمانی است، ۵ الگوی اصلی را در این حوزه مشخص کرده است که به 5C مشاپ های سازمانی معروف شده اند:

۱. مصرف[6] : کاربران نهایی باید بتوانند سرویس ها و مشاپ های وب موجود را به راحتی مورد استفاده قرار دهند.
۲. ایجاد[7] : کاربران نهایی باید بتوانند به راحتی با ترکیب سرویس ها ها و مشاپ های وجود، مشاپ های سازمانی جدید ایجاد کنند.
۳. سفارشی سازی[8] : کاربران نهایی باید بتوانند به راحتی مشاپ های موجود را مطابق سلیقه ها و نیازهای خود سفارشی سازند.
۴. مشارکت[9] : کاربران نهایی باید بتوانند مشاپ های جدیدی را که ایجاد می کنند، به راحتی با دیگران به اشتراک بگذارند.
۵. اطمینان[10] : کاربران نهایی باید اطمینان داشته باشند که مصرف، ایجاد، سفارشی سازی و مشارکت، در یک محیط حفاظت شده و ایمن انجام می گیرند.

1. Accountability
2. Policy Language
3. Domain Knowledge
4. Exceptions
5. http://www.jackbe.com
6. Consume
7. Create
8. Customize
9. Collaborate
10. Confidence

Archive of SID

www.SID.ir



شکل ۷: ایجاد دستی در مقابل ایجاد نیمه خودکار مشاپ های

همانطور که در در شکل ۷ نشان داده شده است، در مدل پیشنهادی ما، کاربر نهایی (کارگر دانش) هم به عنوان توسعه دهنده و هم به عنوان مصرف کننده مشاپ عمل می کند. این کاربر، در مقایسه با سایر افراد سازمان، آگاهی بیشتری نسبت به محیط کاری دارد. در واقع، او نزدیکترین فرد به موقعیت است که مشکل را شناسایی کرده و به یافتن راه حل برای آن می پردازد. این کاربر نهایی که هیچ دانشی در زمینه برنامه نویسی ندارد می تواند با ایجاد مشاپ سازمانی از مزایای معماری سازمانی و وب ۲٫۰ بهره مند شود.

۵- ارزیابی و امکان سنجی مدل پیشنهادی

۵-۱ سناریوی کاربردی سازمانی

برای نشان دادن مدل پیشنهادی خود، از یک سناریوی کاربردی استفاده می کنیم که در آن مدیر بخش منابع انسانی[1] یک سازمان قصد دارد تا برای تسهیل استخدام و جذب نیرو در سازمان، یک مشاپ سازمانی را ایجاد کند. این مشاپ سازمانی، رزومه های آنلاین افراد روی اینترنت را بررسی کرده و افراد دارای شرایط لازم برای موقعیت های

خالی در سازمان را پیدا می کند. نتایج یافت شده، روی نقشه نمایش داده می شوند تا موقعیت جغرافیایی افراد در مقایسه با محل کار آنها به خوبی مشخص باشد.

مدیران بخش منابع انسانی در سازمان، با حجم وسیعی از رزومه ها روبرو هستند که مستقیما به سازمان فرستاده می شوند. با این حال، تعداد زیادی از کاندیداهای شغلی وجود دارند که رزومه خود را بصورت آنلاین و در وب سایت های مربوط به کاریابی آنلاین مانند Monster.com و یا شبکه های اجتماعی مانند Linkedin.com درج می کنند. ایجاد یک مشاپ که اطلاعات موقعیت های خالی در سازمان را با اطلاعات سرویس های رزومه خارجی ترکیب کند، بطور قابل توجهی عمل جذب نیرو در سازمان را تسهیل می بخشد و این فرصت را برای سازمان بوجود می آورد که نگاهی بلادرنگ به بازار افراد با استعداد برای جذب در سازمان داشته باشد.

کدهای مربوط به پیاده سازی سناریوی کاربردی طبق مدل پیشنهادی، از طریق آدرس وب http://ajaxian.ir/semcem/use%20case.htm قابل دسترس می باشد. برای ارجاع به آنها در این مقاله از سناریو – شماره جدول استفاده می شود.

۵-۲ پیاده سازی سناریوی کاربردی بر اساس مدل پیشنهادی

برای درک بهتر مدل پیشنهادی، سناریوی کاربردی تعریف شده را بر اساس مدل پیشنهادی پیاده سازی می کنیم:

- **لایه منبع**

در سناریوی مرجع تعریف شده، مدیر بخش منابع انسانی به عنوان مدیر مشاپ، به یک منبع داخلی دسترسی دارد که موقعیت های خالی در سازمان را تعریف می کند. این منبع می تواند یک خروجی XML از ماژول منابع انسانی سیستم ERP[2] سازمان باشد. در یک فرمت ساده، این منبع به صورت کد XML نشان داده شده در سناریو –جدول ۱ می باشد.

علاوه بر این منبع داخلی، مدیر بخش منابع انسانی، به دو منبع خارجی برای پیدا کردن روزمه های آنلاین و نمایش نتایج روی نقشه نیاز دارد که اولی یک سرویس وب منبع گرا است و دومی یک قابلیت کاربردی است که از طریق یک سرویس وب قابل دسترس است.

1. Human Resource

2. Enterprise Resource Planning

Archive

www.SID.ir

- **لایه سرویس**

در سناریوی مرجع تعریف شده، نیاز به انجام عمل "بسته بندی" داریم تا بتوانیم منبع موقعیت های خالی سازمان را بصورت یک سرویس وب RESTful مصرف کنیم. بنابر این سرویسی را طراحی می کنیم که فقط یک عملیات به نام list دارد که لیست موقعیت های خالی سازمان را به همراه اطلاعات مربوطه بر می گرداند. این عملیات، هیچ ورودی ای ندارد و خروجی XML نشان داده شده در سناریو –جدول ۱ را به عنوان خروجی بر می گرداند. این عملیات از طریق آدرس زیر قابل فراخوانی است:

http://www.semcem.com/positionOpening/list

علاوه بر این، مدیر مشاپ می تواند از سرویس نقشه Google برای نمایش نتایج روی نقشه استفاده کند. این سرویس بطور منطقی لیستی از مکان ها را دریافت کرده و یک کد جاوااسکریپت قابل اجرا را به عنوان خروجی بر می گرداند. این سرویس، یک سرویس نمایشی محض است و بیشتر عملیات تعامل با آن در سمت مشتری صورت می گیرد. بنابر این برای مصرف آن به عنوان یک قابلیت کاربرد یدر لایه سرویس، نیاز به عمل "بسته بندی" داریم. نقشه Google بسته بندی شده، دو عملیات به نام های showAddress و showAddresses را در بر دارد که از متد POST پروتکل HTTP برای گرفتن یک مکان یا مجموعه ای از مکان ها به عنوان ورودی استفاده می کنند. این عملیات از طریق آدرس های زیر قابل فراخوانی هستند:

http://www.semcem.com/GoogleMap/showAddress
http://www.semcem.com/GoogleMap/showAddresses

سرویس وب دیگری که مدیر مشاپ به آن نیاز دارد، یک سرویس وب خارجی برای جمع آوری رزومه های آنلاین است. این سرویس, چندین عملیات را در بر دارد. عملیاتی که مد نظر مدیر مشاپ است، عملیات Search می باشد که یک ورودی به عنوان شرایط مورد نظر را از طریق متد POST پروتکل HTTP دریافت می کند (سناریو –جدول ۲) و مجموعه ای از رزومه ها که شامل آن شرایط هستند را به عنوان خروجی بر می گرداند (سناریو –جدول ۳).

همانطور که بیان شد، در سناریوی تعریف شده، مدیر مشاپ با سه سرویس وب RESTful مواجه است که در ادامه آنها را با نامهای Open Positions، Google Map و Online Resume مورد ارجاع قرار می دهیم.

- **لایه معنا**

در سناریوی مرجع تعریف شده، باید سرویس های وب مورد استفاده را با استفاده از مکانیزم حاشیه گذاری پیشنهادی، حاشیه گذاری کنیم. این حاشیه گذاری می تواند با بهره گیری از فرمت JSON ساده و یا فرمت JSON/RDF باشد. در سناریو –جدول های ۴، ۵ و ۶ ، این حاشیه گذاری ها و همچنین توصیفات معنایی کمینه سرویس های وب مشارکت کننده در مشاپ مورد نظر ، در فرمت JSON ساده آورده شده اند. سناریو –جدول ۷ ، نمونه ای از فرمت RDF/JSON را برای سرویس وب Open Positions نشان می دهد. سناریو –جدول ۸ ، تبدیل شده حاشیه گذاری سناریو –جدول ۴ را پس از تبدیل در فرمت RDF را نشان می دهند. به علت محدود بودن فضا، از آوردن تبدیل شده به فرمت RDF سایر سرویس های وب چشمپوشی می کنیم.

همانطور که در حاشیه گذاری های معنایی مشخص شده است هر سرویس، یک هستان شناسی مختص خود دارد. در اینجا سه هستان شناسی به نام های Job، CV و Location وجود دارند که با استفاده از زبان OWL نوشته شده اند. شماهای بالابر و پایین آور بر اساس این هستان شناسی ها عمل می کنند. برای نمونه شمای بالابر عملیات list از سرویس وب Open Positions در سناریو –جدول ۹ ، آورده شده است.

از آنجا که عملیات list از سرویس وب Open Positions و عملیات Search از سرویس وب Online Resume از دو modelReference متفاوت در خروجی و ورودی استفاده می کنند، الگوریتم تطبیق هستان شناسی یک فایل XSLT به نام alignJobToCV تولید می کند که خروجی های عملیات list را روی ورودی های عملیات search تنظیم می کند. این فایل در سناریو –جدول ۱۰، نشان داده شده است.

- **لایه کاربرد**

در سناریوی مرجع تعریف شده، مدیر مشاپ باید بتواند به راحتی سرویس های مورد نظر را برای ایجاد مشاپ، با هم ترکیب کند. برای این کار نیاز به یک ویرایشگر مشاپ داریم که از مدل پیشنهادی ما پشتیبانی کند. در این مقاله، ویرایشگری ساده با نام SemCEM[1] را پیاده سازی کرده ایم که با استفاده از توصیفات معنایی، به ایجاد نیمه خودکار مشاپ های سازمانی کمک می کند. این ویرایشگر بصورت زیر کار می کند:

1. http://www.semcem.com

Archive of SID
www.SID.ir

۱. کاربر، آدرس توصیفگر سرویس های وب مورد نظر خود را وارد می کند. این آدرس می تواند آدرس عملیات توصیفگر سرویس و یا آدرس مستندات API حاشیه گذاری شده توسط RDFa یا hRESTS باشد.

۲. کاربر، لیست عملیات سرویس وب را مشاهده می کند و از میان آنها، عملیات مورد نظر خود را طبق ترتیب اجرا، انتخاب می کند.

۳. کاربر، مشاپ را اجرا می کند. اگر عملیات های انتخاب شده، از نظر معنایی تطابق داشته باشند، اجرا شروع می شود. در غیر این صورت اجرا متوقف می شود.

- o اگر مقدار خصوصیت inputType اولین عملیات، خالی نباشد، یک فرم ورودی بر مبنایloweringSchema بصورت پویا تولید می شود و کاربر بخش های ضروری فرم را پر می کند (شکل ۵-۲).

۴. کاربر، خروجی مشاپ اجرا شده را مشاهده می کند.

در سناریوی مرجع تعریف شده، مدیر مشاپ پس از وارد کردن آدرس عملیات توصیفگر سرویس های وب مورد نظر، ترتیب اجرا را بصورت زیر مشخص می کند:

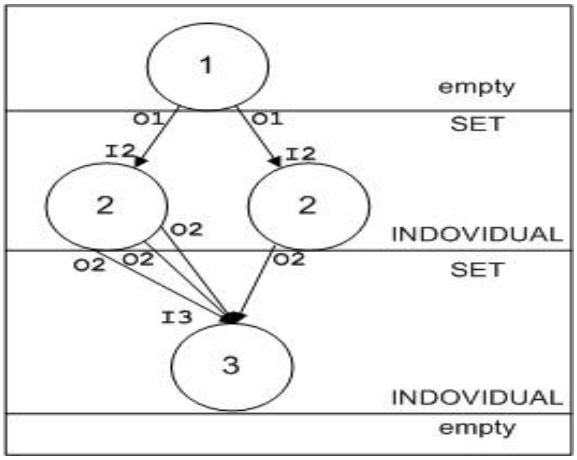
- عملیات list از سرویس وب Open Positions
- عملیات search از سرویس وب Online Resume
- عملیات showAddresses از سرویس وب Google Map

خروجی نهایی مشاپ ایجاد شده در SemCEM بر اساس مقادیر خصوصیات inputType و outputType عملیات ها نمایش داده می شود. این خصوصیات، الگوی فراخوانی عملیات را تعیین می کنند،

برای مثال در سناریوی مرجع تعریف شده، همانطور که در جدول ۳ و شکل ۸ نشان داده شده است، عملیات Search برای هر یک از خروجی های فراخوانی عملیات list ، فراخوانی می شود. سپس، خروجی های فراخوانی عملیات Search ، گردآوری شده و بصورت یک ورودی مفرد برای عملیات showAddresses ارسال می شوند و در نهایت، عملیات showAddresses فراخوانی می شود.

جدول ۳: مقادیر خصوصیات inputType وoutputType برای عملیات استفاده شده در سناریوی مرجع

#	Operation	*inputType*	*outputType*
1	positionOpening/list	Empty	SET
2	onlineResume/search	INDIVIDUAL	SET
3	GoogleMap/showAddresses	SET	Empty

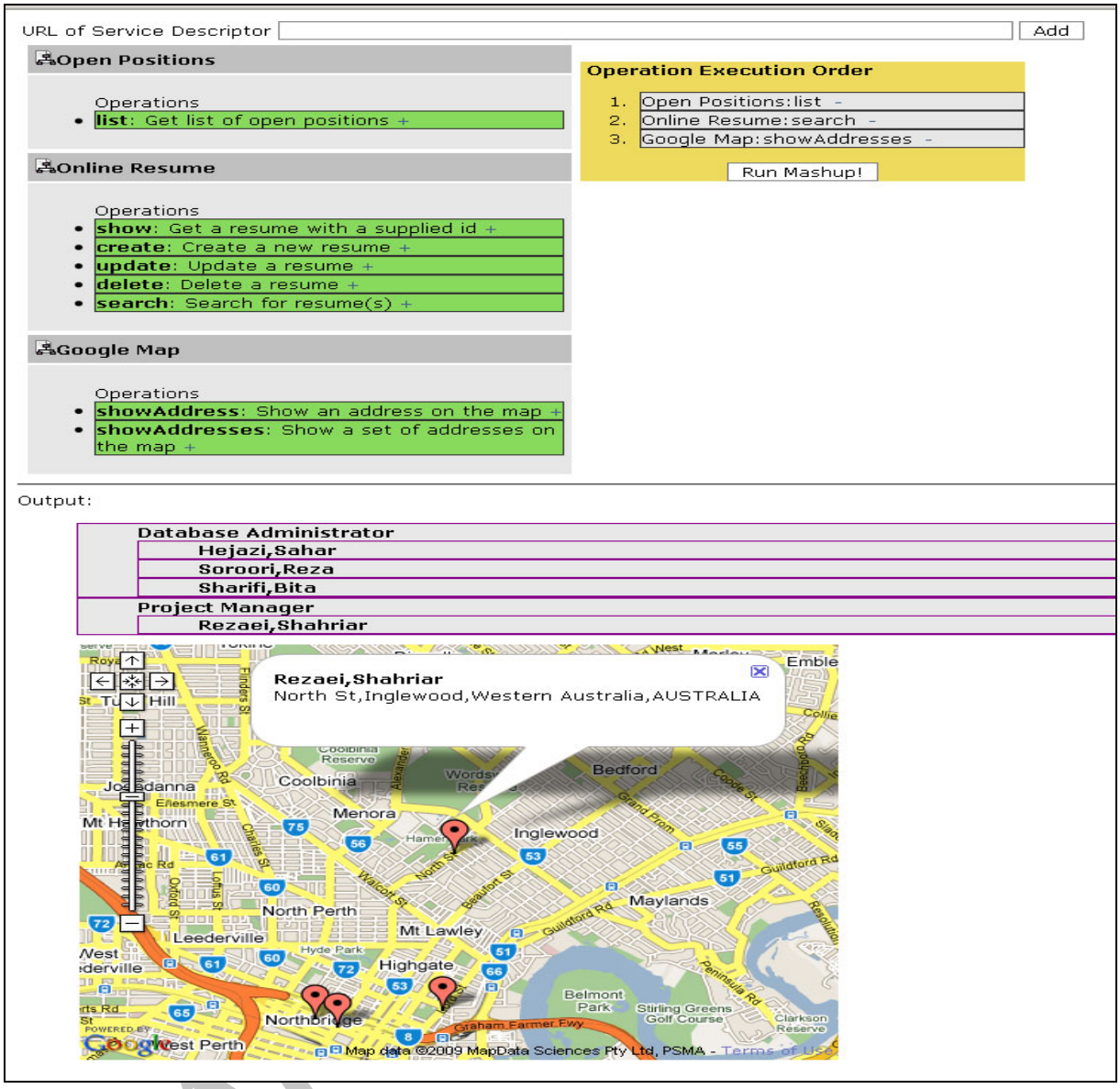


شکل ۸: الگوی فراخوانی عملیات در سناریوی مرجع

شکل ۹، نتیجه اجرای سناریوی کاربردی تعریف شده را نشان می دهد .همانطور که در لایه کاربرد مدل پیشنهادی بیان شد، افزودن توصیفات معنایی به سرویس های وب تشکیل دهنده مشاپ سازمانی، باعث تسهیل بسیاری از مراحل ایجاد آن می شود. در تمام ویرایشگر های مشاپ موجود، تعامل فقط در سطح نحوی صورت می گیرد و همین امر (نبود توصیفات معنایی) باعث می شود که کاربر بسیاری از مراحل ایجاد مشاپ را بطور دستی انجام دهد. همچنین، ایجاد مشاپ های پیچیده و ایمن با استفاده از ویرایشگرهای موجود بسیار دشوار است و نیازمند این است که یک کاربر متخصص فناوری اطلاعات وارد عمل شود. اگر بخواهیم سناریوی کاربردی تعریف شده را بدون در نظرگرفتن توصیفات معنایی پیاده سازی کنیم، کاربر بایستی در ابتدا به جستجوی اطلاعات یا قابلیت های مورد نظر خود بپردازد. از آنجا که هیچ مخزنی برای ذخیره سازی مشخصات منابع ناهمگن مورد نظر کاربر، وجود ندارد، این کار زمان زیادی خواهد برد. پس از یافتن این منابع و تفکیک آنها، کاربر بایستی به مطالعه مستندات سرویس Online Resume بپردازد. پس از آن، کاربر بایستی داده های ناهمگن خروجی موقعیت های خالی سازمان را که از ماژول HR سیستم ERP سازمان دریافت می کند بطور دستی با داده های ورودی سرویس Online Resume تنظیم کند. پس از آن نیز، بایستی همین عمل میانجیگری داده ها را بین خروجی سرویس Online Resume و سرویس Google Map انجام دهد. هیچ تضمینی نیز از لحاظ ایمن بودن اجرای مشاپ ایجاد شده وجود نخواهد داشت. تغییرات و سفارشی سازی این مشاپ سازمانی نیز در آینده بسیار پیچیده و زمانبر خواهد بود.

Archive of SID

www.SID.ir



شکل ۹: نتیجه اجرای مشاپ ایجاد شده برای سناریوی مرجع در SemCEM

۵-۳ ارزیابی کارایی مدل پیشنهادی و کارهای مرتبط

این ایده که کاربر بتواند بدون نوشتن کد، مشاپ های دلخواه خود را ایجاد کند، مورد توجه بسیاری از شرکت های فعال روی وب قرار گرفته است. شرکت های مختلفی مثل یاهو، گوگل، IBM و مایکروسافت ، ابزارهای ویرایشگر مشاپ خود را با قابلیت های فراوانی ارائه داده اند. در اینجا مقایسه های میان ابزار ویرایشگر حاصل این تحقیق (SemCEM) که از توصیفات معنایی بهره می گیرد و ابزارهای ویرایشگر موجود انجام می دهیم تا کارایی مدل پیشنهادی، بیشتر مشخص گردد.

Yahoo Pipes[1] یک برنامه کاربردی تحت وب است که با ارائه یک واسط گرافیکی به کاربران اجازه می دهد تا مشاپ های دلخواه خود را از منابع و سرویس های خاصی ایجاد کنند. این ابزار بر روی جمع آوری منابع از فراهم کننده های مختلف تمرکز دارد. واسط خوش ساخت این ابزار، کاربران غیر فنی را قادر می سازد تا ترکیب های نسبتا پیچیده ای از منابع را ایجاد کنند. یکی از اشکالات این ابزار، این است که محدود به استفاده از منابع و سرویس های خاصی است. برای مثال، استفاده از یک سرویس نقشه نیاز به نوشتن کدهای

1. http://pipes.yahoo.com/pipes

www.SID.ir

اضافی در خارج برنامه دارد. علاوه بر این، چون تمرکز روی منابع است، این ابزار نمی تواند سرویس های دارای ورودی و خروجی پیچیده را اداره کند. برای مثال، هیچ راهی برای ارسال داده های XML از طریق متد POST به یک سرویس وب در این ابزار وجود ندارد.

Google Mashup Editor[1] یک سرویس جدید Google است که هنوز در مرحله آزمایشی است. بر خلاف Yahoo Pipes که روی منابع تمرکز دارد، این ابزار روی ایجاد صفحات وب متمرکز است. واسط متنی این ابزار، کاربران را قادر می سازد تا با استفاده از برچسب های HTML و یک سری برچسب های اختصاصی از قبل تعریف شده برای اشیاء پیچیده تری مثل Google Map، صفحات وب ترکیبی دلخواه خود را ایجاد کنند. نقطه ضعف اصلی این ابزار، این است که منابع مورد استفاده در آن حتما باید دارای داده های توصیفی اضافی باشند تا بتوان آنها را در کنار اشیاء پیچیده ای مثل نقشه مورد استفاده قرار داد. علاوه بر این، سرویس های قابل استفاده در این ابزار، محدود به سرویس های ارائه شده توسط Google مثل نقشه یا جستجوی وب هستند. از آنجا که این ابزار از یک واسط متنی استفاده می کند، کاربر قادر خواهد بود تا با نوشتن کدهای سفارشی جاوااسکریپت یا HTML ، قابلیت های مشاپ ایجاد شده را بالا ببرد اما از طرف دیگر، منحنی یادگیری کاربر نیز شیب زیادی خواهد داشت و کاربر مجبور است علاوه بر شناخت برچسب های سفارشی ابزار، درک زیادی از HTML، جاوااسکریپت و XML داشته باشد.

IBM QEDWiki[2] چیزی شبیه بوم نقاشی[3] است که کاربر روی آن مشاپ دلخواه خود را طراحی می کند. این ابزار به کاربر اجازه می دهد تا از سرویس های وب مبتنی بر SOAP، سرویس های وب RESTful و منابع مختلف برای ایجاد مشاپ استفاده کند. این ابزار، نسبت به دو ابزار معرفی شده قبلی بسیار قدرتمندتر و انعطاف پذیرتر می باشد اما این قدرت و انعطاف پذیری هزینه هایی را نیز در پی دارد. کاربر در این ابزار، نیاز به شماهای تطابق ورودی و خروجی سرویس ها و نیز داشتن یک درک خوب از منطق برنامه نویسی دارد. این ابزار، بیشتر یک سکو است که کاربران به کمک آن کدهای ایجاد مشاپ دلخواه خود را بنویسند.

در تمام ویرایشگر های مشاپ موجود، تعامل فقط در سطح نحوی[4] صورت می گیرد و همین امر (نبود توصیفات معنایی) باعث می شود که کاربر بسیاری از مراحل ایجاد مشاپ را بطور دستی انجام دهد. جدول ۴ مقایسه ای را میان ابزار ویرایشگر این مشاپ و ابزارهای بررسی شده در بالا نشان می دهد. همانطور که مشاهده می شود، SemCEM به دلیل استفاده از توصیفات معنایی، بسیاری از مراحل مربوط به ایجاد مشاپ را خودکار کرده است. در نتیجه دانش برنامه نویسی مورد نیاز برای توسعه دهنده در آن بسیار کم است. علاوه بر این از تمامی متدهای پروتکل HTTP پشتیبانی می کند و مولفه های تشکیل دهنده مشاپ در آن به راحتی قابل جایگزینی هستند.

جدول ۴: مقایسه SemCEM با ویرایشگرهای مشاپ موجود

امکانات/ ویرایشگر	SemCEM	Google Mashup Editor	Yahoo Pipes	IBM QEDWiki
پشتیبانی از سرویس های نمایشی مانند Google Map	✓	✓	×	✓
پشتیبانی از HTTP GET	✓	✓	✓	✓
پشتیبانی از HTTP POST	✓	×	×	✓
پشتیبانی از HTTP PUT	✓	×	×	✓
پشتیبانی از HTTP DELETE	✓	×	×	×
قابلیت جایگزینی مولفه های مشاپ بدون نیاز به تنظیم مجدد پیکربندی مشاپ	✓	×	×	×
قابلیت توسعه و به اشتراک گذاری مشاپ ایجاد شده	✓	✓	✓	✓
دانش برنامه نویسی مورد نیاز برای توسعه دهنده	بسیار کم	متوسط	کم	زیاد

1. http://code.google.com/gme/

2. http://services.alphaworks.ibm.com/qedwiki/
3. Canvas
4. Syntactic

www.SID.ir

۶- نتیجه گیری و تحقیقات آتی

مشاپ های سازمانی، نسل بعدی برنامه های کاربردی در سازمانها را تشکیل می دهند. این برنامه ها توسط کارگران دانش که دانش محدودی در زمینه برنامه نویس دارند ایجاد می شود. این کاربران نسبت به محیط آگاهی خوبی داشته و نزدیک ترین فرد به مساله ای هستند که برنامه مربوطه برای حل آن توسعه داده می شود. اگرچه در حال حاضر ابزارهای زیادی برای تسهیل فرایند ایجاد مشاپ های سازمانی ارائه شده اند اما این فرایند همچنان زمانبر و پیچیده است و نیاز دارد که توسعه دهنده مشاپ تجربه هایی را در زمینه مولفه ها و فرمت داده های استفاده شده در مشاپ، داشته باشد. در این مقاله یک مدل جدید برای ایجاد نیمه خودکار مشاپ های سازمانی ارائه شده است. مدل پیشنهادی از چهار لایه تشکیل شده است: لایه منبع، لایه سرویس، لایه معنا و لایه کاربرد.

منابع که بلوک های اصلی سازنده مشاپ های سازمانی هستند توسط مکانیزم "بسته بندی" بصورت سرویس در می آیند. این سرویس ها توسط یک مدل حاشیه گذاری معنایی شبیه مدل SAWSDL اما با تفاوت های جزئی، حاشیه گذاری می شوند. یک عملیات جدید برای هر سرویس وب پیشنهاد شده است که به عنوان توصیفگر سرویس عمل می کند. عملیات توصیفگر، از فرمت داده ای JSON برای انتقال توصیفات نوشته شده در مدل داده ای RDF استفاده می کند. این مکانیزم حاشیه گذاری جدید، یک طبیعت پویا به حاشیه گذاری ها می دهد بطوریکه می توان آنها را بصورت پویا و بر اساس تغییرات صورت گرفته در سرویس بهنگام رسانی کرد. حاشیه گذاری های معنایی توصیفات معنایی را به سرویس های وب متصل می کنند و باعث تسهیل ترکیب سرویس ها مستقل از جنبه های فنی آنها می شوند. بدین وسیله، کارگران دانش که توسعه گران مشاپ های سازمانی هستند قادر خواهند بود بدون داشتن دانش برنامه نویسی و در کمترین زمان ممکن، مشاپ های سازمانی ایمن و دلخواه خود را ایجاد کنند.

برای نشان دادن مدل پیشنهادی و امکان سنجی آن، از یک سناریوی کاربردی استفاده کردیم. این سناریو مربوط به ایجاد یک مشاپ سازمانی برای تسهیل فرایند جذب نیرو در سازمان بود. این سناریو توسط ویرایشگر مشاپ پیاده سازی شده بر اساس مدل پیشنهادی، شبیه سازی و نتیجه اجرای آن ارائه گردید.

دستاورد اصلی این تحقیق، فراهم آوردن امکان ایجاد مشاپ های سازمانی ایمن توسط کارگران دانش در سازمان، در کمترین زمان ممکن و بدون داشتن مهارت های برنامه نویسی وب می باشد. استفاده از توصیفات معنایی، می تواند بسیاری از مراحل ایجاد مشاپ های سازمانی را تسهیل ببخشد. در نتیجه راه بسوی خلاقیت و نوآوری در ایجاد مشاپ های با ارزش افزوده به جای صرف وقت در مسائل مربوط به پیاده سازی آنها، باز می گردد. همچنین، مشاپ های ایجاد شده، دارای انعطاف پذیری بالایی بوده و به راحتی قابل تغییر و سفارشی سازی هستند.

به عنوان تحقیق بیشتر در آینده در نظر داریم که ویژگی های ویرایشگر مشاپ خود را ارتقا دهیم به نحوی که بتواند از یک مدل جدید برای ارتباطات بین دامنه ای ایمن[1] در مرورگر وب استفاده کند. همچنین قصد داریم روی الگوریتم های تطبیق هستان شناسی ها تحقیق بیشتری انجام داده و پیاده سازی قوی تری را برای انجام عمل میانجیگری داده ها ارائه دهیم. علاوه بر این، در نظر داریم تا از Semantic Style Sheets به عنوان یک راه حل جایگزین دیگر برای انتقال توصیفات معنایی مدلسازی شده در RDF بهره ببریم. تحقیق بیشتر روی ایجاد قوانین در وب معنایی برای اجرای سیاست های سازمان نیز جزو تحقیقات آتی بشمار می آید.

مراجع

[1] Young, G., et al. The Mashup Opportunity. s.l. : Forrester, 2008.

[2] Bradley, A. and Gootzit, D. Who's Who in Enterprise Mashup Technologies. s.l. : Gartner Research, 2007.

[3] Serious Business - Web 2.0 goes Corporate. s.l. : The Economist Intelligence Unit, 2007.

[4] Kongdenfha, W., et al. , Rapid Development of Spreadsheet-based Web Mashups. Madrid : s.n., 2009. WWW 2009.

[5] Oasis: SOA Adoption Blueprint. [Online] 2006. http://www.oasis-open.org/committees/download.php/17616/06-04-00002.000.doc.

[6] Liu, X., et al. , Towards service composition based on mashup. 2007. IEEE International Conference on Service Computing (SCC 2007). pp. 332–339.

[7] Business Process Execution Language for Web Services version 1.1. [Online] February 8, 2007. http://www.ibm.com/developerworks/library/specification/ws-bpel/.

[8] Web Service Choreography Interface (WSCI) 1.0. [Online] August 8 , 2002. http://www.w3.org/TR/wsci/.

[9] Ease of interaction plus ease of integration: Combining Web2.0 and the Semantic Web. Heath, T. and Motta, E. s.l. : Journal of Web Semantics, Elsevier, 2007.

[10] O'Reilly, T. What is Web 2.0? Design Patterns and Business Models for the Next Generation of Software. [Online] September 2005.

1. Secure Cross-Domain Communication

Archive of SID
www.SID.ir

International Journal for Geographic Information and Geovisualization, Vol. 41, pp. 187–199.
[29] Cho, A. , An introduction of mashups for health libranrians. 1, 2007, Journal of the Canadian Health Libraries Association, Vol. 28, pp. 19–22.
[30] Watt, S. Mashups - the evolution of the soa, part 2: Situational applications and the mashup ecosystem. [Online] 2007. http://www.ibm.com/developerworks/webservices/library/ws-soa-mashups2/.
[31] Clarkin, L., Holmes, J. , Enterprise mashups. 2007, The Architecture Journal, Vol. 13.
[32] Salesforce: Mashups: The what and why. [Online] 2007. http://wiki.apexdevnet.com/index.php/.
[33] Wikipedia: Mashups. [Online] 2008. http://en.wikipedia.org.
[34] Sapir, J. Situational Applications: Cost-effective software solutions for immediate business challenges. [Online] February 22, 2009. http://www.powerinthecloud.com/.
[35] Makki, S. K. and Sangtani, J. s.l. , Data Mashups & Their Applications in Enterprises.: IEEE, 2008. IEEE ICIW 2008.
[36] . R., Kailar و. s.l. , Reasoning about Accountability in Protocols for Electronic Commerce: IEEE Computer Society, 1995. IEEE Symposium on Security and Privacy. p. 236.
[37] Zou, J. and Pavlovski, C.J. , Towards accountable enterprise mashup services. 2007. IEEE International Conference on e-Business Engineering (ICEBE 2007). pp. 205-212.
[38] . Khalili, A. and Mohammadi, S. s.l. , Using Logically Hierarchical Meta Web Services to Support Accountability in Mashup Services: IEEE, 2008. IEEE APSCC2008. pp. 410-415.
[39] . Jackson, C. and Wang, H. , Subspace: Secure crossdomain communication for web mashups2007. 6th International Conference on the World-Wide Web. pp. 5-10.
[40] Hinchcliffe, D. The 10 top challenges facing enterprise mashups. [Online] October 16, 2007. http://blogs.zdnet.com/Hinchcliffe/?p=141.
[41] Hoyer, V. and Fischer, M. s.l. , Market Overview of Enterprise Mashup Tools.: Springer-Verlag Berlin Heidelberg, 2008. ICSOC 2008. Vol. LNCS 5364, pp. 708–721.
[42] Carrier, N., et al. The business case for enterprise Mashups. Web 2.0 technology solutions White paper. [Online] 2008. www.ibm.com/software/info/Mashup-center/library.html.
[43] Paikari, E., Habibi, J. and Yeganeh, S. H. , Semantic Composability Measure for Semantic Web Services. 2007. First Asia International Conference on Modelling & Simulation (AMS'07). pp. 88- 93.
[44] Haller, A., et al. s.l. , WSMX - a semantic service-oriented architecture.: IEEE, 2005. IEEE International Conference on Web Services. pp. 321- 328.
http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-isWeb-20.html.
[11] Hendler, J. and Golbeck, J. s.l. , Metcalfe's law, Web 2.0, and the Semantic Web.: Journal of Web Semantics, Elsevier, 2007.
[12] Floyd, I. R., et al. s.l. , Web mash-ups and patchwork prototyping: User-driven technological innovation with Web 2.0 and open source software.: Annual Hawaii International Conference on System Sciences (HICSS'07), 2007. pp. 86– 95.
[13] Gartner's top 10 strategic technologies for 2008. [Online] October 9, 2007. http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9041738.
[14]). Battle, R. and Benson, E. 1, s.l. , Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST: Elsevier Science Publishers, 2008, Vol. 6, pp. 61-69 .
[15] Antoniou, G. A semanticWeb primer/. s.l. : Massachusetts Institute of Technology, 2004. 0-262-01210-3.
[16] PASSIN, T. B. Explorer's Guide to the Semantic Web. s.l. : Manning Publications, 2004. 1-932394-20-6.
[17] Hausenblas, M. s.l. , Exploiting Linked Data for Building Web Applications.: IEEE Internet Computing, 2009.
[18] Ankolekar, A., et al. , The two cultures: Mashing up Web 2.0 and the Semantic Web. s.l. : Journal of Web Semantics, Elsevier, 2007.
[19] . Janner, T., et al. s.l. , Enterprise Mashups: Putting a face on next generation global SOA: Springer, 2007. WISE 2007. Vol. LNCS 483.
[20] Soriano, J., et al. , Foster Innovation in a Mashup-oriented Enterprise 2.0 Collaboration Environment. 2007. System and Information Sciences Notes 1. Vol. 1, pp. 62–68.
[21] Daniel, F., et al. , Understanding UI Integration. A Survey of Problems, Technologies, and Opportunities. 11, s.l. : IEEE, 2007, IEEE Internet Computing, Vol. 3, pp. 59–66.
[22] Blogs, mashups, wikis oh my. Dearstyne, B. 4, 2007, Information Management Journal, Vol. 14, pp. 24–33.
[23] O'Brien, D. and Fritzgerald, B., Mashups, remixes and copyright law. 2, 2007, Internet Law Bulletin, Vol. 9, pp. 17–19.
[24] Gerber, R. 8, Mixing it up on the web: Legal issues arising from the internet mashup., 2007, Intellectual Property and Technology Law Journal, Vol. 18, pp. 11–14.
[25] The Economist: Mashing the web. s.l. : The Economist - Special Section, 2005. p. 376.
[26] Hof, R. Mix, match, and mutate. s.l. : Business Week Magazine, 2005.
[27] Mashups: Emerging application development paradigm for a digital journal. Kultathuramaiyer, N. 1, 2007, Journal of Universal Computer Science, Vol. 13, pp. 531–542.
[28] Miller, C., A beast in the field: The google maps mashup at gis/2. 3, 2007, Cartographica -The

Archive of SID

www.SID.ir

[45] Chow, S. W. PHP Web 2.0 Mashup Projects. s.l. : Packt Publishing, 2007. 978-1-847190-88-8.
[46] Gudgin, M., et al. SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). [Online] April 27 , 2007. http://www.w3.org/TR/soap12-part1/.
[47] Ajax and Mashup Security. Open Ajax Alliance. [Online] 2008. http://www.openajax.org.
[48] Fensel, D., et al. Enabling Semantic Web Services -The Web Service Modeling Ontology. s.l. : Springer Berlin Heidelberg, 2007. 978-3-540-34519-0.
[49] Martin, D. and al., et. OWL-S: Semantic Markup for Web Services. W3C Member Submission. [Online] November 22, 2004. http://www.w3.org/Submission/OWL-S/.
[50] Battle, S. and al., et. Semantic Web Services Framework (SWSF). W3C Member Submission . [Online] September 9, 2005. http://www.w3.org/Submission/SWSF/.
[51] Vitvar, T., et al. s.l. , WSMO-Lite Annotation for Web Services.: Springer, 2008. 5th European Semantic Web Conference (ESWC 2008).
[52] Hadley, M. Web Application Description Language (WADL). [Online] April 2006. https://wadl.dev.java.net/.
[53] Kopecky, J., et al. , SAWSDL: Semantic Annotation for WSDL and XML Schema. 2007. IEEE Internet Computing. pp. 60-67.
[54] Sheth, A. P., Gomadam, K. and Lathem, J. , SA-REST: Semantically Interoperable and Easier-to-Use Services and Mashups2007. IEEE Internet Computing. pp. 91-94.

Archive of SID

www.SID.ir