

معماری کتابخانه‌های دیجیتال

مهدی علیپورحافظی^۱

مهشید کریمی^۲

چکیده

امروزه، کتابخانه‌های دیجیتال به تدریج، جای خود را به عنوان یک ضرورت در جامعه باز می‌کنند و از ابزاری تجملی به ابزاری مؤثر و مفید تبدیل می‌شوند. آشنایی کتابداران با این پدیده، به جهت ضرورت حضور آن‌ها در تیم‌های طراحی کتابخانه دیجیتال، حائز اهمیت است. بیان جایگاه معماری در طراحی کتابخانه دیجیتال و نیز معرفی انواع معماری‌های مورد استفاده در کتابخانه‌های دیجیتال با بررسی متون موجود محقق گردیده است تا هدایتگر کتابداران در این زمینه باشد. انتخاب نوع معماری بسته به نوع نیاز در کتابخانه‌های مختلف، متفاوت است و می‌تواند از معماری یک لایه تا چند لایه متغیر باشد. لذا انتخاب معماری مناسب و متناسب با نیاز از میان انواع معماری‌های موجود، از اهمیت بسزایی برخوردار است.

فرآیند طراحی کتابخانه دیجیتال با تحلیل سیستم آغاز و با طراحی معماری و برنامه‌نویسی ادامه می‌یابد و در مراحل بعدی با آزمون و نصب و پشتیبانی به تکامل می‌رسد و محصول نهایی یعنی کتابخانه دیجیتال متولد می‌شود. معماری نرم‌افزار با معماری ساده سرویس‌دهنده که مربوط به نسل‌های اولیه طراحی نرم‌افزارهاست، آغاز و با معماری پیچیده سه لایه سرویس‌دهنده/سرویس‌گیرنده، که از جمله معماری تولید نرم‌افزار کتابخانه دیجیتال است، به پیشرفت خود ادامه می‌دهد. امروزه نرم‌افزارهای کتابخانه دیجیتال غالباً با استفاده از این نوع معماری تولید می‌شوند. از جمله این معماری‌ها می‌توان به معماری هاروست، سارا، ادپت و غیره اشاره کرد. برای آشنایی بیشتر با این معماری‌ها مثال‌هایی از نرم‌افزارهای کتابخانه دیجیتال نظیر کتابخانه دیجیتال گرین استون، اسکندریه و غیره آورده شده است که به فهم بهتر و سریع‌تر متن کمک می‌کند.

کلیدواژه‌ها

کتابخانه‌های دیجیتال، معماری، مدل‌های معماری، نرم‌افزار.

۱. دانشجوی دکتری علوم کتابداری و اطلاع‌رسانی دانشگاه آزاد اسلامی، واحد علوم و تحقیقات تهران meh.hafezi@gmail.com

۲. کارشناس کتابداری و اطلاع‌رسانی دانشگاه علامه طباطبائی meh.karimi@yahoo.com

مقدمه

کتابخانه دیجیتال، کتابخانه‌ای است که منابع و خدمات کتابخانه‌ای را به صورت دیجیتال در اختیار کاربران قرار می‌دهد. این کتابخانه‌ها در راستای توسعه کتابخانه‌ها از اواسط سده بیستم به بعد مطرح شدند و از اواخر همین سده پا به عرصه وجود گذاشتند. این کتابخانه‌ها در امتداد توسعه و پیشرفت کتابخانه‌های سنتی مطرح شدند و نسل جدیدی از کتابخانه‌ها را معرفی کردند. بنابراین کتابخانه‌های دیجیتال نمی‌خواهند و نمی‌توانند جایگزین کتابخانه‌های سنتی شوند، بلکه همان‌گونه که اشاره شد این کتابخانه‌ها برای پاسخ‌گویی به نیازهای نوین کاربران به وجود آمده‌اند. از این‌رو، از فن‌آوری‌های نوین نیز در این راستا بهره می‌جویند. امروزه با ظهور و گسترش اینترنت و وب گسترده جهانی، محیطی جهانی برای ارائه اطلاعات و خدمات کتابخانه‌ها فراهم شده است تا کتابخانه‌ها بتوانند خود را، همیشه و همه جا در اختیار کاربران قرار دهند. سرعت پیشرفت فن‌آوری‌های اطلاعاتی و ارتباطی در عصر حاضر، طبعاً نیازها و توقعات افراد را نیز تحت‌تأثیر قرار داده است و به طور خلاصه باید گفت که امروزه سرعت زندگی، به تبع سرعت پیشرفت فن‌آوری، بالا رفته و تنها راه چاره، تلاش بیش‌تر و همگام شدن با این پیشرفت است. همین افراد طبیعتاً برای گذران امور روزمره خود و نیز پژوهش که بخشی از زندگی روزمره ما را تشکیل می‌دهد راهی به کتابخانه‌ها خواهند یافت. پاسخ‌گویی به خیل عظیم نیازهای کاربران در عصر حاضر نیاز به

حضور کتابخانه‌های دیجیتال را توجیه می‌کند. امروزه در سراسر جهان حرکت کتابخانه‌ها به این سمت، امری ملموس و قابل دفاع است. در ایران نیز بسیاری از کتابخانه‌ها به تبعیت از این ضرورت، حرکت به این سمت را آغاز کرده‌اند و یا در حال برنامه‌ریزی برای ایجاد آن هستند. ولی متأسفانه این امر نیز همانند بسیاری از موارد مشابه به شکل تب‌انی و زودگذر، خود را در جامعه مدیریت عالی سازمان‌ها و دانشگاه‌های کشور نمایان کرده است. چنین توقعاتی ثمراتی چون عدم تهیه محصولی مناسب، صرف هزینه‌های هنگفت بدون نتیجه مطلوب و عدم برنامه‌ریزی و اجرای صحیح را به همراه داشته است. مدیران در سازمان‌ها همواره در پی فعالیت‌های زودبازده مطابق با عمر مدیریتی خود هستند که این خود می‌تواند دلایل مختلفی داشته باشد که خارج از بحث ماست، در صورتی که ایجاد کتابخانه دیجیتال، فعالیتی زمان‌بر و طولانی‌مدت است. در کنار تهیه نرم‌افزار کتابخانه دیجیتال تبدیل منابع منحصربه‌فرد کتابخانه‌ها به فرمت دیجیتال نیز به نوبه خود فعالیتی زمان‌بر و پرهزینه است که خود باعث ایجاد شک و تردید در مدیران جهت سرمایه‌گذاری در این زمینه می‌شود.

کتابخانه دیجیتال به علت حجم فعالیت‌ها در جهت ارائه منابع تمام‌متن دیجیتال و نیز خدمات دیجیتال، نیاز به نرم‌افزاری دارد که بتواند توقعات مذکور را جامعه عمل بپوشاند. در این راستا، کتابخانه دیجیتال نیاز به استفاده از مجموعه نرم‌افزارهای مختلف برای ارائه خدمات که از ارائه اطلاعات تمام‌متن و

جستجو در منابع گرفته تا ایجاد امنیت در سیستم و سطح دسترسی کاربران دارد. از آن جایی که محیط فعالیت کتابخانه دیجیتال، محیطی رایانه‌ای در سطح شبکه جهانی اینترنت است، باید از قواعد و استانداردهای نرم‌افزارهای رایانه‌ای تحت شبکه نیز تبعیت کند. همان‌گونه که در ایجاد کتابخانه سنتی باید مراحل را در جهت طراحی و ایجاد ساختمان کتابخانه‌هایی که متناسب با نوع فعالیت کتابخانه باشند، طی کرد، در ایجاد راه‌اندازی کتابخانه‌های دیجیتال نیز باید مراحل خاصی را در این جهت پیمود تا محصولی قابل قبول و بالاخص قابل اطمینان جهت استفاده در اختیار داشت. طراحی ساختمان کتابخانه‌های سنتی با مشارکت کتابداران در فرآیند طراحی و توسط متخصصان مربوطه انجام می‌گیرد. در کتابخانه‌های دیجیتال نیز چنین اقدامی باید صورت پذیرد تا به نتیجه مطلوب دست یافت. مهم‌ترین مسئله در این میان مشارکت کتابداران در امر طراحی است که همانند بسیاری از موارد دیگر در طراحی‌های کتابخانه‌های دیجیتال داخلی نادیده گرفته می‌شود. شاید یکی از علل آن عدم آشنایی ما کتابداران با مباحث مربوط به طراحی نرم‌افزارها و بالاخص نرم‌افزار کتابخانه دیجیتال باشد!

طراحی نرم‌افزار کتابخانه دیجیتال، همان‌گونه که قبلاً ذکر شد، بی‌شبهت به طراحی نرم‌افزار نیست. لذا باید با فرآیند طراحی و مراحل آن آشنا بود و دانست که در هر مرحله چه فعالیت‌هایی با چه کیفیتی باید انجام گیرد. پس شناخت جایگاه معماری

در طراحی نرم‌افزار و شناخت انواع معماری نرم‌افزار کتابخانه دیجیتال در این فرآیند ضرورت می‌یابد. مقاله حاضر با این رویکرد به طرح مسائل مذکور در جهت بیان مراحل طراحی نرم‌افزار کتابخانه دیجیتال به صورت کلی و با تأکید خاص بر معماری کتابخانه دیجیتال و توصیف آن به نحوی که مطلوب نظر کتابداران و متخصصان این حوزه باشد، می‌پردازد تا شاید بتواند گامی در راستای آشنایی متخصصان کتابداری با معماری کتابخانه دیجیتال بردارد. در این صورت کتابداران می‌توانند جایگاه خود را در این میان درک کنند و به انجام مطالعات گسترده‌تر در این زمینه مشتاق‌تر شوند.

معماری و جایگاه آن در طراحی کتابخانه دیجیتال

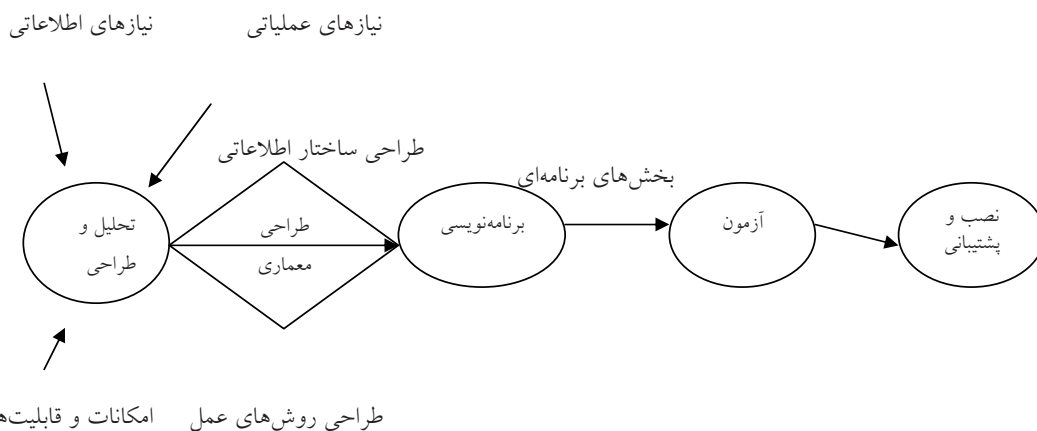
هر جا که نیاز به طراحی موجودیت یا سیستمی باشد که ابعاد یا پیچیدگی آن از یک واحد معین فراتر برود، یا نیازمندی‌های خاصی را تحمیل کند، نگرشی ویژه و همه‌جانبه را لازم خواهد داشت که در اصطلاح به آن معماری گفته می‌شود (۲). بدین معنی که سیستم‌های ساده بدون گذراندن مرحله معماری می‌توانند محقق شوند ولی اگر خواسته باشیم سیستمی با عملکردهای پیچیده داشته باشیم، نیازمند معماری متناسب با آن هستیم. نقشه و طراحی، در ساخت یک بنا مثال مناسبی برای این امر است. زمانی که قصد بنای یک چهاردیواری کوچک داشته باشیم، نیازی به نقشه‌کشی و مراعات اصول ریز و جزئی آن نیست ولی هر وقت بخواهیم یک ساختمان ده طبقه بنا

کنیم، حتماً به نقشه‌ای کامل با رعایت تمامی استانداردها و ضوابط نقشه‌کشی نیاز خواهیم داشت. بنابراین برای تهیه این نقشه، طراح ساختمان باید ترکیبی از علم، هنر و تجربه را به کار بندد تا بتواند طرحی دقیق و متناسب با نیاز را تهیه کند. در طراحی نرم‌افزار نیز چنین اصلی مراعات می‌شود. زمانی که قصد طراحی نرم‌افزاری را برای تولید اعداد تصادفی دارید، طبیعتاً نیاز به معماری، یا همان نقشه در ساختمان، نیست. ولی مطمئناً شما نیز با این نظر موافق هستید که برای طراحی نرم‌افزارهایی با رفتارهای پیچیده مانند کتابخانه دیجیتال که اعمالی نظیر سفارش، سازمان‌دهی، ذخیره، بازیابی، اشاعه اطلاعات و سایر موارد دیگر جزء اجزای عملیاتی آن به شمار می‌روند، ضرورتاً نیاز به معماری متناسب با نوع عملکرد آن وجود دارد. بنابراین معماری، ارائه توصیفی فنی از یک سیستم است که نشان‌دهنده ساختار اجزای آن، ارتباط بین آن‌ها و اصول و قواعد حاکم بر طراحی و تکامل آن‌ها در گذر زمان باشد (۲).

از بدو مطرح شدن نرم‌افزار تاکنون، معماری‌های متفاوتی به منظور طراحی و پیاده‌سازی ارائه شده است. معماری‌های مذکور از یک طرف برخاسته از امکانات و ماهیت سخت افزارها در زمان خود و از طرف دیگر نمایانگر نوع و نگرش انتظارات طرح شده توسط کاربران است. به خاطر داشته باشیم که نرم‌افزار دارای ماهیتی پویا بوده است و در هر زمان، می‌بایست مطابق با خیل عظیم نیازها و انتظارات جدید کاربران باشد؛ چرا که نرم‌افزار عصاره خواسته‌های انسانی

به منظور بالفعل شدن در بستر سخت‌افزار در گذر زمان است. بدیهی است از گذشته تاکنون، هم طیف خواسته‌های انسانی تغییر کرده و خواهد کرد و هم سخت‌افزارها دچار تغییر و تحول گسترده‌ای بوده و خواهند بود. در این راستا، لازم است نرم‌افزار نیز با رعایت کامل اصل انعطاف‌پذیری، پذیرای تمامی تحولات از گذشته تاکنون باشد و بتواند در هر زمان رسالت خود را به خوبی انجام دهد. بر همین اساس تاکنون معماری‌های متفاوتی به منظور طراحی و پیاده‌سازی نرم‌افزار کتابخانه دیجیتال ارائه شده است. هر معماری دارای شاخص‌ها و ویژگی‌های منحصر به فرد خود بوده و نرم‌افزارهایی که با اتکا به هر یک از معماری‌های فوق پیاده‌سازی می‌گردند، خصایص خود را از معماری به کار گرفته شده به ارث خواهند برد (۱).

طراحی نرم‌افزار، به طور کلی، هفت مرحله امکان‌سنجی، آنالیز و تحلیل، طراحی (منطقی و فیزیکی)، پیاده‌سازی، آزمون (ارزیابی و تعیین صحت نرم‌افزار)، کنترل کیفی (قابلیت اطمینان نرم‌افزار) و نصب سیستم و پشتیبانی را دربرمی‌گیرد. مراحل مذکور باید به ترتیب به اجرا درآیند تا بتوان محصولی قابل اطمینان در اختیار داشت. زمانی که بررسی‌های اولیه و مرحله شناخت (مرحله امکان‌سنجی و آنالیز) به اتمام برسد و حاصل آن مراحل تأیید شود، کار ایجاد سیستم نرم‌افزاری شامل مراحل تحلیل و طراحی، پیاده‌سازی و آزمون و کنترل کیفی نرم‌افزار شروع می‌شود. جریان کار ایجاد سیستم در طی این مراحل، در شکل زیر نشان داده شده است.

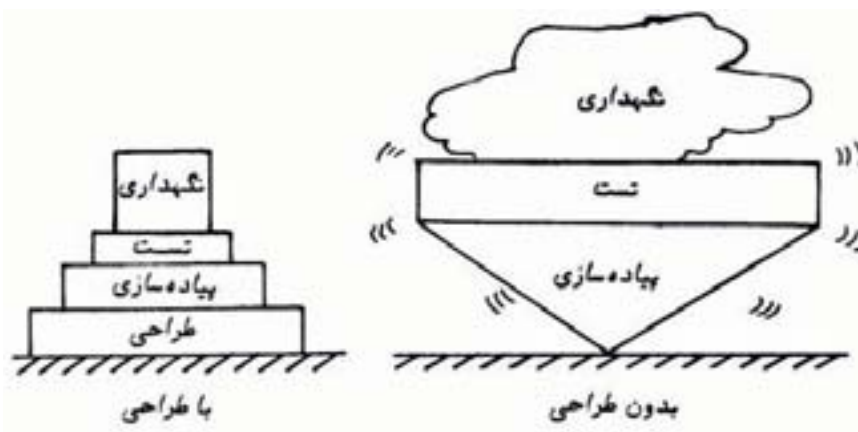


شکل ۱. مراحل طراحی نرم‌افزار

به طراحی سیستم به عنوان نقطه آغاز کار ایجاد سیستم بستگی پیدا می‌کند. اهمیت مرحله طراحی از این نکته روشن می‌شود (۴: ۱۰۱-۱۰۲).

اهمیت معماری را می‌توان در یک کلمه: «کیفیت» خلاصه کرد. معماری سیستم تنها راهی است که به کمک آن می‌توان نیازهای کاربر را به صورت یک سیستم نرم‌افزاری درآورد. معماری نرم‌افزار به منزله پایه‌ای برای پیاده‌سازی، آزمون و نگهداری سیستم است.

با نگاهی اجمالی به شکل، می‌توان متوجه شد که مراحل اولیه، پایه و اساس مراحل بعدی هستند و بدون گذر از یک مرحله نمی‌توان به مرحله بعد گام گذاشت. مرحله ایجاد سیستم نرم‌افزاری تقریباً ۷۵ درصد هزینه کلی سیستم (منهای هزینه نگهداری) را به خود اختصاص می‌دهد. به همین دلیل، موفقیت در پیاده‌سازی یک نرم‌افزار کتابخانه دیجیتال و سادگی تعمیر و نگهداری آن مهم بوده و توجیه‌گر این هزینه است و همه این‌ها



شکل ۲. اهمیت معماری (۴)

بدون معماری، سیستم ناپایداری خواهیم داشت که هر آن احتمال فرو ریختن آن می‌رود و امکان نگهداری آن وجود ندارد. شکل زیر عوارض ناشی از یک طراحی ضعیف را نشان می‌دهد. همان‌طور که ملاحظه می‌شود، معماری عبارت از تبدیل نیازها به نمایشی از نرم‌افزار است. روشن است که در آغاز، چنین نمایشی از نرم‌افزار ممکن است قدری کلی و مبهم باشد، ولی با دقیق‌سازی و تصفیه آن می‌توان به طرحی رسید که به برنامه‌هایی که باید نوشته شوند بسیار نزدیک باشد.

به منظور طراحی خوب و با کیفیت یک نرم‌افزار کتابخانه دیجیتالی و به‌طور اعم یک نرم‌افزار، موارد زیر را می‌توان در نظر گرفت:

۱. طراحی باید متضمن سازمانی سلسله‌مراتبی بوده و بین عناصر نرم‌افزار کنترلی هوشمندانه را ممکن کند.

۲. طراحی باید بخش‌بخش باشد، بدین معنی که نرم‌افزار را باید به عناصری که کارهای معینی را انجام می‌دهند، تقسیم‌بندی کرد.

۳. طراحی باید شامل نمایشی روشن و مجزا از داده‌ها و روش عمل باشد.

۴. طراحی باید به بخش‌هایی ختم شود که هر کدام از آن‌ها کار جداگانه و مستقلاً را انجام دهند (۴: ۱۰۳).

یک معماری خوب در دل مرحله طراحی خود را نمایان می‌کند. با داشتن معماری منطقی و متناسب با نیاز کاربران، که در راستای اهداف از پیش تعیین شده برای

نرم‌افزار است، می‌توان به داشتن محصول خوب و مطمئن امیدوار بود. کتابخانه دیجیتالی نیز ضرورتاً نیاز به معماری پیچیده‌ای دارد که با گذر زمان روند تکاملی خود را طی کرده است تا امروزه شاهد کتابخانه‌هایی با کارایی و اطمینان بالا باشیم.

معماری نرم‌افزار

در این بخش، با معرفی انواع معماری‌های تولید نرم‌افزار، سعی می‌شود زمینه‌های لازم به منظور شناخت معماری به کار گرفته شده در نرم‌افزار کتابخانه دیجیتالی فراهم شود. در معرفی این معماری‌ها بیش‌تر تلاش شده تا سیر تکاملی معماری‌ها، از معماری‌های ساده به پیچیده که می‌تواند به نوعی سیر تاریخی نیز محسوب شود، مراعات شود:

- **معماری سرویس‌دهنده**^۳. در این معماری، بخش‌های مختلف برنامه - داده و منطق^۴ - در یک محل ذخیره می‌شوند و از یک محیط پردازش یکسان برخوردارند. در این معماری مسئولیت انجام تمام پردازش‌ها با رایانه اصلی^۵ است، که محل استقرار داده و منطق برنامه است. اعمال این معماری به دلیل تمرکز داده‌ها و منطق برنامه بر روی یک سیستم، آسان و امنیت در آن بسیار بالاست. ولی همان‌گونه که مشخص است با از کار افتادن سیستم اصلی، تمامی سرویس‌ها قطع خواهد شد و این می‌تواند به‌عنوان معضل بزرگ این معماری محسوب شود.

- **معماری سرویس‌دهنده فایل**^۶. در معماری

3. Main frame

4. Data & logic

5. Host

6. File server

سرویس دهنده فایل، که چرخشی ۱۸۰ درجه‌ای نسبت به معماری سرویس دهنده دارد، داده و منطق برنامه بر روی سرویس دهنده مستقر است. داده‌ها می‌توانند داده‌های اشتراکی باشند که توسط سرویس دهنده گردآوری شده‌اند. سرویس دهنده به محض دریافت درخواست کاربر، داده و منطق مربوطه را در سیستم سرویس گیرنده اجرا می‌کند. با توجه به این ویژگی، وجود محدودیت‌های خاصی نظیر میزان حافظه و یا نوع پردازشگر سرویس گیرنده‌ها، می‌تواند میزان برخورداری مفید از یک برنامه با این معماری را کاهش دهد. بنابراین، سرویس دهنده، وظیفه حفظ داده‌ها و منطق برنامه را برعهده دارد و با دریافت درخواست از طرف سرویس گیرنده، امور مربوطه به سیستم سرویس گیرنده منتقل می‌شود. این نوع معماری در شبکه‌های داخلی کوچک قابل اجرا بوده و همان‌گونه که اشاره شد، معضل اصلی آن، محدودیت‌های احتمالی سرویس گیرنده است. با این وجود، حتی اگر سیستم‌های سرویس گیرنده قوی نیز تهیه می‌شدند، هزینه بالای تهیه چنین سیستم‌هایی می‌تواند غیرقابل توجیه باشد (۱).

● **معماری سرویس دهنده/ سرویس گیرنده.**
در این معماری، برخلاف دو معماری قبل، اصل تقسیم کار دنبال می‌شود و از سرویس دهنده‌ها و سرویس گیرنده‌هایی با خصایص متفاوت استفاده می‌شود. تقسیم وظایف بین سرویس دهنده و سرویس گیرنده، از ویژگی‌های این معماری است که امروزه نیز در برخی از امور از آن استفاده می‌شود. عملیات سنگین با پردازش بالا برعهده

سرویس دهنده و عملیات سبک برعهده سرویس گیرنده است. تاکنون مدل‌های متفاوتی از این معماری پیاده‌سازی شده است. از جمله این مدل‌ها می‌توان به:

○ مدل پردازش‌های مبتنی بر میزبان - که بیش از آن‌که به منزله یک مدل سرویس دهنده/ سرویس گیرنده محسوب شود، مشابه مدل سرویس دهنده است.

○ پردازش‌های مبتنی بر سرویس دهنده - که انجام تمامی پردازش‌ها برعهده سرویس دهنده و مسئولیت اجرای بخش رابط کاربر برعهده سرویس گیرنده است.

○ پردازش‌های مبتنی بر سرویس گیرنده - که تمامی عملیات بر روی سرویس گیرنده انجام می‌شود و عملیات مربوط به بررسی صحت داده‌ها و منطق بانک‌های اطلاعاتی، بر روی سرویس دهنده اجرا می‌شود. و نیز:

○ پردازش‌های مبتنی بر همیاری - که سرویس دهنده و سرویس گیرنده، به منظور انجام یک فعالیت با یکدیگر تشریک مساعی می‌کنند (۱).

● **معماری دولایه.** این معماری نیز مشابه معماری سرویس دهنده/ سرویس گیرنده است، و همان‌طور که از نامش پیداست از دولایه، یک سرویس دهنده و یک سرویس گیرنده، تشکیل می‌شود. این دولایه شامل سه بخش هستند: بخش‌های رابط کاربر، مدیریت پردازش‌ها و مدیریت بانک‌های اطلاعاتی. در این معماری، منطق برنامه میان محل فیزیکی این دولایه (سرویس دهنده و سرویس گیرنده) توزیع می‌شود و رابط کاربر به طور مستقیم با منابع تأمین داده مرتبط می‌شود. بنابراین رابط

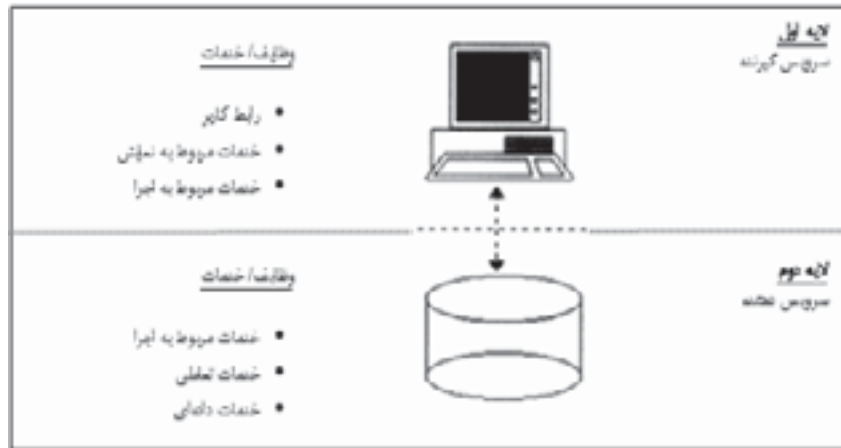


شکل ۳. مدل‌های معماری سرورس دهنده/ سرورس گیرنده (۱)

مجموعه‌ای از ابزارها به منظور دستیابی به منابع سیستم بوده است و مسئولیت مدیریت پردازش‌ها، تجزیه و یا ترکیب نتایج حاصله از منابع داده‌ای (نظیر بانک‌های اطلاعاتی) را برعهده دارد، که می‌تواند به دو و یا بیش از دو بخش^۷ با عملکردهای متمایز تقسیم شود. در این معماری، امنیت داده‌ها ارتقا می‌یابد، چرا که ارتباط سرورس گیرنده با لایه میانی برقرار می‌شود و مدیریت درخواست‌ها و فعالیت‌ها توسط این لایه انجام می‌شود (۱). حال که تا حدودی، زمینه لازم جهت آشنایی با معماری کتابخانه‌های دیجیتال فراهم

کاربر در سیستم سرورس گیرنده اجرا می‌شود و درخواست‌های سرورس گیرنده را به سیستم سرورس دهنده، که حاوی داده‌ها و پردازش‌های مربوط به آن‌هاست، ارسال می‌کند و داده‌های بازیابی شده را در اختیار کاربر قرار می‌دهد. سیستم‌های کتابخانه‌ای تحت شبکه موجود از این معماری سود می‌جویند.

● معماری سه لایه. در واقع مدل کامل‌تر معماری دولایه است که علاوه بر دولایه سرورس دهنده بانک‌های اطلاعاتی و سرورس گیرنده (رابط کاربر)، از یک لایه میانی نیز برخوردار است. لایه میانی شامل



شکل ۴. معماری دولایه (۱)

سرویس‌هایی که یک کتابخانه دیجیتال قادر به ارائه آن‌هاست، به شدت به این اجزا و ارتباط آن‌ها وابسته است. در پاره‌ای موارد، تعدادی از سرویس‌ها جزء سرویس‌های اولیه و زیربنایی هستند و سرویس‌های دیگر بر پایه آن‌ها ارائه می‌شوند. در این گونه موارد، خود سرویس به عنوان جزئی از معماری سیستم در کنار سایر اجزا قرار می‌گیرد و با آن‌ها ارتباط دارد، به طوری که در برخی از معماری‌های ارائه شده بسیاری از اجزای سیستم را سرویس‌های سیستم تشکیل می‌دهند.

علاوه بر این، باید به این نکته نیز توجه داشت که چون کتابخانه‌های دیجیتال با اهداف مختلفی ایجاد می‌شوند، معماری‌های ارائه شده نیز اهداف مختلفی را برآورده می‌سازند و هر کدام یک یا چند جنبه برجسته دارند و به همین دلیل است که گاهی برخی از اجزا در یک سیستم مشاهده می‌شوند، در حالی که در سیستم دیگر مشاهده نمی‌شوند و یا در سیستم‌های مختلف دارای ارتباطات

شد، می‌توانیم به بررسی انواع معماری‌های به کار رفته در کتابخانه‌های دیجیتال پردازیم. نکته‌ای که باید به آن اشاره شود، این است که به دلیل وسعت فعالیت‌های کتابخانه‌های دیجیتال و سرویس‌های متنوعی که ارائه می‌شود، بسته به کاربرد و محیط، ممکن است تنها روی بخش خاصی از آن تمرکز شود و به همین دلیل نیز معماری‌های ارائه شده بسیار متنوع هستند و در بسیاری از موارد جامع نیستند و در آن‌ها فقط به یک یا چند بخش خاص توجه شده است. هدف ما نیز از این بررسی، ارائه‌اشنایی اجمالی با تعدادی از انواع معماری‌های کتابخانه‌های دیجیتال است.

نکته مهم دیگری که باید به آن توجه شود، این است که معماری یک کتابخانه دیجیتال و سرویس‌هایی که آن کتابخانه ارائه می‌کند، دو مقوله مستقل نیستند، بلکه کاملاً یکدیگر را تحت تأثیر قرار می‌دهند. در واقع، معماری یک کتابخانه دیجیتال عبارت است از رابطه‌ای که بین اجزای سیستم وجود دارد. در نتیجه،

متفاوتی با اجزا دیگر هستند. همچنین لازم به ذکر است که در زمینه کتابخانه‌های دیجیتال، به دلیل جدید بودن فن آوری، مشخص نبودن دقیق مرزهای آن و عدم وجود استانداردهای فراگیر برای بسیاری از بخش‌ها، هنوز برای این کتابخانه‌ها، معماری با پایه و اجزای کاملاً استاندارد وجود ندارد. اکنون با مقدمه‌ای که اشاره شد، به معرفی برخی از معماری‌های کتابخانه‌های دیجیتال اجرا شده می‌پردازیم.

● معماری هاروست^۸ (معماری یک جستجوی توزیع‌یافته): این معماری در زمینه جستجوی توزیع یافته روی انواع منابع دیجیتال موجود در مراکز مختلف و اینترنت است و تنها امکان جستجوی توزیع‌یافته روی منابع و مجموعه‌های دیجیتال را ایجاد می‌کند. این معماری از چهار جزء: گردآورنده^۹، دلال^{۱۰}، حافظه موقت اشیاء^{۱۱} و مدیر کپی‌ها^{۱۲} تشکیل شده است. گردآورنده، اطلاعات نمایه‌سازی شده را از مجموعه‌های کتابخانه‌ای و اینترنتی فراهم می‌آورد. دلال با استفاده از این اطلاعات، یک پایگاه اطلاعاتی پویا ایجاد کرده است و به درخواست کاربران پاسخ می‌دهد. مدیر کپی نیز جهت کپی گرفتن از پایگاه اطلاعاتی دیگر در پایگاه اطلاعاتی محلی مورد استفاده قرار می‌گیرد. به این ترتیب علاوه بر افزایش سرعت پاسخ‌گویی به درخواست‌های کاربران، میزان ترافیک شبکه نیز کاهش می‌یابد. مدیر کپی‌ها علاوه بر انجام این محلی‌سازی اطلاعات، به روزرسانی اطلاعات کپی شده با اطلاعات اصلی رانیز برعهده دارد (۳: ۴۸-۴۹، ۷۴: ۱۶).

در مجموع، این معماری از اولین روش‌های جستجوی توزیع شده است و به جای استفاده از یک پایگاه اطلاعاتی رابطه‌ای قدرتمند، از یک سیستم فایل ساده برای نگهداری اطلاعات استفاده می‌کند. اما درعین حال، می‌تواند کارایی زیادی در انجام جستجوی توزیع‌یافته داشته باشد. درکل، می‌توان گفت که اطلاعات نمایه‌سازی شده در این مجموعه‌های کتابخانه‌ای و اینترنتی، در پایگاه اطلاعاتی گردآوری می‌شود و منابع اطلاعاتی نیز توسط مدیرکپی‌ها، نسخه‌برداری و به صورت فایل‌هایی در سرویس‌دهنده ذخیره می‌شوند. در این روش، کاربر با جستجو در پایگاه اطلاعاتی می‌تواند به اطلاعات مورد نیاز خود دست یابد (۸: ۱۶؛ ۱۵: ۷).

با وجود این‌که، معماری هاروست یک مدل مناسب و کارآمد برای جستجوی توزیع‌یافته ارائه می‌کند، اما نمی‌تواند معماری مناسبی برای یک کتابخانه دیجیتال جامع باشد. زیرا اولاً به خودی خود دارای مدل داده‌ای مستقلی نیست و بر اساس یک لایه زیرین که سرویس‌های ذخیره و نگهداری داده را ارائه می‌دهد بنا می‌شود. ثانیاً تمامی سرویس‌های یک کتابخانه دیجیتال را ارائه نمی‌کند؛ چرا که سرویس‌های یک کتابخانه دیجیتال فراتر از یک سرویس جستجوی توزیع‌یافته است. در نهایت این‌که این معماری می‌تواند معماری مناسبی برای ذخیره و بازیابی داده‌ها در پایگاه‌های داده پراکنده باشد ولی نمی‌توان آن را معماری مناسبی برای کتابخانه دیجیتال دانست.

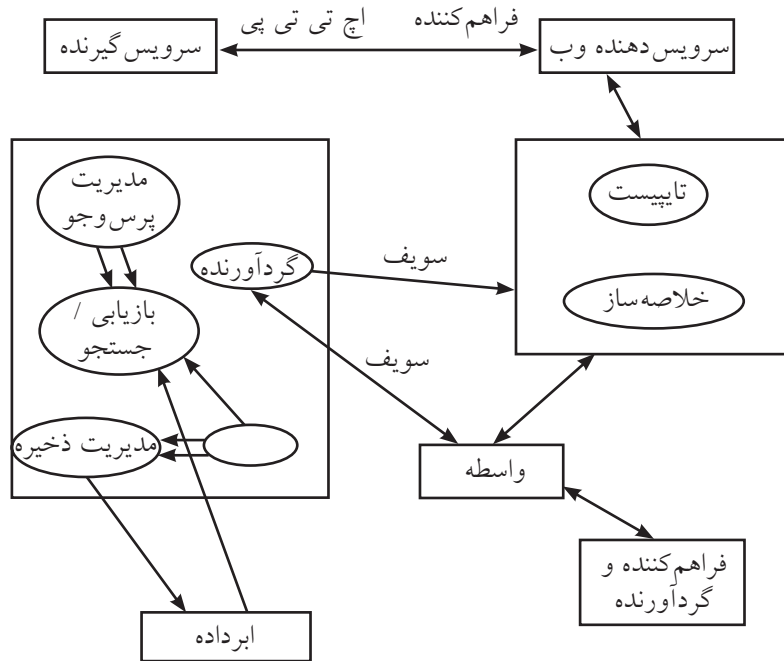
8. Harvest architecture

9. Gatherer

10. Broker

11. Object cache

12. Replition manager

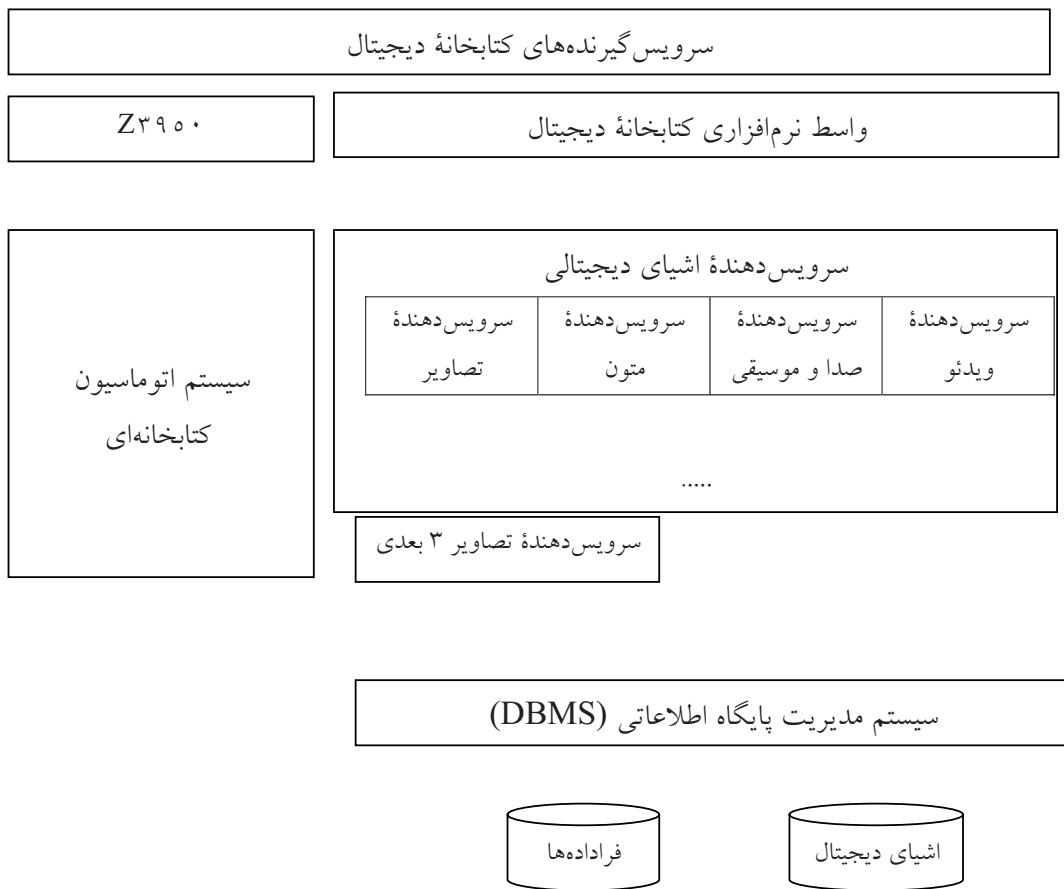


Summary Object Interchange Format (SOIF) = سوئیف*

شکل ۵. معماری هاروست (۶)

تا به حال در دست سرویس گیرنده بود، به سرویس دهنده منتقل می شود. عامل های متحرک یکی از راهبردهای کارآمد در زمینه پردازش و تحلیل اطلاعات هستند. به ویژه در خصوص اطلاعات توزیع یافته، به کارگیری آنها به میزان زیادی، سرعت پردازش را افزایش می دهد و ترافیک شبکه را کم می کند. در این معماری، درخواست کاربر به پایگاه های داده مستقل ارسال می شود و عمل جستجو و بازیابی در آنها توسط مدیریت پایگاه داده انجام می پذیرد و پاسخ به رابط کاربری در سمت سرویس گیرنده ارسال می شود. بنابراین در این معماری،

• معماری سارا^{۱۳} (معماری یک جستجوی توزیع یافته مبتنی بر عامل متحرک): این معماری مبتنی بر عامل و یا عامل های متحرک است. عامل متحرک با حجم کم از طریق شبکه به محلی که اطلاعات در آن واقع است، حرکت می کند و به طور محلی در کنار داده ها اجرا می شود. خودمختاری، استقلال، عدم نیاز به سرپرست و ارتباط مستمر با کاربر، از مزایای مهم عامل متحرک است. در این معماری، هنگام حرکت عامل متحرک، کنترل عامل نیز منتقل می شود. مثلاً هنگامی که عامل از طرف سرویس گیرنده به طرف سرویس دهنده حرکت می کند، کنترل اجرای عامل نیز که



شکل ۶. معماری عمومی ارائه شده توسط بوگن

پایگاه داده خاصی وجود ندارد و سرعت بازایی اطلاعات زیاد است. این معماری نیز مشابه معماری قبلی، تنها یک مدل جستجوی توزیع شده را ارائه می‌کند و نمی‌تواند یک مدل جامع برای کتابخانه دیجیتال باشد (۳: ۵۳-۵۴، ۷۵؛ ۱۹: ۵۴).

- معماری عمومی ارائه شده به وسیله بوگن^{۱۴}. این معماری از چهار لایه تشکیل شده است:
 ۱. اشیا دیجیتال و فراداده‌های مرتبط به آن‌ها
 ۲. سرویس دهنده اشیا دیجیتالی
 ۳. واسط نرم افزاری
 ۴. سرویس گیرنده

در این مدل، جنبه‌های بیشتری از کتابخانه دیجیتال مثل انبار داده با قابلیت ذخیره انواع داده‌ها، پروتکل Z39.50 برای انجام جستجوی توزیع یافته، سرویس‌های اتوماسیون کتابخانه‌ای و واسط کاربر پوشش داده شده است (۳: ۵۷-۵۸، ۷۵؛ ۷: ۱۸).

- معماری سیستم داینست^{۱۵}. داینست

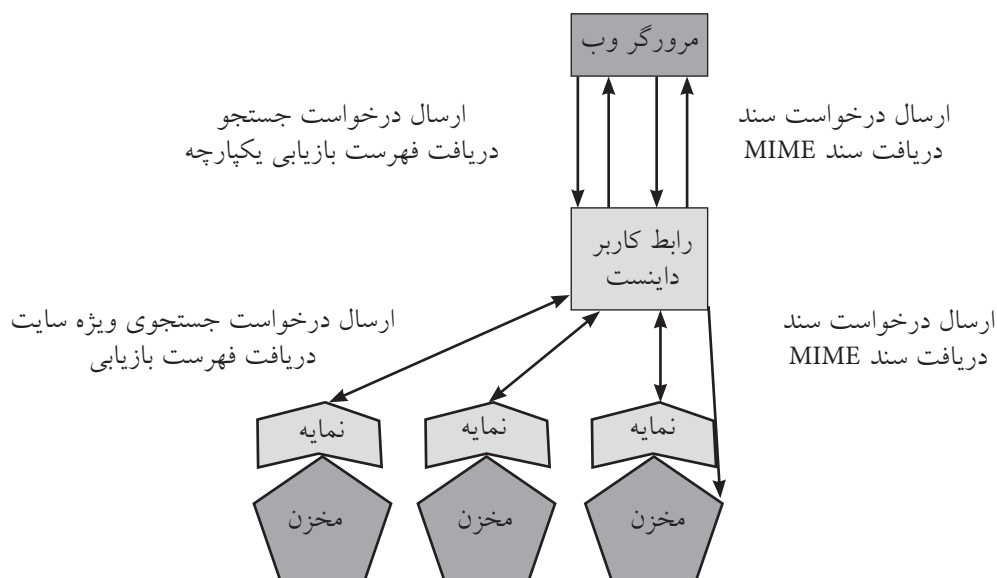


ارتباط بین سرویس‌های مجزا، از طریق تفاهم‌نامه‌ی باز و مشخصی انجام می‌گیرد که امکان ترکیب این سرویس‌ها را به شیوه‌های ابداعی و با ایجاد لایه‌های سرویس، بر روی سرویس‌های موجود و پایه‌ای فراهم می‌کند. این معماری، یک مدل کاملاً علمی و جامع برای کتابخانه دیجیتال است. در عین حال، ایراد این معماری، پیچیدگی بیش از حد مدل داده‌ای آن است (۹: ۶؛ ۱۰: ۴۳).

● معماری کتابخانه دیجیتال اسکندریه^{۱۶}. پروژه کتابخانه دیجیتال اسکندریه در سال ۱۹۹۵، با توسعه منابع جغرافیایی و خدماتی برای دسترسی به منابع آغاز شد. عملیات اجرایی پروژه در دانشگاه کالیفرنیا در سانتا باربارا^{۱۷} انجام پذیرفت. کتابخانه دیویدسون^{۱۸} میزبانی کتابخانه دیجیتال اسکندریه را برعهده دارد.

سیستمی برای پیکربندی مجموعه‌ای از خدمات مجرد است که در سرویس‌دهنده‌های توزیع‌یافته برای همکاری در ارائه خدمات کتابخانه دیجیتال اجرا می‌شود (۱۶: ۱۱). اجزای اصلی این معماری عبارتند از:

- مخزن، که مسئول نگهداری مدارک است.
 - سرویس‌ها، که با استفاده از پروتکل‌های مشخصی می‌توان از قابلیت‌های آن‌ها استفاده کرد.
 - مجموعه‌ها، که شامل مدارک هستند.
 - نواحی، که توزیع شدن مجموعه‌ها را امکان‌پذیر می‌کند (۱۱؛ ۱۲).
- این معماری از چند سرویس مجزا که بعد از ترکیب، یک کتابخانه دیجیتال توزیع یافته را ایجاد می‌کنند، تشکیل شده است.

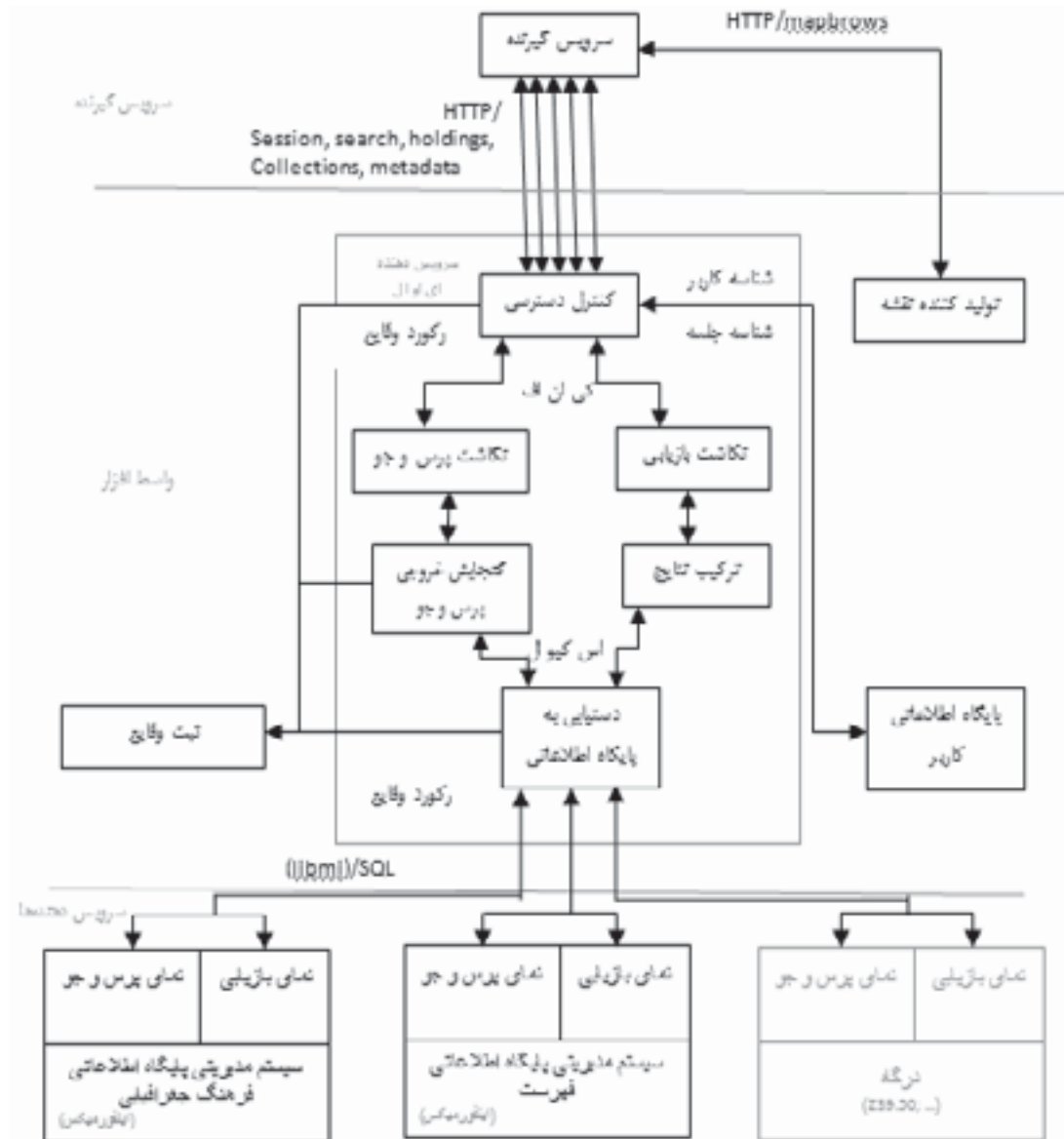


شکل ۷. تعامل سرویس‌های داینست (۱۶)

16. Architecture of Alexandria Digital Library (ADL)

18. Davidson

17. Santa Barbara



*کی ان اف (Kevin's Normal Format)

شکل ۸. معماری کتابخانه دیجیتال اسکندریه (۱۴)

(۶). این کتابخانه از معماری سه لایه، سرویس گیرنده، واسط افزار و سرویس دهنده بهره گرفته است (۱۶: ۸).

● معماری کتابخانه دیجیتال گرین استون^{۱۹}. کتابخانه دیجیتال گرین استون با همکاری دانشگاه وایکاتو^{۲۰}، یونسکو و پروژه کتابخانه‌های علوم انسانی^{۲۱} با هدف تهیه اطلاعات برای جهان در حال توسعه راه اندازی شد. گرین استون اطلاعات درخواستی را بازیابی و برای نمایش آن‌ها صفحات اچ تی ام ال را تولید می‌کند. طرح اولیه سیستم شامل بخش‌های زیر است:

● رایانه دستی قابل حمل^{۲۲}. سرویس گیرنده‌ای که مرورگر اچ تی ام ال را اجرا می‌کند. این مرورگر صفحات اچ تی ام ال تولید شده را نمایش می‌دهد. مرورگر مذکور قادر به ارسال درخواست‌ها و دریافت پاسخ‌های اچ تی تی پی^{۲۳} است. این قابلیت با استفاده از فن آوری بی سیم تی سی پی^{۲۴} امکان پذیر می‌شود. این فن آوری امکان اجرای شبکه بی سیم بدون نیاز به ترجمه اچ تی تی پی به سایر تفاهم‌نامه‌ها را فراهم می‌کند.

● ابزار شخصی سازی^{۲۵}. ابزاری که جهت شخصی سازی گرین استون برای نمایشگرهای کوچک کاربران^{۲۶} قابل استفاده است. این ابزار فایل‌های پیکربندی^{۲۷} را تولید می‌کند که توسط گرین استون برای تولید صفحات

اچ تی ام ال - که با اولویت‌های کاربران همخوانی داشته و همچنین مناسب با ابزارهای نمایشی کوچک است - مورد استفاده قرار می‌گیرد. ابزار شخصی سازی، خروجی کاملی از داده‌های گرین استون فراهم می‌کند و امکان شخصی سازی نمایش صفحات اچ تی ام ال توسط کاربران را فراهم می‌کند.

● صفحه خانگی. امکان تعریف ساختار صفحه خانگی و ایجاد قابلیت جستجو در مجموعه‌های خاص در صفحه خانگی را برای کاربران فراهم می‌آورد.

● ساختار سند. ساختار اسناد را در هر مجموعه مشخص می‌کند.

● اولویت‌های جستجو. امکان شخصی سازی تسهیلات جستجو (به عنوان مثال، جستجوی عمومی یا جستجو براساس تاریخ) را برای کاربران فراهم می‌کند.

● ساختار صفحه. امکان شخصی سازی نحوه نمایش صفحات (به عنوان مثال، چه چیزی در سرآیند^{۲۸} و پاورقی هر صفحه نمایش داده شود) را برای کاربران فراهم می‌کند (۱۶: ۱۳؛ ۱۷).

● معماری ادپت^{۲۹}. این معماری در قالب اجزا و جریان اطلاعاتی بین آن‌ها بیان می‌شود و از سه جزء عناصر (اساسی‌ترین جزء)، مجموعه‌ها و کتابخانه‌ها تشکیل یافته است. در این معماری سه نوع اطلاعات فراداده‌ها،

19. Architecture of Green Stone Digital Library (GSDL)

20. University of Waikato

21. Human libraries project

22. Portable Digital Assistant (PDA)

23. HTTP= Hypertext Transfer Protocol

24. TCP= Transport Control Protocol

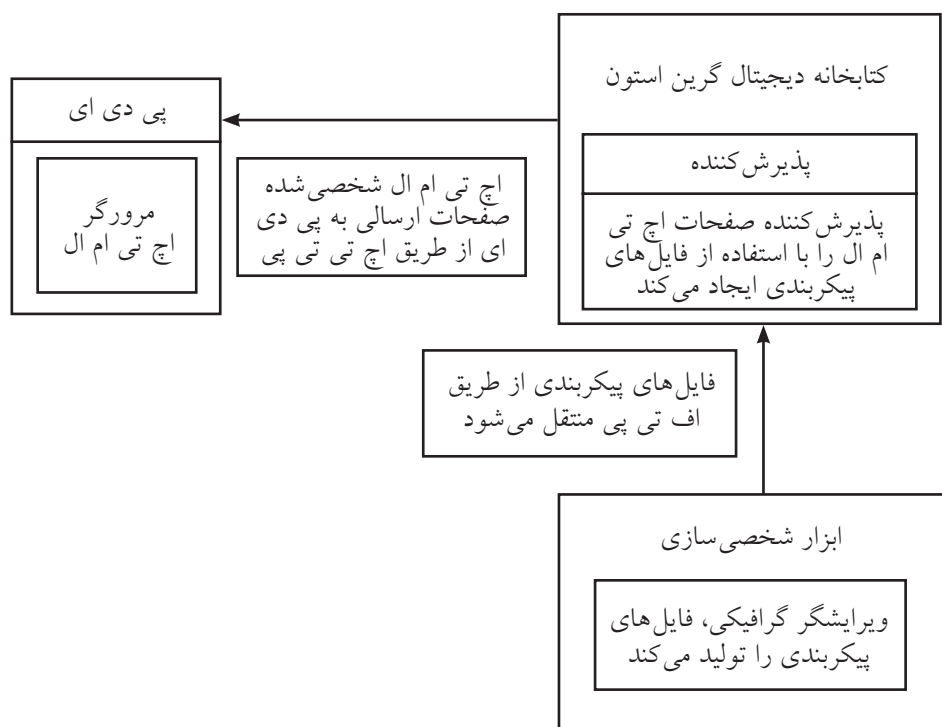
25. Customization tool

26. Personal Digital Assistant (PDA)

27. Configuration files

28. Heather

29. ADEPT = The Alexandria Digital Earth Proto Tipe



شکل ۹. معماری کتابخانه دیجیتال گرین استون (۱۶)

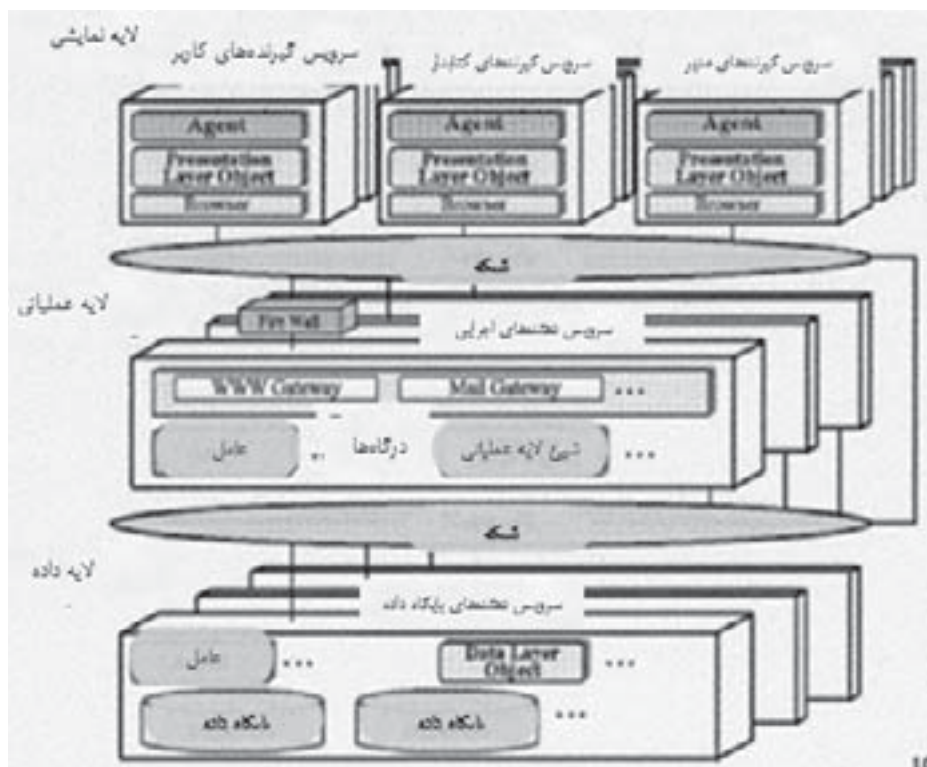
معماری سه لایه

به منظور طراحی و تولید نرم افزار کتابخانه دیجیتال، با توجه به ماهیت این نوع کتابخانه ها، محوریت کار طراحی معماری و تولید نرم افزار، روی سرویس ها، خدمات و فعالیت هایی که این کتابخانه ها می بایست ارائه کنند، قرار می گیرد. بدین معنی که، سرویس های متعدد و متنوع، وسعت فعالیت ها و خدمات، و همچنین خیل عظیم کاربران (به دلیل عدم وجود محدودیت های مکانی و زمانی کتابخانه های سنتی)، نرم افزار کتابخانه های دیجیتال را در ردیف نرم افزارهای پیچیده و سنگین قرار می دهند. طراحی و تولید چنین نرم افزارهایی،

پرسش ها و مجموعه نتایج بین اجزا مبادله می شود. مهم ترین هدف این معماری، ارائه سرویس های اطلاعاتی سطح بالا و کارآمد، بر روی مجموعه هاست که این مجموعه ها می توانند هر ساختار فراداده ای دلخواهی داشته و یا دارای استانداردهای متفاوتی باشند. این معماری از نظر مدل فراداده ای بسیار کارآمد است و سرویس جستجوی قدرتمندی را ارائه می کند. با این حال مدل کاملی برای یک کتابخانه دیجیتال نیست، زیرا فاقد سرویس های متنوع است. اما می تواند به عنوان زیربنای فراداده ای یک کتابخانه دیجیتال به کار گرفته شود (۳، ۶۸-۶۹، ۵۷).

با استفاده از معماری‌های پیچیده چندلایه به‌ویژه معماری‌های سه لایه (سرویس‌دهنده / سرویس‌گیرنده)^{۳۰} صورت می‌پذیرد. بر همین اساس، برای طراحی معماری نرم‌افزار کتابخانه‌های دیجیتال، از مدل‌های معماری سه لایه، به‌ویژه مدل تقسیم‌بندی براساس نوع خدمات، استفاده می‌شود. البته نحوه استفاده از این معماری در پروژه‌های کتابخانه‌های دیجیتال، متفاوت است. این تفاوت‌ها بسته به نوع خدمات، منابع، نوع و تعداد کاربران، می‌تواند زیاد یا کم باشد. اما با کمی دقت می‌توان به استفاده از معماری سه لایه در این

نرم‌افزارها پی برد. تمامی نرم‌افزارهایی که با استفاده از مدل تقسیم‌بندی براساس نوع خدمات، طراحی می‌شوند، کلیه خدمات و سرویس‌های خود را در سه بخش سرویس‌های رابط کاربر^{۳۱}، سرویس‌های پردازش^{۳۲}، سرویس‌های داده^{۳۳} ارائه می‌کنند (۱). با مروری در منابع موجود می‌توان عنوان‌های دیگری را برای هر یک از این بخش‌ها نظیر: سرویس نمایش، سرویس‌های عملیات و داده یافت (۶: ۱-۳)، اما در کل، محتوا و وظایف هر بخش به شرحی است که در ذیل به آن‌ها می‌پردازیم.



شکل ۱۰. مدل مرجع معماری سه لایه (۱۳)

30.Three Tier (client/server)

31.User services

32. Business services

33. Data services

در مدل معماری سه لایه، مسئولیت ارائه هر یک از سرویس‌های فوق به یک لایه واگذار می‌شود و هر لایه، سرویس‌های خاصی را ارائه می‌کند. بدین معنی که هر لایه، به عنوان مجموعه‌ای از سرویس‌ها در نظر گرفته می‌شود که مسئول ارائه سرویس به عناصر موجود در همان لایه و یا سایر لایه‌های مرتبط با آن است. عناصر موجود و مسئول ارائه سرویس و خدمات در هر لایه، قادر به ارتباط و درخواست سرویس از عناصر موجود در لایه خود و سایر لایه‌های موجود در سطح بالا و یا پایین خود هستند. ارتباط بین هر یک از این لایه‌ها، با استفاده از تفاهم‌نامه‌های خاصی نظیر تفاهم‌نامه‌ی اچ تی تی پی یا غیره انجام می‌گیرد. البته استفاده از تفاهم‌نامه‌ها، بستگی مستقیمی به نوع رابطه لایه‌ها با یکدیگر دارد.

نکته بسیار مهم در رابطه با وضعیت ارتباط لایه‌ها با یکدیگر، این است که یک درخواست اخذ سرویس نمی‌تواند یک لایه را حذف و خود مستقیماً، با لایه ثانویه (بعدی) ارتباط برقرار کند و اصطلاحاً، یک لایه را دور بزند. مثلاً عناصر موجود در لایه رابط کاربر نمی‌توانند مستقیماً، درخواست خود را به لایه داده ارسال کنند و البته لایه داده نیز چنین امکانی را نخواهد داشت. هر یک از سه بخش فوق، مسئولیت‌های خاصی را برعهده دارند و زمانی که یک بخش به خدمات بخش دیگر نیاز داشته باشد، درخواست خود را برای اخذ سرویس در اختیار بخش مورد نظر قرار می‌دهد و بخش فوق، سرویس درخواستی

را در قالب اجرای یک یا چند عنصر انجام و ماحصل را در اختیار بخش مربوطه قرار می‌دهد.

در معماری سه لایه، لایه رابط کاربر، مسئولیت ارائه رابط گرافیکی^{۳۴} را برعهده دارد. عناصر موجود در این لایه، مسئولیت ارتباط با سایر عناصر موجود در لایه خود و لایه پردازش را نیز برعهده دارند. عناصر موجود در لایه رابط کاربر، قادر به برقراری ارتباط مستقیم با عناصر لایه داده نیستند، بنابراین، جهت ارتباط با عناصر این لایه، ناگزیر به برقراری ارتباط با لایه پردازش هستند. لایه پردازش، مسئولیت انجام عملیات فوق برعهده می‌گیرد و ارتباط با لایه داده را برقرار می‌سازد. عناصر موجود در لایه پردازش، همچنین مسئول ارائه تمامی سیاست‌های راهبردی نرم‌افزار هستند. سیاست‌های راهبردی در هر نرم‌افزار از جمله مواردی است که بالاترین میزان تغییرات را در مدت زمان حیات یک نرم‌افزار خواهد داشت. با تغییر یک سیاست، بدیهی است که لازم است تغییرات دیگری نیز در نرم‌افزار اعمال شود. در چنین مواردی که لازم است تغییراتی در عناصر سرویس‌دهنده نرم‌افزار، جهت اعمال خواسته (سیاست) جدید صورت پذیرد، این تغییرات در لایه پردازش انجام می‌گیرد. لایه پردازش، علاوه بر این‌که، مستقیماً با لایه رابط کاربر ارتباط دارد، قادر به برقراری ارتباط مستقیم با لایه داده نیز هست. لایه داده، مسئول ذخیره و بازیابی داده‌ها در منابع داده‌ای است. البته در لایه داده، به هیچ

34. Graphical User Interface (GUI)

وجه از عناصر لایه پردازش استفاده نشده و عناصر این دو لایه، کاملاً از یکدیگر متمایز است، که این از جمله ویژگی‌های لایه داده یعنی استقلال در دستیابی به داده‌ها، صرف نظر از نوع منابع داده‌ای است. بنابراین، لایه پردازش، درخواست‌های کاربر را دریافت و پردازش‌های لازم را اجرا و وظیفه هر بخش را مشخص می‌کند و در نهایت، درخواست را به مرکزی که مسئولیت پاسخ‌گویی به آن درخواست را دارد، ارسال می‌کند و منتظر دریافت پاسخ می‌ماند. در اصل، این لایه را می‌توان مغز متفکر سیستم نامید. لایه پردازش، پس از دریافت پاسخ، مسئولیت پردازش یافته‌ها را و تبدیل آن‌ها به فرمی که مورد نیاز و پسند کاربر است برعهده می‌گیرد و در نهایت نتایج حاصله را در اختیار لایه رابط کاربر قرار می‌دهد (۱).

مدل فوق که توضیحاتی راجع به آن ارائه گردید، صرفاً یک مدل منطقی است و جهت انجام عملیات مربوط به طراحی و پیاده‌سازی نرم‌افزار، نمی‌تواند تصویر فیزیکی روشنی را از آنچه مورد نظر است ترسیم کند. لذا جهت پیاده‌سازی نرم‌افزار، لازم است مدل منطقی به مدل فیزیکی تبدیل شود. به منظور پیاده‌سازی نرم‌افزارهایی با معماری‌های سه لایه (سرویس‌دهنده / سرویس‌گیرنده) چهار مدل فیزیکی ارائه شده است:

- سرویس‌دهنده مجرد^{۳۵}
- سرویس‌دهنده پردازش^{۳۶}
- سرویس‌دهنده تراکنش^{۳۷}

• سرویس‌دهنده وب^{۳۸}

در مدل سرویس‌دهنده مجرد، سرویس‌دهنده شبکه محل نصب کلیه عناصر لایه داده و پردازش است. عناصر لایه رابط کاربر، بر روی سرویس‌گیرنده مستقر می‌شود. البته در بعضی مواقع، محل نصب و قرار گرفتن عناصر لایه‌های پردازش و رابط کاربر تغییر می‌کند. مثلاً اگر در یک نرم‌افزار خاص، لازم باشد سیاستی در رابطه با لایه رابط کاربر اعمال شود، عناصری از لایه پردازش که مجری اعمال این سیاست هستند، به صورت محلی در لایه رابط کاربر نصب و فعال می‌شوند. و یا در مواردی که رابط کاربر به گونه‌ای طراحی شده باشد که بتواند مورد استفاده چند نرم‌افزار دیگر نیز قرارگیرد، رابط کاربر باید در سرویس‌دهنده نصب شود.

در مدل سرویس‌دهنده پردازش (عملیاتی)، علاوه بر سرویس‌دهنده اصلی، از سرویس‌دهنده دیگری با نام سرویس‌دهنده عملیاتی استفاده می‌شود. سرویس‌دهنده عملیاتی به صورت یک رابط عمل می‌کند. بدین معنی که ارتباط تمامی سرویس‌گیرنده‌ها در ابتدا با سرویس‌دهنده عملیاتی برقرار می‌گردد. سرویس‌گیرنده‌ها خواسته خود را به سرویس‌گیرنده عملیاتی ارسال می‌کنند و این سرویس‌دهنده، با لایه داده ارتباط برقرار می‌کند (محل استقرار عناصر لایه داده، سرویس‌دهنده اصلی است). در واقع، تمامی عناصری که باید به اشتراک گذاشته شوند، بر روی سرویس‌دهنده عملیاتی مستقر می‌شوند (۱).

35. Single server

36. Business server(Application)

37. Transaction server

38. Web server

سرویس‌دهنده تراکنشی نیز یکی از مدل‌های فیزیکی معماری سه لایه است که در آن از چند سرویس‌دهنده استفاده می‌شود. عناصر لایه‌های داده، پردازش و رابط کاربر بر روی این سرویس‌دهنده‌ها قرار می‌گیرند. استقرار این عناصر بر روی سرویس‌دهنده‌ها به صورت پویاست. یعنی امکان جابه‌جایی آن‌ها بر روی سایر سرویس‌دهنده‌ها وجود دارد. قابلیت جابه‌جایی عناصر بر روی سرویس‌دهنده‌ها، باعث می‌شود تا با افزایش تعداد کاربران، سیستم با مشکل افزایش ترافیک و کاهش کارایی مواجه نشود. در این مدل، علاوه بر استفاده از چند سرویس‌دهنده، از یک سرویس‌دهنده با نام سرویس‌دهنده تراکنش نیز استفاده می‌شود. این سرویس‌دهنده، مسئولیت مدیریت و ذخیره‌سازی عناصری را برعهده دارد که جهت انجام یک فعالیت تراکنش لازم هستند. فعالیت تراکنش، فعالیتی شامل چندین عملیات سلسله‌مراتبی است که باید مجموعه عملیات، تماماً، با موفقیت اجرا شوند تا این فعالیت انجام گردد. همچنین این سرویس‌دهنده، مسئولیت‌های، ارسال درخواست پیغام به یکی از سرویس‌دهنده‌ها، اتمام اجرای یک برنامه و بررسی صحت عملکرد یک عنصر را نیز برعهده دارد (۱).

در مدل سرویس‌دهنده وب، علاوه بر سرویس‌دهنده اصلی، یک سرویس‌دهنده با نام سرویس‌دهنده وب نیز در شبکه وجود دارد. این سرویس‌دهنده، مسئولیت سرویس‌های وب را برعهده دارد. در این مدل، سرویس‌گیرنده‌ها جهت برقراری ارتباط (درخواست صفحات ایستا و پویا)، در

ابتدا، با سرویس‌دهنده وب ارتباط می‌یابند. سرویس‌گیرنده‌ها برای برقراری این ارتباط، لازم است که به نرم‌افزارهای ارتباطی، نظیر مرورگرهای وب مجهز باشند. به عبارت دیگر، کاربران (سرویس‌گیرنده‌ها) با استفاده از یک مرورگر وب، با سرویس‌دهنده وب مرتبط خواهند شد.

مرورگرهای وب، به عنوان ابزارهای ارتباط با سرویس‌دهنده وب محسوب می‌شوند که بسیار حائز اهمیت هستند. کاربران از طریق مرورگرهای وب، درخواست‌های خود را برای سرویس‌دهنده وب ارسال می‌کنند و سرویس‌دهنده وب - که مدیر پردازش درخواست‌های واصله است - پاسخ مربوطه را توسط این مرورگرها برای کاربران ارسال می‌کند.

لازم به ذکر است که استاندارد بودن نرم‌افزارهای مرورگر، از اهمیت خاصی برخوردار است. چرا که با استاندارد بودن این نرم‌افزارها، تمامی سرویس‌گیرنده‌ها از طریق یک ابزار واحد استاندارد، خواسته‌های خود را برای سرویس‌دهنده‌های وب مطرح می‌کنند. بدیهی است در چنین حالتی، پاسخ‌گویی به این درخواست‌ها، از طرف سرویس‌دهنده‌های وب، به مراتب ساده‌تر و با اطمینان خاطر بیش‌تری صورت می‌پذیرد.

در سه مدل گفته شده قبلی، می‌بایست تمامی سیاست‌های راهبردی نرم‌افزار (لایه پردازش) بر روی سرویس‌دهنده متمرکز می‌شد تا اعمال تغییرات به سادگی صورت پذیرد. اما در مدل سرویس‌دهنده وب، به دلیل اینکه سرویس‌دهنده وب این پتانسیل را

داراست که به صورت اتوماتیک عناصری را بر روی سرویس‌گیرنده مستقر کند، ضرورت تمرکز سیاست‌های راهبردی نرم‌افزار بر روی سرویس‌دهنده، از بین می‌رود. بدین ترتیب، اگر یک برنامه مبتنی بر وب در بستر اینترنت اجرا شود، می‌توان برخی از عناصر را برای سرویس‌گیرنده ارسال کرد تا به صورت محلی در رایانه سرویس‌گیرنده اجرا شوند. و اگر برنامه مبتنی بر وب در بستر اینترنت اجرا شود، می‌توان براساس توان کامپیوترهای سرویس‌گیرنده و سرویس‌دهنده و پهنای باند موجود، برخی از عناصر را بر روی سرویس‌دهنده و برخی دیگر را بر روی سرویس‌گیرنده اجرا کرد. به هر حال، سرویس‌دهنده وب قادر است به صورت اتوماتیک، اجزای مورد نیاز را بر روی سرویس‌گیرنده نصب کند. این ویژگی از یک طرف، ضرورت استقرار تمامی عناصر بر روی سرویس‌گیرنده‌ها را از بین می‌برد و از طرف دیگر، به سرویس‌گیرنده‌ها استقلال لازم را می‌دهد تا در صورت ضرورت بتوانند در اجرای برخی از عناصر سهیم باشند. در نرم‌افزار کتابخانه دیجیتال نیز که جزء برنامه‌های تحت وب است، غالباً از مدل سرویس‌دهنده وب، که راهکار فیزیکی مناسب برای پیاده‌سازی برنامه‌های تحت وب به شمار می‌رود، استفاده می‌شود.

نتیجه‌گیری

فلسفه وجودی کتابخانه، در هر زمان،

پاسخ به نیازهای اطلاعاتی کاربران بوده است. کتابخانه‌ها در عصر حاضر نیز از این اصل کلی مستثنی نیستند. در عصر حاضر، فن‌آوری‌های نوین اطلاعاتی و ارتباطی که به سرعت در حال ظهور و گسترش هستند، نیازهای اطلاعاتی خاصی را موجب شده‌اند که این نیازها جز از طریق بهره‌مندی از همین فن‌آوری‌های جدید، قابل پاسخ‌گویی نیستند. از این رو، کتابخانه‌ها نیز جهت حفظ رسالت خود و در راستای پاسخ‌گویی به نیازهای نوین کاربران، ناگزیر از بهره‌مندی از فن‌آوری‌های نوین هستند تا بتوانند پاسخ‌گوی خیل عظیم نیازهای کاربران در عصر حاضر باشند.

کتابخانه دیجیتال که کتابخانه‌ای است دربرگیرنده منابع^{۳۹} و خدمات^{۴۰} اطلاعاتی به صورت دیجیتال، همگام با ظهور و گسترش این فن‌آوری‌ها پا به عرصه وجود گذاشته است تا پاسخ‌گوی نیازهای نوین کاربران کتابخانه‌ها باشد. همان‌گونه که موجودیت هر کتابخانه‌ای در مرحله طراحی ساختمانی برای کتابخانه و متناسب با نوع فعالیت آن کتابخانه شکل می‌گیرد، کتابخانه دیجیتال نیز جهت ایجاد و راه‌اندازی باید مراحل خاصی را در زمینه طراحی طی کند تا بتواند پا به عرصه وجود گذارد.

با توجه به این که اساسی‌ترین مسئله در طراحی و ساخت هر سیستم بزرگ نرم‌افزاری، معماری^{۴۱} آن نرم‌افزار است، اهمیت پرداختن به مسئله معماری نرم‌افزار کتابخانه دیجیتال

۳۹. منابع دیجیتالی، منابعی هستند که با ابزارها و شبکه‌های دیجیتالی ذخیره، پردازش و بازیابی می‌شوند.

۴۰. خدمات دیجیتالی، خدماتی هستند که به صورت دیجیتال در شبکه‌های رایانه‌ای مبادله می‌شوند.

۴۱. معماری در یک نگاه کلی، نحوه گردآوری و اثر متقابل اجزا را بیان می‌کند.

به روشنی قابل درک است. یک معماری خوب می‌تواند این اطمینان را به وجود آورد که سیستم به درستی پاسخ‌گوی نیازهایی از قبیل کارآیی، قابلیت اطمینان، قابلیت جابه‌جایی، مقیاس‌پذیری و تعامل متقابل است (۱۸: ۲) و در مقابل، یک معماری بد می‌تواند سردرگمی و گرفتاری به وجود آورد.

معماری در طراحی نرم‌افزار کتابخانه دیجیتالمانند پل ارتباطی میان نیازمندی‌ها و پیاده‌سازی، نقشی کلیدی بازی می‌کند و با ارائه یک توصیف انتزاعی از سیستم، دستورالعملی قابل اجرا را در اختیار کل سیستم قرار می‌دهد تا به طراحان امکان دهد درباره توانایی سیستم در پاسخ‌گویی به نیازمندی‌های مشخص و ارائه نمونه اولیه برای ساخت و ترکیب سیستم تفکر کنند (۵).

از آنجایی که نقش متخصصان هر حوزه از دانش در امور مرتبط با آن حوزه، غیرقابل انکار است، نقش کتابداران را نیز به عنوان متخصصان حوزه علوم کتابداری و اطلاع‌رسانی در امور و فعالیت‌های خاص این حوزه از علوم، نمی‌توان نادیده گرفت. لذا، حضور فعال کتابداران در تمام مراحل ایجاد و راه‌اندازی، و اداره و پشتیبانی کتابخانه‌های دیجیتال امری لازم و ضروری است.

با توجه به بررسی‌های انجام گرفته در متن حاضر در زمینه انواع معماری‌ها و معرفی چند معماری به صورت نمونه، می‌توان اذعان داشت که به علت پیچیدگی عملکردهای کتابخانه دیجیتال، معماری چند لایه از جمله معماری‌های مناسب برای کتابخانه‌های دیجیتال است. در این معماری،

استفاده از محیط وب و قابلیت‌های آن بسیار حائز اهمیت است. لذا مدل سرویس‌دهنده وب از جمله مدل‌های فیزیکی مناسب برای پیاده‌سازی آن با توجه به قابلیت‌های ذکر شده به شمار می‌آید.

منابع

۱. «اصول و مفاهیم برنامه‌نویسی وب». ۲۰۰۶. [قابل دسترسی در]: www.srco.ir/tutorial/servicemodel.asp
۲. «تعریف معماری». ۲۰۰۶. [قابل دسترسی در]: www.behsad.com/iasem/IAINTRO.htm
۳. شاهزیدی، محمود. «بررسی سرویس‌ها و پروتکل‌های کتابخانه‌های دیجیتال و ارائه یک مدل برای کتابخانه دیجیتال توزیع شده». پایان‌نامه کارشناسی مهندسی کامپیوتر، دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، ۱۳۸۲.
۴. قراخانی بهار، اکبر. *مبانی طراحی سیستم: کاربرد اصول و مفاهیم مهندسی نرم افزار در تجزیه و تحلیل، طراحی و پیاده‌سازی سیستم‌های کاربردی*. تهران، امور فنی و هنری: سازمان پژوهش، ۱۳۷۴.
۵. گارلن، دیوید. «معماری نرم‌افزار: مسیر حرکت». ترجمه ساسان کمیلی‌زاده. ۱۳۸۴. [قابل دسترسی در]: www.systemgroup.net/fa/view/style.asp?p=5.3&=455. [20 Oct. 2007].
6. Alexandria digital library. [on-line]. Available: www.alexandria.ucsb.edu
7. Bogen, M. ... [et al]. "Requirements and architecture for administration and maintenance of digital object libraries".

H.; Boddie, Stefan J. "A distributed digital library architecture incorporating different index styles". *Proceedings of the Advances in Digital Libraries Conference*. Department of computer science, The University of Waikato, New Zealand, 1998. [on-line]. Available: <http://portal.acm.org/citation.cfm?id=785934&dl=acm&coll=&CFID=15151515&CFTOKEN=6184618>

16. Pandey, Richa. "Digital library architecture". In Bangalore, DRTC workshop on digital libraries: theory and practice. 2003. [on-line]. Available: http://drtc.isibang.ac.in/bitstream/1849/22/2/B_architecture_richa.pdf#search=%22Digital%20library%20architecture%22%20Pandey%22. [20 Oct. 2007].

17. Patel, D.; Ramsamy, N.; Marsden, G. "Customizing digital library for small screen devices". 2002. [on-line]. Available: www.cs.uct.ac.za/courses/CS400W/project2002/libpda/paper/paper.htm. [20 Oct. 2007].

18. Payette, Sandra; Staples, Thornton. "The Mellon Fedora project: digital library architecture meets XML and web services". 2004. [on-line]. Available: <http://www.fedora.info/documents/ecdl2002final.pdf>. [20 Oct. 2007].

19. Yang, Y. ... [et al]. "Mobile agents and the SARA digital library. IEEE

2001. [on-line]. Available: <http://www.archimuse.com/mw2001/papers/bogen/bogen.html>. [20Oct. 2007].

8. Bowman, C.M.; Danzig, P.B.; Hardy, D.R. *Harvest: a scalable, customizable, discovery and access system*. Colorado: Department of computer science, University of Colorado – Boulder, 1995.

9. Davis, J.R. ... [et al]. Dienst protocol specification. 2000. [on-line]. Available: www.cs.cornell.edu/cdlrg/dienst/protocols/DienstProtocol.htm

10. Davis, J. R; Krafft, D. B.; Lagoze, C. *Dienst: building a production technical report server, in advances in digital libraries*. U.S.A: Springer Verlag, 1995.

11. Dienst architecture summary description. 2000. [on-line]. Available: www.cs.cornell.edu/cdlrg/dienst/architecture/architecture.htm. [20 Oct. 2007].

12. "Dienst software summary description". 2000. [on-line]. Available: www.cs.cornell.edu/cdlrg/dienst/software/DienstSoftware.htm. [20 Oct. 2007].

13. "Digital library architecture". [on-line]. Available: www.wtec.org/loyola/digilibs/04_06.htm. [20 Oct. 2007].

14. Frew, James ... [et al]. "The Alexandria digital library architecture". Presented in ECDL'98, LNCS 1513, USA.

15. McNab, Rodger; Witten, Ian

advances in digital libraries". 2000.
[on-line]. Available: [http:// users. cs.cf.
ac.uk/David.W.Walker/DIGITAL_LIBS/
washington00.html](http://users.cs.cf.ac.uk/David.W.Walker/DIGITAL_LIBS/washington00.html). [20 Oct. 2007].

تاریخ دریافت: ۱۳۸۵/۷/۱۵

