

فصل ۱۴: طراحی معماری نرم افزار

معماری چیست؟

وقتی در مورد معماری یک ساختمان یا بنا صحبت می کنیم، نگرش های متفاوتی به ذهنمان می رسد. در ساده ترین سطح، شکل کلی ساختمان فیزیکی را در نظر می گیریم. اما در واقع معماری آن، بسیار پیچیده تر از اینهاست. این حالتی است که در آن اجزای مختلف ساختمانی به صورتی با هم ترکیب می شوند که یک کل سازمان یافته و منسجم را تشکیل می دهند. این شیوه ای است که با آن ساختمان، همراه با ساختمان های مجاور خود در محیط سازگاری می یابد و هماهنگ می شود. در این جا این ساختار سیستم تا حدی هدف مشخص شده خود را نشان داده و نیازهای صاحبش را برطرف می کند. در این جا حس زیباشناسی ساختمان یعنی تأثیر بصری ساختمان، و شیوه ترکیب بافت ها، رنگ ها و مواد برای خلق نما و محیط زنده درونی مطرح است. در این جا جزئیات کوچک مثل طراحی تسهیلات نور، نوع کف پوش و ... فهرستی بلند بالا را به وجود می آورند و نهایتاً، موضوع اصلی هنر است. در مورد ساختار نرم افزاری چه می توان گفت؟ Bass و همکارانش [BAS98] این عبارت دشوار و دیرفهم را به شکل زیر تعریف کرده اند:

معماری نرم افزار در یک برنامه یا سیستم محاسباتی عبارتست از ساختار یا ساختارهای سیستم که شامل اجزای نرم افزاری، مشخصه های مشهود برونی این اجزاء و ارتباطات میان آنها می باشد.

چرا معماری؟

- معماری، یک نرم افزار عملیاتی نیست. بلکه نمودی است که مهندس نرم افزار را قادر می سازد:
- (۱) میزان تأثیر طرح را در مرتفع نمودن نیازمندی های بیان شده، تحلیل کند.
 - (۲) معماری های جایگزین دیگر را در مرحله ای که تغییر طرح هنوز نسبتاً آسان است، بررسی کند.
 - (۳) خطرات مربوط به ساخت نرم افزار را کاهش دهد.

طراحی داده ها

طراحی داده ها، هم چون دیگر فعالیتهای مهندسی نرم افزار (که گاهی به آن معماری داده ها نیز می گویند) مدلی از داده ها و/ یا اطلاعات را در سطح بالایی از حالت انتزاعی، ایجاد می کند. سپس مدل داده ای به صورت بازنمایی خاص پیاده سازی درمی آید که می توان آن را به وسیله سیستم مبتنی بر کامپیوتر پردازش کرد. در بسیاری از برنامه های کاربردی نرم افزار، معماری داده ها تأثیر شگرفی بر معماری نرم افزار دارد که باید آن را پردازش کند.

- اشیاء داده ای را پالایش کرده و مجموعه ای از انتزاعات داده ای را توسعه می دهد.
- صفات اشیاء داده ای را به عنوان یک یا چند ساختمان داده ای پیاده سازی می کند.
- ساختمان های داده ای به منظور اطمینان از ارتباط های مناسب بازنگری می گردد.
- ساختمان داده ها تا حد ممکن ساده سازی می گردند.

طراحی مبتنی بر داده ها یک روش طراحی معماری نرم افزار است که به راحتی امکان گذر از مرحله تحلیل نیاز به طراحی توصیف ساختمان برنامه را می دهد.

مدل سازی داده، ساختار داده، پایگاه داده، و انبار داده ها

اشیای داده ای که در طول تحلیل نیازهای نرم افزاری تعریف شده اند، با استفاده از نمودارهای موجودیت/ رابطه و فرهنگ داده ها، مدلسازی می شوند. کار طراحی داده ها، عناصر مورد نیاز مدل نیازها را در سطح جزء نرم افزاری به ساختمان داده ها و وقتی لازم باشد معماری پایگاه داده ای را در سطح برنامه کاربردی، تبدیل می کند.

طراحی تفصیلی داده‌ها (در سطح اجزاء)

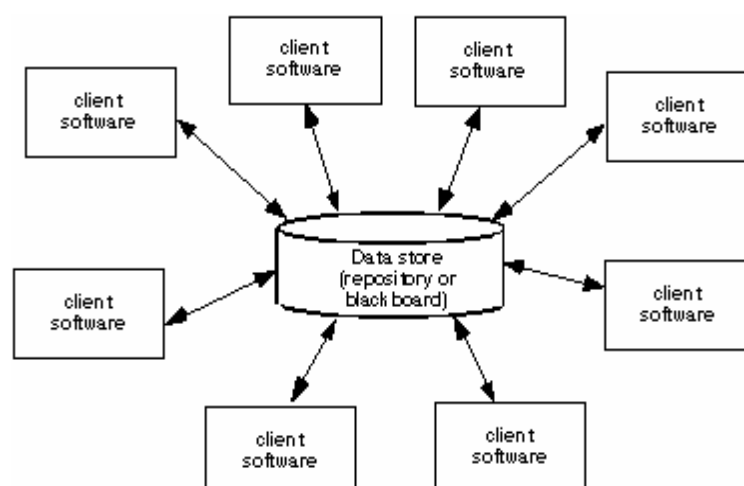
طراحی داده‌ها در این سطح روی آرایه و نمایش ساختمان داده‌هایی متمرکز می‌شود که مستقیماً توسط یک یا چند جزء نرم‌افزاری قابل دسترسی هستند. Wasserman [WAS80] مجموعه اصولی را پیشنهاد کرد که ممکن است برای مشخص کردن و طراحی چنین ساختارهایی به کار روند. در واقع، طراحی داده‌ها در طول خلق مدل تحلیل شروع می‌شود. فراخوانی تحلیل نیازها و طراحی اغلب هم‌پوشانی می‌شود. مجموعه اصول زیر را برای مشخصه‌های داده‌ای، در نظر بگیرید:

- ۱- اصول تحلیل نظام‌مند که در مورد عملکرد و رفتار به کار می‌رود باید در مورد داده‌ها نیز به کار رود.
- ۲- تمام عملیات و ساختمان داده‌ها باید شناسایی شوند، در طراحی یک ساختار داده‌ای کارآمد باید عملیاتی که روی ساختار داده‌ها صورت می‌گیرد را در نظر گرفت.
- ۳- یک فرهنگ داده‌ای به وجود آورده و از آن برای تعریف طراحی داده‌ها و برنامه استفاده کنید.
- ۴- تصمیمات مربوط به سطوح پایین طراحی داده‌ها را باید تا اواخر فرآیند طراحی به تعویق انداخت.
- ۵- نمودار ساختاری داده‌ها تنها باید برای پیمانه‌هایی شناخته شده باشد که مستقیماً از داده‌های موجود در ساختار استفاده می‌کنند.
- ۶- کتابخانه‌ای از ساختارهای داده‌ای مفید و عملیاتی که ممکن است در مورد آنها به کار رود، باید ایجاد گردد.
- ۷- طراحی نرم‌افزار و زبان برنامه‌نویسی باید خصوصیات و شناسایی انواع داده‌های انتزاعی را پشتیبانی کند.

سبک‌های معماری

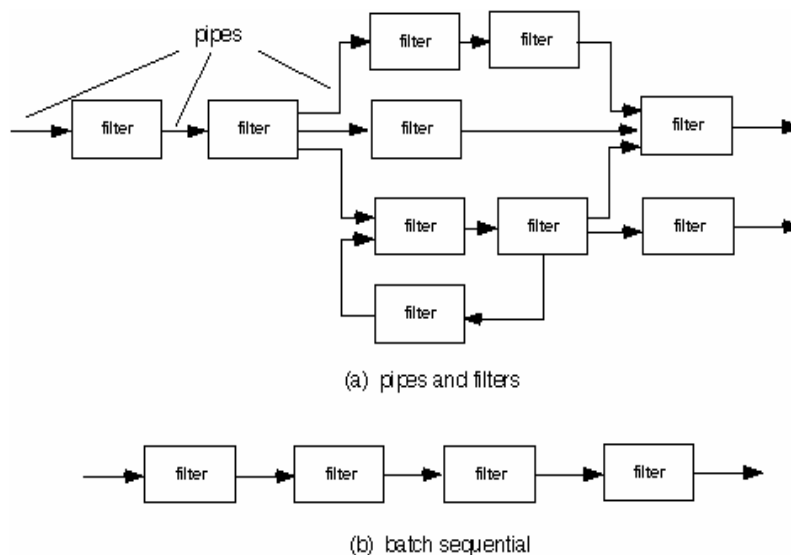
- o Data-centered architectures
- o Data flow architectures
- o Call and return architectures
- o Object-oriented architectures
- o Layered architectures

معماری‌های متمرکز بر داده‌ها - یک مخزن داده‌ای در مرکز این معماری قرار دارد و اغلب توسط دیگر اجزایی که به روزسازی، افزودن، حذف یا کارهای دیگر اصلاحی را در مورد مخزن انجام می‌دهند، قابل دسترسی است.



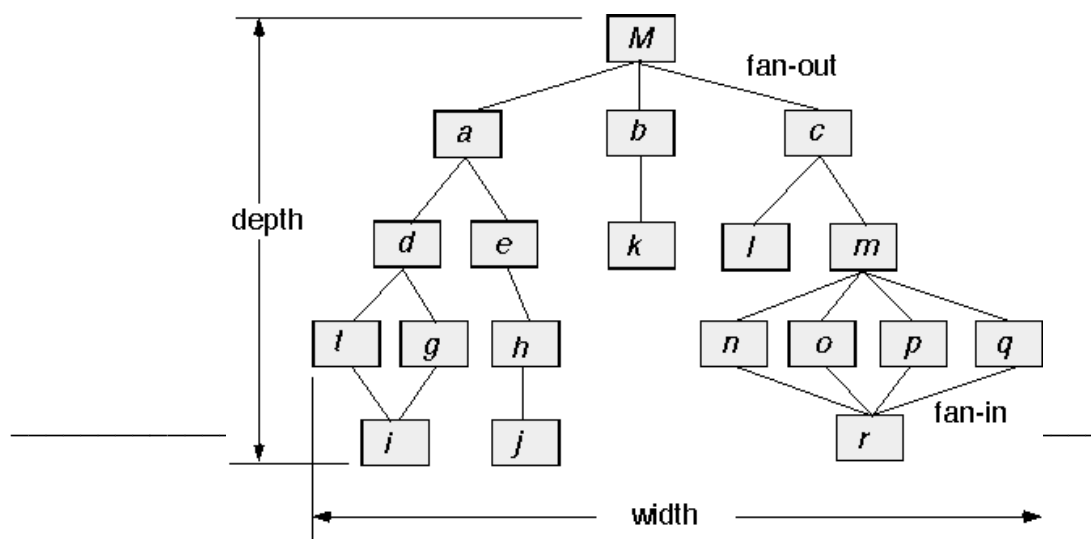
شکل ۱: نمودار معماری داده‌گرا

معماری‌های جریان داده‌ها. این معماری وقتی به کار گرفته می‌شود که داده‌های ورودی قرار است از طریق یک سری اجزاء محاسباتی یا تغییراتی، به داده‌های خروجی تبدیل شوند.



شکل ۲: نمودار معماری جریان داده

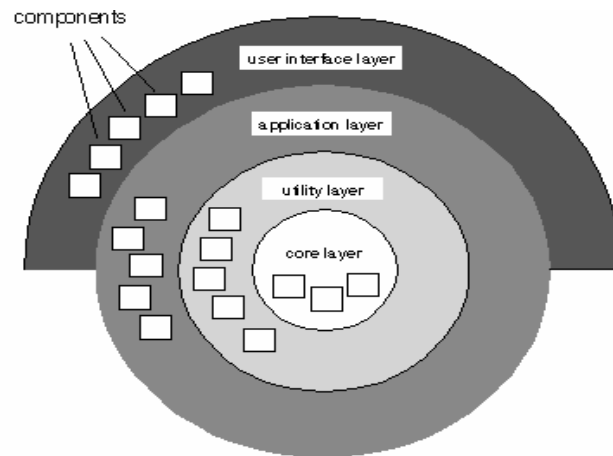
معماری فراخوانی و بازگشت- این سبک از معماری طراح نرم افزار (معمار سیستم) را قادر می سازد تا به ساختار برنامه ای دست یابد که از نظر اصلاح و ارزیابی نسبتاً ساده است.



شکل ۳: نمودار معماری فراخوانی و برگشت

معماری های شی گرا. اجزای یک سیستم در برگرفته داده ها و عملیاتی هستند که باید برای تغییر داده ها به کار روند. ارتباط و هماهنگی بین اجزا از طریق عبور پیام ها حاصل می شود (این معماری در فصول آتی مورد بحث قرار می گیرد).

معماری های لایه ای. ساختار اصلی این معماری در شکل زیر نشان داده شده است. تعدادی لایه های مختلف تعریف شده اند که هر کدام به عملیاتی دست می یابند که به طور گسترده ای به مجموعه دستورات ماشین نزدیک تر می شوند. در لایه خارجی تر، اجزاء در خدمت عملیات واسط کاربر هستند. در لایه داخلی تر، اجزاء کار ارتباط سیستم عامل را انجام می دهند. لایه های میانی خدمات استفاده و بهره برداری و عملیات کارکردی نرم افزار را مهیا می کنند.



شکل ۴: نمودار معماری لایه‌ای

روش‌های طراحی معماری (An Architectural Design Method)

فعالیت‌های تحلیل طراحی به صورت تکراری انجام می‌گیرند:

۱- جمع‌آوری (سناریوها) طرح‌ها. مجموعه‌ای از موارد مورد استفاده ارایه شده‌اند تا نمایانگر سیستم از دیدگاه کاربر باشند.

۲- بدست آوردن نیازها، محدودیت‌ها و توصیف محیط. این اطلاعات به عنوان بخشی از مهندسی نیازها، لازمند و برای اطمینان از این امر استفاده می‌شوند که تمام نگرانی‌های مشتری، کاربر و سهامداران مورد توجه قرار گرفته‌اند.

۳- توصیف سبک‌ها/ الگوهای معماری که برای بررسی طرح‌ها و نیازها انتخاب شده‌اند. این سبک‌ها را باید با استفاده از دیدگاه‌های معماری مورد توصیف قرار داد مثل:

- دیدگاه پیمانه
- دیدگاه پردازشی
- دیدگاه جریان داده‌ای

۴- ارزیابی صفات خاصه کیفی با در نظر گرفتن هر یک به طور جداگانه. تعداد صفات خاصه انتخابی برای تحلیل، تابعی از زمان موجود برای بازبینی و مقداری است که نسبت به آن این نگرش‌ها با سیستم مورد نظر مرتبط می‌شوند. نگرش‌های کیفی برای برآورد طراحی معماری شامل قابلیت اطمینان، عملکرد، امنیت، قابلیت نگهداری، انعطاف‌پذیری، آزمون‌پذیری، قابلیت حمل، استفاده مجدد و قابلیت عملیاتی در همکاری می‌باشد.

۵- شناسایی حساسیت صفات خاصه به صفات مختلف معماری برای یک سبک به خصوص. این کار با ایجاد تغییرات کوچکی در ساختار و تعیین چگونگی حساسیت یک صفت خاصه کیفی مثلاً عملکرد نسبت به تغییر، انجام می‌گیرد. صفت خاصه‌ای که تا حد زیادی تحت تأثیر تنوع معماری قرار دارد، " نقاط حساسیت " نامیده می‌شود.

۶- متخصصان با استفاده از تحلیل حساسیت که در مرحله ۵ صورت گرفته، معماری‌های کاندیدا را پیشنهاد می‌کنند.

پیچیدگی معماری

یک روش مفید برای ارزیابی پیچیدگی کلی معماری پیشنهادی، عبارتست از در نظر گرفتن وابستگی میان اجزاء درون معماری. این وابستگی‌ها ناشی از جریان کنترل/اطلاعات درون سیستم می‌باشند.

ZHAO سه نوع وابستگی ارائه می‌دهد: [ZHA98]

- ♦ **وابستگی‌های مشترک** نمایانگر ارتباطات وابسته‌ای در میان مصرف‌کنندگانی هستند که از منبعی یکسان استفاده کرده یا تولیدکنندگانی که برای مصرف‌کنندگانی یکسان تولید می‌کنند. مثلاً، در مورد دو جزء V, U اگر U و V هر دو به یک سری داده سرتاسری یکسانی رجوع کنند، یک واسطه وابستگی مشترک بین V, U وجود دارد.
 - ♦ **وابستگی‌های جریان** نمایانگر ارتباطات وابستگی میان تولیدکننده و مصرف‌کننده‌های منبع است. در مورد دو جزء V, U ، اگر باید U قبل از جریان یافتن کنترل در V تکمیل شود یا اگر U به وسیله پارامترهایی با V ارتباط برقرار می‌کند، در این صورت یک جریان وابستگی میان این دو وجود دارد.
 - ♦ **وابستگی‌های محدود شده** نمایانگر محدودیت‌ها و قیودی در جریان نسبی کنترل میان مجموعه‌ای از فعالیت‌ها هستند. مثلاً در مورد دو جزء V, U ، آنها نمی‌توانند در یک زمان اجرا شوند پس یک واسطه وابستگی محدود و مقید شده بین آنها وجود دارد.
- وابستگی‌های مشترک و جریان که مورد توجه ZHAO قرار گرفت از بعضی جهات شبیه مفهوم انسجام و اتصال هستند.

نگاشت نیازها در یک معماری نرم افزار

انتقال از جریان داده‌ها به ساختمان نرم افزار با طی ۶ مرحله انجام می‌گیرد. طراحی ساختیافته اغلب به عنوان یک روش طراحی مبتنی بر جریان داده‌ای معرفی می‌گردد زیرا انتقال راحت‌تر از نمودار جریان داده‌ای (DFD) را به معماری نرم افزار فراهم می‌سازد. کار انتقال از جریان اطلاعات به ساختار برنامه به عنوان بخشی از فرآیند مرحله ۶ با موفقیت به انجام می‌رسد:

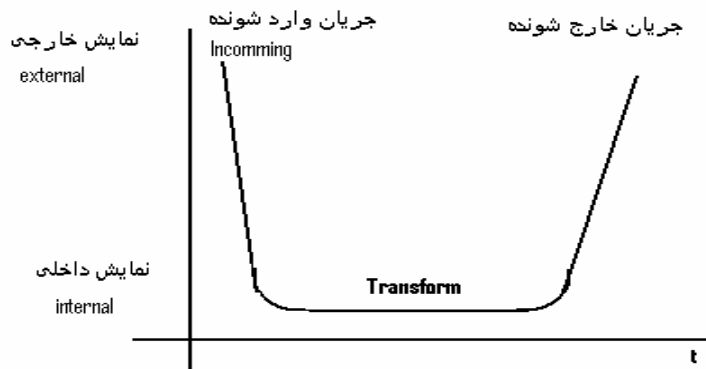
- (۱) نوع جریان اطلاعاتی تعیین می‌گردد.
- (۲) سرحدات جریان مشخص می‌شوند.
- (۳) DFD در ساختمان برنامه نگاشت می‌شود.
- (۴) سلسله مراتب کنترلی تعیین می‌گردد.
- (۵) ساختمان بدست آمده با استفاده از معیارهای طراحی و علوم تجربی بازنگری می‌شود.
- (۶) توصیف معماری بازنگری و تعیین می‌گردد.

تعیین نوع جریان اطلاعات

دو نوع جریان اطلاعات وجود دارد:

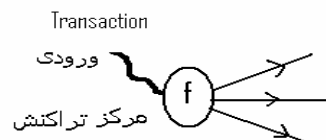
- ۱- جریان تبدیلی (Transform Flow)،
- ۲- جریان تراکنشی (Transaction Flow)

در جریان تبدیلی روال به صورت زیر است:



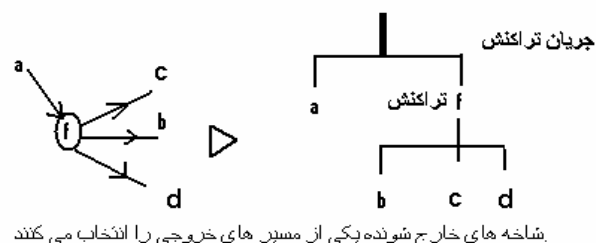
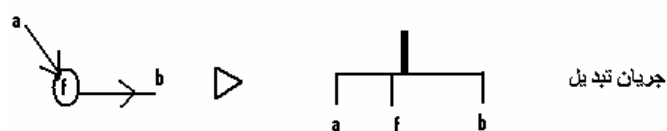
شکل ۵: نمودار جریان تبدیلی

در روش تراکنشی، جریانی وارد سیستم می شود و سیستم انتخاب می کند که از کدام شاخه خارج شود.



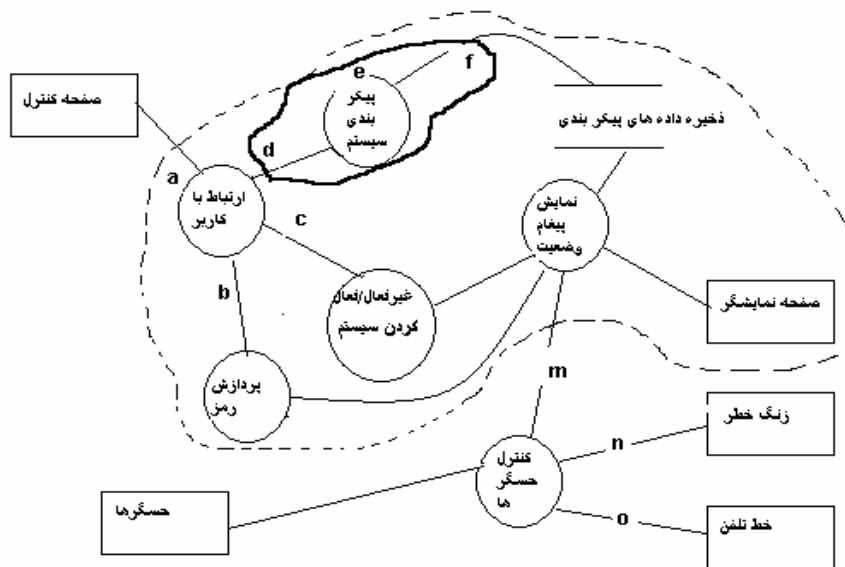
شکل ۶: نمودار جریان تراکنشی

مثال: تبدیل جریان داده به معماری نرم افزار:



شکل ۷: نحوه تبدیل جریان های تبدیلی و تراکنشی به معماری (ساختار) برنامه

مثال: سطح ۱ DFD خانه امن را در نظر بگیرید. داریم:

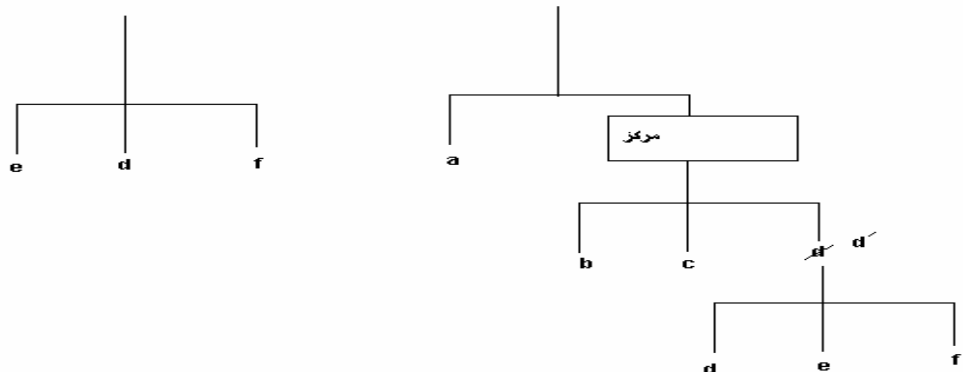


شکل ۸: نمودار جریان داده سیستم خانه امن

حال با توجه به DFD فوق مراحل طراحی معماری را پیش می گیریم.

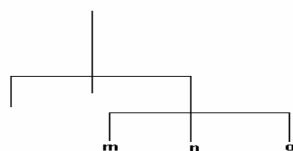
مرحله ۲۱: تعیین مرز و نوع جریان:

در ابتدا قسمت خط چین در شکل را در نظر می گیریم. به نمونه مثال زیر توجه کنید.



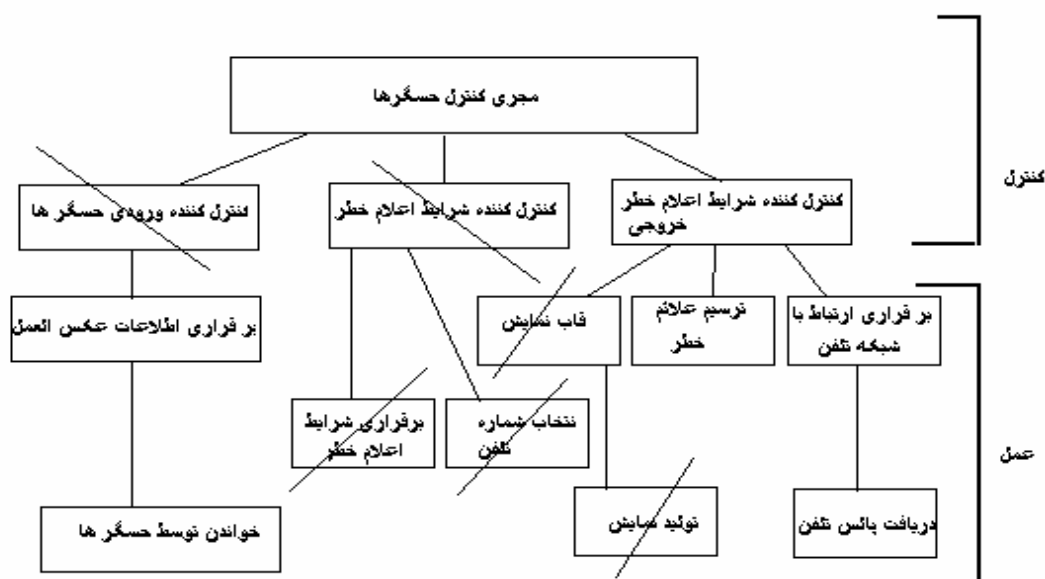
شکل ۹: مثالی از نحوه نگاشت جریان داده به ساختار برنامه

در سطوح بالایی معماری کنترل مطرح است ولی در سطوح پایینی نحوه تبدیل اطلاعات را می توان دید. و یا قسمت کنترل حسگر یک تبدیل است چون در تراکش یکی از خروجی ها انتخاب می شود ولی در کنترل حسگر هر سه خروجی همزمان اتفاق می افتد.



شکل ۱۰: مثالی از نحوه نگاشت جریان تراکنشی به ساختار برنامه

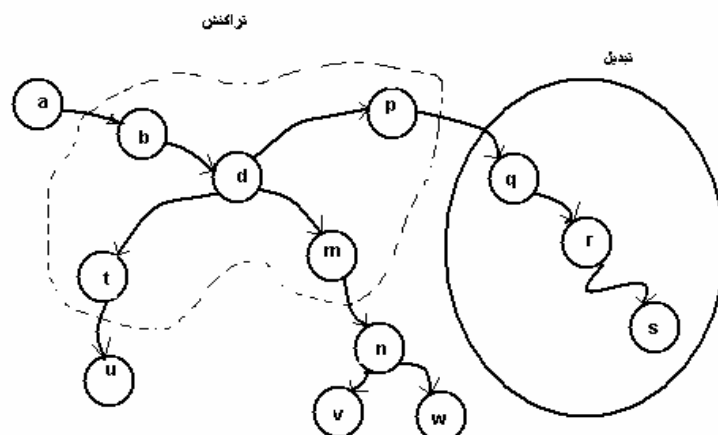
اگر مراحل ۳ تا ۵ که عبارت است از رسم تفصیلی و تعیین جریان و مرز آن را انجام دهیم در مرحله ۶ خواهیم داشت:



شکل ۱۱: مثالی از نحوه نگاشت جریان تراکنشی و تبدیلی سیستم خانه امن به ساختار

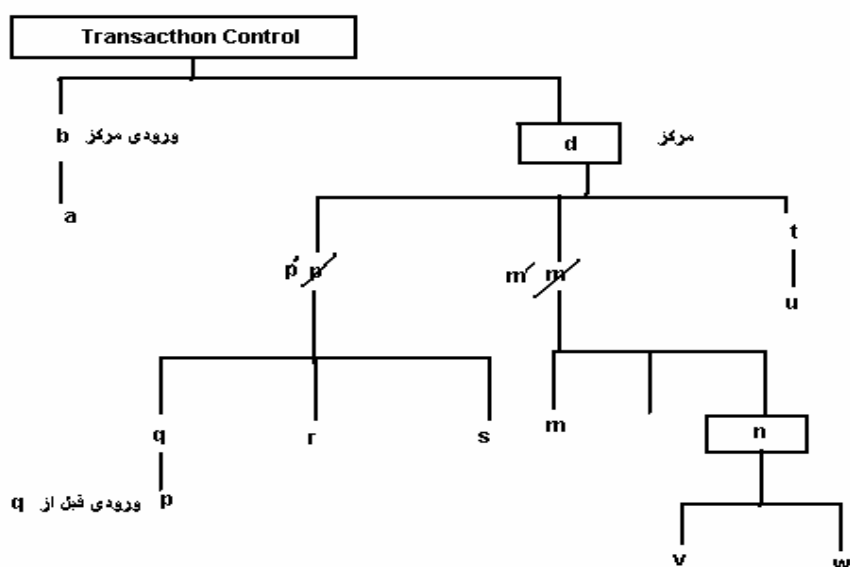
باز نگری حاصل ممکن است نتیجه را تعدیل کند(می توان این عمل را در مرحله دیگری مثل مرحله ۷ انجام داد). به عنوان مثال می توان بخش "قالب نمایش" و "تولید نمایش" را در یک مرحله با عنوانی جدید مثل "پردازش نمایش اطلاعات انجام داد". ... در این مرحله بازنگری انجام می گیرد و نیز پالایش ساختار اولیه نرم افزار به منظور بهبود کیفیت نرم افزار انجام می گیرد. در بازنگری باید اصول و ضوابط طراحی، درک، مهارت، تجربه و قواعد بازنگری دخالت داشته باشند.

مثال: معماری متناظر با DFD داده شده را به دست آورید.



شکل ۱۲: مثالی از جریان های تراکنشی و تبدیلی

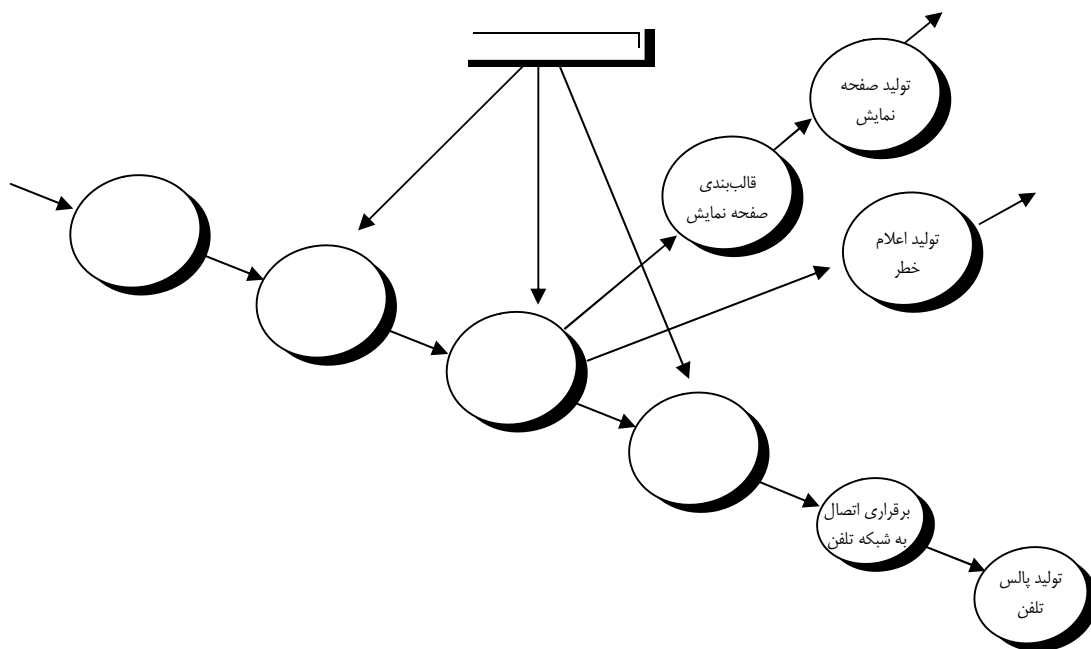
حل: از روی شکل در می یابیم که DFD ما تراکنشی است مگر آنکه شرحی روی آن نوشته شده باشد مبنی بر اینکه هر سه خروجی همزمان اتفاق می افتند در این صورت می گوییم جریان تبدیلی است.



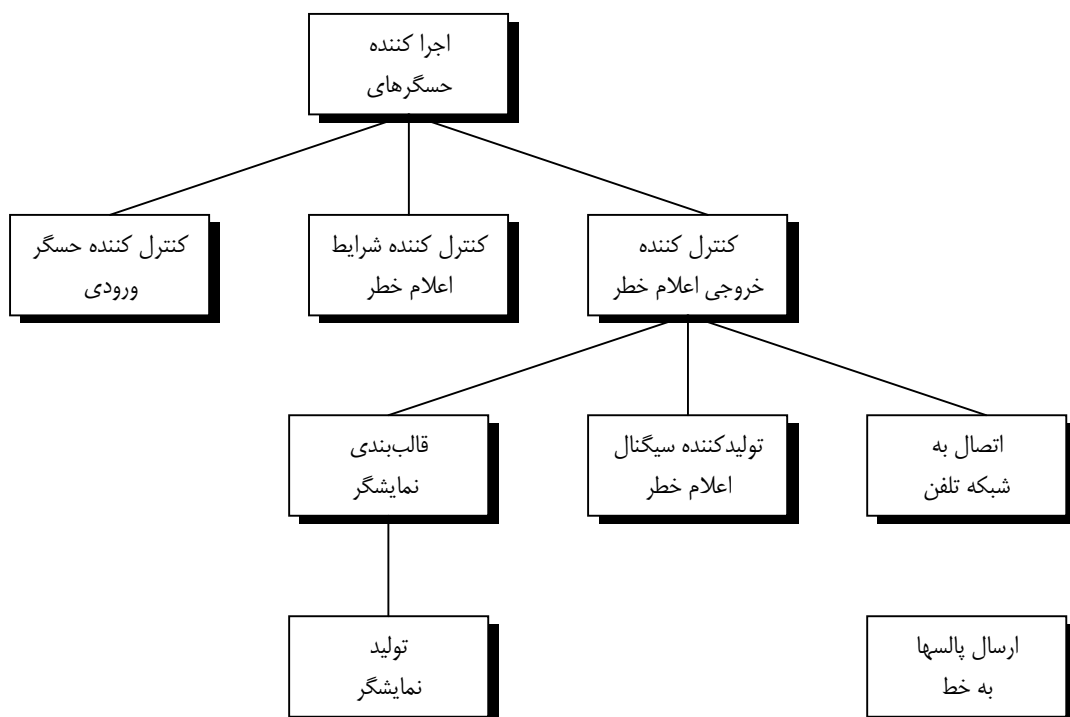
شکل ۱۳: نگاشت جریان ها به ساختار برنامه

مثال: معماری برنامه کتناظر با سیستم خانه امن را با استفاده از DFD های داده شده به دست آورید.

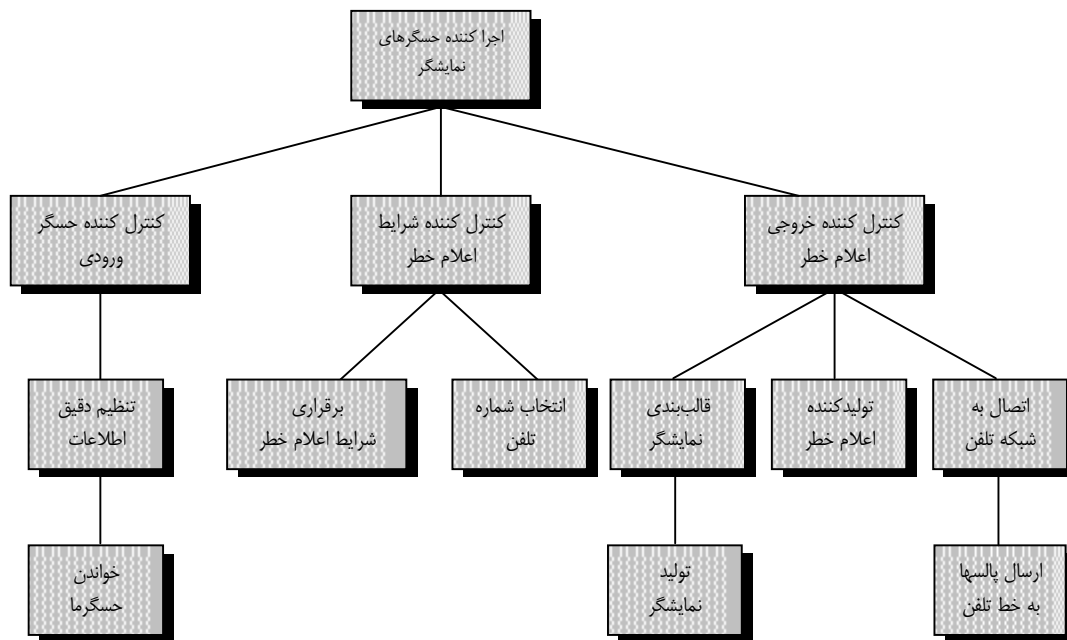
حل: بخشی از جریان داده ها سیستم خانه امن در سطح یک در صفحه بعد نشان داده شده است.



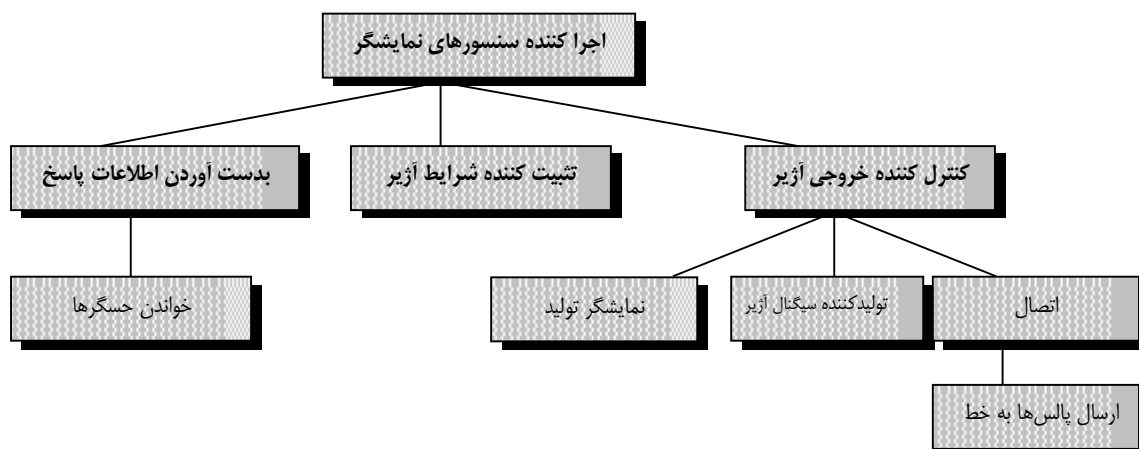
شکل ۱۴: نمودار جریان داده‌های سطح ۱ برای حسگرهای خانه امن



شکل ۱۵: نمودار ساختار برنامه برای حسگرهای نمایش دهنده



شکل ۱۶: ساختار قبل از پالایش برای حسگرهای نمایش دهنده



شکل ۱۷: ساختار پالایش شده برنامه برای حسگرهای نمایش دهنده

استفاده موفق از نگاشت تراکنش یا تبدیل با کارهای اضافی دیگری تکمیل می شود که به عنوان بخشی از طراحی معماری ضروری هستند. بعد از ساخت ساختار برنامه و اصلاح آن، کارهای زیر باید تکمیل گردد:

- ♦ گزارش از پردازش های هر پیمانه ارایه شود.
- ♦ یک توضیح در مورد واسط برای هر پیمانه داده شود.
- ♦ ساختارهای داده ای محلی و سراسری تعریف شوند.
- ♦ تمام محدودیت های طراحی ذکر شوند.
- ♦ مجموعه ای از بازنگری طراحی ها آماده شود.
- ♦ اصلاح در صورت لزوم در نظر گرفته شود.

تست های فصل چهارده: طراحی معماری

- ۱- کدامیک از گزینه های زیر قسمتی از معماری نرم افزار نیست؟
 (الف) جزئیات الگوریتم (ب) پایگاه داده ها (ج) طراحی داده ها (د) ساختمان برنامه
- ۲- طراحی داده در واقع در ضمن تحلیل مدل معماری نرم افزار بوجود می آید.
 (الف) درست (ب) غلط
- ۳- معماری که برای تشخیص کیفیت یک طراحی معماری استفاده می شود باید بر مبنای چه ویژگی های از سیستم باشد؟
 (الف) کنترل و داده (ب) تابعی بودن (ج) جزئیات پیاده سازی (د) واسطها
- ۴- یک تکنیک مفید برای ارزشیابی پیچیدگی کلی یک معماری پیشنهادی با نگاه به مولفه ها چیست؟
 (الف) تعداد و اندازه مولفه ها (ب) جریان وابستگی ها و تقسیم وابستگی ها (ج) اندازه و هزینه (د) هیچکدام
- ۵- هدف پالایش DFD در ضمن نگاشت آن به معماری نرم افزار کوشش برای دستیابی به دایره هایی است که وابستگی زیاد را نشان دهند.
 (الف) درست (ب) غلط
- ۶- وقتی با هر دو جریان تبدیل و تراکنش در یک DFD مواجه می شویم آنرا تقسیم بندی می کنیم و تکنیک نگاشت مناسب را برای هر قسمت از DFD به کار می بریم.
 (الف) درست (ب) غلط
- ۷- در پالایش DFD ضمن نگاشت تراکنش ضروریست که یک PSPEC تا وقتی که تنها یک CSPEC در رابطه با نوع روش معماری نرم افزار است، بوجود آوریم.
 (الف) درست (ب) غلط
- ۸- در نگاشت تراکنش اولین سطح، تولید نتایج در کدامیک از موارد زیر حاصل می گردد؟
 (الف) بوجود آوردن یک CFD (ب) بدست آوردن سلسله مراتب کنترل (ج) توزیع ماژولهای کاری (د) پالایش دیدگاه ماژول
- ۹- کدامیک از موارد زیر در ارتباط با انبار داده ها درست ولی در مورد پایگاه داده ها درست نیست.
 (الف) مبتنی بر سطح کسب و کار و اندازه پروژه (ب) دقت و صحت اطلاعات (ج) یکپارچگی و پایداری (د) تمام موارد فوق
- ۱۰- سبک های معماری کدامیک از مؤلفه های زیر را در خود جای داده است؟
 (الف) محدودیت ها (ب) مجموعه ای از مولفه ها (ج) مدل های معنایی (Semantic) (د) تمام موارد فوق
- ۱۱- برای تعیین سبک معماری یا ترکیبی از سبک ها که مناسب ترین حالت برای سیستم پیشنهادی باشد مهندسی نیازها باید برای کشف کدام یک از موارد زیر انجام گیرد:
 (الف) پیچیدگی الگو (ب) ویژگی ها و محدودیت ها (ج) کنترل و داده (د) الگوهای طراحی
- ۱۲- معیارهای ارزیابی کیفیت طراحی معماری باید مبتنی بر سیستم باشد؟
 (الف) قابلیت دسترسی و قابلیت اعتماد (ب) داده و کنترل (ج) قابلیت کارکردها (د) جزئیات پیاده سازی
- ۱۳- وقتی یک جریان کلی در قسمتی از نمودار جریان داده، خیلی طولانی و پشت سرهم باشد این امر نشان دهنده است.
 (الف) اتصال پایین (ب) پیمانه پذیری خوب (ج) جریان تراکنشی (Transaction) (د) جریان تبدیلی (Transform)
- ۱۴- وقتی جریان اطلاعات در قسمتی از یک نمودار جریان داده ها توسط ویژگی یک مؤلفه تنها شناسایی می شود که سایر جریان داده ها در طول یک یا چند مسیر است، این امر نشان دهنده است.
 (الف) اتصال بالا (ب) پیمانه پذیری ضعیف (ج) جریان تراکنشی (د) جریان تبدیلی
- ۱۵- در طی فرآیند نگاشت وقتی DFD مورد پالایش قرار می گیرد، هدف آن است که به حباب هایی دست یابیم که دارای انسجام (cohesion) بالا هستند.
 (الف) درست (ب) نادرست