فصل ۱۲ مدل سازی تحلیل

مفاهیم کلیدی (مرتب بر حروف الفبا)

تجزیه گرامری ، تعیین مشخصه های کنترل (CSPEC) ، فرهنگ داده ها ، گسترش زمان واقعی (real time) ، مدل تحلیل ، مدل جریان کنترل ، مدل سازی داده ها ، مدل سازی رفتاری ، مدل سازی کارکردی ، مشخص سازی فرآیندها (PSPEC) ، مکانیک های تحیل ساختاری ، نمودارهای جریان داده ها (DFD) ، نمودارهای موجودیت – رابطه (ERD)

KEY CONCEPTS
analysis model , behavioral modeling , control flow model , CSPECs , data dictionary , DFDs , data modeling , ERDs , functional modeling , PSPECs , grammatical parse , real-time extensions , structured analysis mechanics

نگاه اجمالی

مدل‌سازی تحلیلی چیست؟ ساختار نوشتاری معمولاً بهترین وسیله برای برقراری ارتباط می‌باشد. اما این وسیله ضرورتاً نمی‌تواند بهترین روش برای نشان دادن نیازمندیهای نرم‌افزار کامپیوتری باشد. مدل‌سازی تحلیلی برای نشان دادن نیازمندیهای داده‌ها، توابع- کارکرد و رفتار ترکیبی از متن و اشکال نموداری را به‌گونه‌ای استفاده می‌کند که درک آن ساده است و مهم‌تر آن‌که بررسی صحت، کامل و سازگار بودن آن بسیار ساده و آسان می‌باشد.

چه کسی کار مدل‌سازی تحلیلی را انجام می‌دهد؟ یک مهندس نرم‌افزار (که در برخی موارد تحلیل‌گر نیز نامیده می‌شوند) با استفاده از نیازمندیهای ذکر شده از سوی مشتری به ساخت مدل مورد نیاز اقدام می‌کند.

چرا کار مدل‌سازی تحلیلی از اهمیت به‌سزایی برخوردار است؟ برای تأیید اعتبار نیازمندیهای نرم‌افزاری، شما باید این نیازمندیها را از نقطه نظرهای مختلف مورد بررسی قرار دهید. مدل‌سازی تحلیلی این نیازمندیها را به‌صورت سه‌بعدی نمایش می‌دهد و بدین‌ترتیب احتمال یافتن خطاهای موجود را افزایش می‌دهد و احتمال ظاهر شدن ناسازگاری و مشخص شدن موارد از قلم افتاده را افزایش می‌دهد.

مراحل مدل‌سازی تحلیلی کدامند؟ نیازمندیهای داده‌ای، کارکردی و رفتاری با استفاده از تعدادی از شکلهای مختلف نموداری مدل‌سازی می‌شوند. مدل‌سازی داده‌ها می‌تواند اشیاء داده‌ای، ویژگی‌ها و روابط را مشخص کند. مدل‌سازی کارکردی، نحوهٔ تبدیل داده‌ها در درون یک سیستم را نشان می‌دهد. مدل‌سازی رفتاری نیز تأثیر رویدادهای مختلف را نشان می‌دهد. پس از ایجاد مدل‌های اولیه، این مدل‌ها مورد پالایش

قرار می‌گیرند و برای ارزیابی کامل بودن و سازگار بودن و صحیح بودن باید این مدل‌ها مجدداً مورد تجزیه و تحلیل قرار گیرند. بعد از این مرحله مشخصهٔ تشکیل‌دهندهٔ مدل موردنظر ایجاد می‌شود و سپس مورد تأیید مهندسین نرم‌افزار و مشتریان/ کاربران قرار می‌گیرد.

نتیجهٔ حاصل از کار مدل‌سازی تحلیلی چیست؟ توضیحات مربوط به اشیاء داده‌ای، نمودار رابطه‌های رابطه-موجودیت نمودار جریان داده‌ها، نمودارهای‌انتقال وضعیت، مشخصه‌های پردازشی و مشخصه‌های کنترل به‌عنوان بخشی از فعالیت مدل‌سازی تحلیلی ایجاد می‌شوند.

چطور می‌توان مطمئن شد که کار مدل‌سازی تحلیلی درست انجام شده است؟ برای مطمئن شدن از صحت کار مدل‌سازی تحلیلی باید محصولات (نتایج) کار نمونه‌سازی تحلیلی از لحاظ صحت، کامل و سازگار بودن مجدداً مورد بررسی قرار داده شوند.

مهندسی نرم‌افزار در سطح فنی با مجموعه‌ای از وظائف مدل‌سازی آغاز می‌شود که این وظائف مشخص‌سازی کامل نیازمندیها و بازنمایی طراحی جامع نرم‌افزار موردنظر برای ساخت، منتهی می‌شود. مدل تحلیلی[1]، و در واقع مجموعه‌ای از مدل‌ها، اولین بازنمایی فنی یک سیستم می‌باشند. در طول این سال‌ها شیوه های بسیار متفاوتی برای مدل‌سازی تحلیلی پیشنهاد شده است. اما با وجود این تعداد زیاد فقط دو عدد از این مدل‌ها غالب می‌باشند. اولین مدل غالب و مشهور، یعنی تحلیل ساخت یافته[2] یک شیوه مدل‌سازی قدیمی است و در این فصل توضیحات مفصلی در ارتباط با این شیوه ارائه خواهد شد. شیوه دوم، یعنی تحلیل شی‌گرا[3] به صورت کامل در فصل ۲۱ این کتاب مورد بررسی قرار خواهد گرفت. سایر شیوه های تحلیلی متداول نیز در بخش ۱۲-۸ این فصل به‌طور خلاصه و به اجمال بررسی خواهند شد.

تحلیل ساخت یافته فعالیتی است که در راستای ساختن مدل انجام می‌شود. با اعمال اصول تحلیل عملیاتی که در فصل یازدهم این کتاب بررسی خواهد شد، ما داده‌ها، مدل‌های رفتاری و کارکردی که نشانگر ضرورت موارد مورد نیاز برای ایجاد تحلیل ساخت یافته هستند را، ایجاد نموده و تقسیم‌بندی کرده‌ایم. این داده‌ها و مدل‌های رفتاری و کارکردی نشان می‌دهند که برای ایجاد تحلیل ساخت یافته از یک شیوه واحد برای تمام موارد کاربرد استفاده نمی‌شود، بلکه این روش تحلیلی (تحلیل ساخت یافته) ترکیبی از شیوه های مختلف است که در طول بیش از ۳۰ سال به‌وجود آمده‌اند.

آقای تام دماکرو [DEM79][4] در کتاب اصلی (بدوی) خود در ارتباط با موضوع مدل‌سازی تحلیلی، تحلیل ساخت یافته را به‌صورت زیر توصیف کرده است:

1.analysis model

2.structured analysis

3.object oriented analysis

4.DeMarco ,T.

"با توجه به مشکلات و معایب شناخته شده در مرحلهٔ تحلیل مشخص می‌شود که ما باید بخش‌های زیر را به مجموعهٔ اهداف تحلیلی خود اضافه کنیم."

- *نتایج تحلیل باید قابل تغییر و نگهداری باشند. این مسأله به‌ویژه در مورد مشخصات نرم‌افزاری سند مقصد (مشخصه‌های نیازمندیهای نرم‌افزار) اعمال می‌شود.*
- *مشکلات مربوط به اندازه باید با استفاده از شیوه مؤثر و مناسب تقسیم‌بندی، برطرف شوند.*
- *در صورت امکان و هر کجا می‌شود باید از علائم گرافیکی استفاده شود.*
- *ما باید مفروضات منطقی [ضروری] و فیزیکی [پیاده‌سازی] را از یکدیگر متمایز کنیم و در نهایت به موارد زیر نیاز خواهیم داشت ...*
- *هر چیزی که بتواند ما را در تقسیم‌بندی نیازمندیها و اسناد که قبل از به‌وجود آمدن مشخصه تفکیک شده است، یاری دهد.*
- *وسایلی که بتوانند ردگیری واسطه‌ها را حفظ کنند و آنها را مورد ارزیابی قرار دهند.*
- *وسایل جدید برای توصیف منطقی و توصیف سیاست موجود، یعنی استفاده از وسایلی که بتوانند بهتر از متن روایتی* عمل کنند.*

شاید هنوز شیوه ای در زمینهٔ مهندسی نرم‌افزاری وجود نداشته باشد که به‌اندازهٔ شیوه مدل‌سازی بتواند موقعیت هایی را برای نقد و به چالش کشیدن موضوع، از طرف افراد مختلف ایجاد نماید. مباحثه، بحث و جدال های مختلفی بوجود آمده، اما این شیوه در جامعهٔ مهندسی نرم‌افزاری شکوفا شده و پیشرفت کرده است.

۱۲-۱ تاریخچه ای مختصر

تحلیل ساخت یافته نیز همانند بسیاری از شیوه های مهم در زمینهٔ مهندسی نرم‌افزار فقط با استفاده از یک مقاله تاریخی و یا فقط با استفاده از یک کتاب معرفی نشده است. کارهای ابتدایی و اولیه در زمینهٔ مدل‌سازی تحلیلی در اواخر دههٔ ۱۹۶۰ و اوایل دههٔ ۱۹۷۰ آغاز شده است، اما ظهور اولیهٔ رهیافت تحلیل ساخت یافته مقارن با یک موضوع مهم دیگر ـ یعنی "طراحی ساخت یافته" ـ بوده است. محققان (برای مثال [STE74][1] و [YOU78][2]) برای نمایش داده‌ها و فرآیند تغییر دهنده و تبدیل‌کنندهٔ این داده‌ها به نشان‌گذاری گرافیکی نیاز دارند. این روش‌های پردازشی در نهایت به ساختار طراحی تبدیل می‌شوند.

* شیوه‌ای غیررسمی برای توضیح فرآیند- م.

1. Stevens ,W.P.
2. Yourdon , E.N.

واژهٔ "تحلیل ساخت یافته"[1] در اصل توسط داگلاس راس[2] ابداع شده است و توسط دمارکو [DEM79][3] مشهور شده است. دمارکو در کتاب خود، نمادهای اصلی گرافیکی و مدل‌های مربوط به آنها را معرفی و نام‌گذاری کرده است. در سال‌های بعدی پیشنهادات مربوط به ایجاد تغییرات در روش تحلیل ساخت یافته از سوی محققانی هم‌چون پیج جونز [PAG80][4]، گین و سارسون [GAN82][5] و بسیاری دیگری از محققان مطرح شده است. در هر موردی این شیوه بر کاربرد سیستم‌های اطلاعاتی تأکید دارد و نشان‌گذاری کافی برای ارزیابی جنبه‌های کنترل و رفتاری مسائل بلادرنگ مهندسی را فراهم می‌کند.

اواسط دههٔ ۱۹۸۰ مسأله "توسعه" بلادرنگ از سوی وارد و ملور[WAR85][6] و در سال‌های بعدی توسط هتلی و پیربهای [HAT87] مطرح شد. این نوع توسعه موجب به‌وجود آمدن شیوه تحلیلی نیرومند و مؤثری شد که این شیوه می‌توانست برای برطرف ساختن مسائل مهندسی بسیار مؤثر واقع شود. انجام تلاش‌ها و کارهایی در راستای توسعه و تکمیل یک نشان‌گذاری هماهنگ و ثابت پیشنهاد شده است [BRU88][7] و [YOU89][8] و برخورد مدرنیزه با مسائل به‌منظور هماهنگ نمودن استفاده از ابزارهای CASE انتشار یافته است.

۱۲-۲ عناصر مدل تحلیل

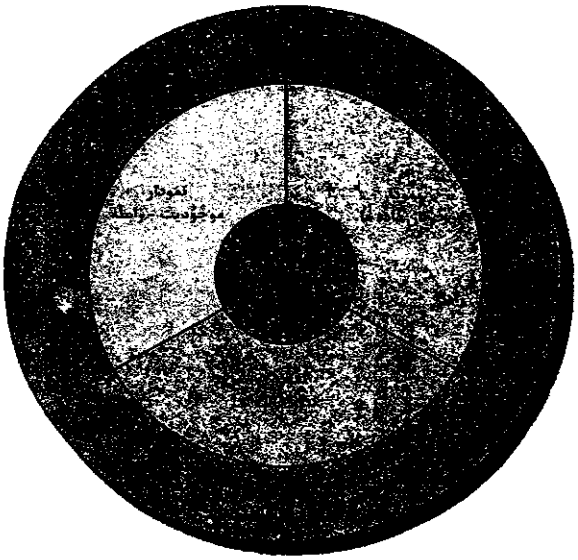
مدل تحلیلی (یعنی مدل تحلیلی ساخت یافته) باید سه هدف اولیهٔ زیر را تأمین کند: (۱) توصیف کردن نیاز مشتری (۲) به‌وجود آوردن مبنایی برای ایجاد طراحی نرم‌افزاری و (۳) تعریف کردن مجموعه‌ای از نیازمندیها که پس از ساخته شدن نرم‌افزار می‌توانند معتبر شوند. مدل تحلیلی که در طول تحلیل ساخت یافته به‌دست می‌آید، برای تأمین این اهداف، شکل نشان داده شده در تصویر ۱۲-۱ را به‌خود گرفته است.

در مرکز این مدل واژه‌نامهٔ داده‌ها[9] ـ یعنی مجموعه‌ای که شامل توضیحات مربوط به تمام اشیاء داده‌ای مصرف شده یا به‌وجود آمده به‌وسیلهٔ نرم‌افزار است ـ قرار گرفته است و سه نمودار متفاوت نیز پیرامون بخش مرکزی این مدل یعنی واژه‌نامهٔ داده‌ها وجود دارد. نمودار رابطه- موجودیت (ERD) روابط

1.structured analysis
2.Douglas Ross
3.DeMarco,T.
4.Page – Jones , M
5.Gane,T. and CC. Sarson
6.Ward ,P.T.and S.J.
7.Bruyn , W. et al.
8.Yourdon ,E.N.
9.data dictionary

موجود میان اشیاء داده‌ای را نشان می‌دهد. نمودار رابطه- موجودیت[1] (ERD) نشان‌گذاری‌ای است که برای اجرای مدل‌سازی داده‌ها مورد استفاده قرار می‌گیرد. صفات مشخصهٔ مربوط به اشیاء داده‌ای را با نمودار می‌توان توصیف نمود.

نمودار جریان داده‌ها (DFD) [2] دو هدف مهم را برآورده می‌سازد: این دو هدف عبارتند از: (۱) فراهم کردن



شکل ۱۲ - ۱ ساختار یک مدل تحلیلی

شاخصی در ارتباط با نحوهٔ تبدیل داده‌ها هنگام حرکت داده‌ها در طول سیستم (۲) نشان دادن کارکردهای اصلی (وکارکردهای فرعی) که تبدیل و تغییر جریان داده‌ها را انجام می‌دهند. نمودار جریان داده‌ها اطلاعات اضافی و بیشتری را فراهم می‌کند که می‌توان در طول تحلیل میدان اطلاعات از آنها استفاده نمود. همچنین این اطلاعات اضافی می‌توانند مبنایی را برای مدل‌سازی یک کارکرد (تابع) به‌وجود آورند. توصیف هر یک از کارکردهای ارائه شده در DFD در مشخصات فرآیند[3]، (PSPEC) گنجانده می‌شود.

نمودارگذار حالت[4](STD) نحوهٔ برخورد سیستم در نتیجهٔ وقوع رویدادهای خارجی را نشان می‌دهد. STD برای نشان دادن کامل نحوهٔ برخورد سیستم، حالت‌های مختلف سیستم و روشی که در آن گذر (تغییر) از حالتی به حالت دیگر پدید می‌آید را نشان می‌دهد. STD به‌عنوان مبنای مدل‌سازی رفتاری

1.entity relation diagram
2.Data Flow Diagram
3.process specification
4.state transition diagram

عمل می‌کند. اطلاعات اضافی در مورد جنبه‌های کنترل نرم‌افزاری را می‌توان در مشخصات کنترل[1] (CSPEC) یافت.

مدل تحلیلی هر یک از نمودارها، مشخصات، توصیفات و واژه نامهٔ موجود در شکل ۱۲-۱ را احاطه کرده است. بررسی جزیی‌تر این عناصر تحلیلی در بخش‌های بعدی ارائه خواهد شد.

۱۲-۳ مدل سازی داده ها

مدل‌سازی داده‌ها پاسخ‌گوی مجموعه‌ای از سؤالات خاص است که در ارتباط با کاربرد پردازش داده‌ها می‌باشند. به‌عنوان مثال: اشیاء داده‌ای اولیه که باید توسط سیستم پردازش شوند، چه چیزهایی هستند؟ ترکیب اشیاء داده‌ای چه چیزی می‌باشد و چه صفات خاصه‌ای را توصیف می‌کنند؟ این اشیاء در کجا مستقر هستند؟ روابط میان اشیاء و فرآیندهای تغییردهنده و پردازش‌کنندهٔ این اشیاء چه چیزهایی می‌باشند؟

پرسش

مدل سازی داده، چه سؤالاتی را پاسخ می دهد؟

برای پاسخ‌گویی به این سؤالات، شیوه های مدل‌سازی داده‌ها از نمودار رابطه- موجودیت (ERD) استفاده می‌کنند. ERD، که به‌صورت مفصل در بخش‌های بعدی این فصل توضیح داده خواهد شد، امکان شناسایی اشیاء داده‌ای و روابط آنها را با استفاده از نشانه‌گذاری گرافیکی برای مهندسین نرم‌افزار فراهم می‌کند. در زمینهٔ تحلیل ساخت یافته، ERD تمام داده‌های ورودی، ذخیره شده، تغییر یافته و به‌وجود آمده در هنگام استفاده را تعریف می‌کند.

نقل قول

قدرت رهیافت ER قدرت تشریح موجودیت ها در جهان واقعی اداری و تجاری و ارتباط میان آنهاست مرتیا مدل

نمودار رابطه- موجودیت فقط بر داده‌ها تأکید دارد و یک "شبکهٔ داده‌ای" را که در سیستم مفروض وجود دارد، نشان می‌دهد. استفاده از ERD برای مواردی که داده‌ها و روابط حاکم بر داده‌ها پیچیده می‌باشند، بسیار مفید و سودمند است. برخلاف نمودار جریان داده‌ها (که در بخش ۱۲-۴ مورد بررسی قرار خواهد گرفت و برای نشان دادن نحوهٔ تبدیل داده‌ها مورد استفاده قرار می‌گیرند) مدل‌سازی داده‌ها، داده‌های موجود را مستقل از روند پردازشی تغییردهندهٔ آنها در نظر می‌گیرد.

۱۲-۳-۱ اشیاء داده ای، صفات خاصه و روابط

یک شیء داده ای بازنمای هر اطلاع ترکیبی است که توسط نرم افزار کامپیوتر پردازش می شود.

مدل داده‌ها از سه بخش مرتبط اطلاعاتی تشکیل شده است. این سه بخش عبارتند از: شئ داده‌ای، ویژگی‌های توصیف‌کنندهٔ شئ داده ای و روابطی که میان اشیاء داده‌ای ارتباط برقرار می‌کنند.

اشیاء داده‌ای، شئ داده‌ای[2] نشانگر هر یک از ترکیبات اطلاعاتی است که باید توسط نرم‌افزار درک شود. منظور از اطلاعات مرکب[3] (ترکیب شده) اطلاعاتی است که دارای ویژگی‌ها و صفات خاصه متفاوت

1.control specification

2.data object

3.composite information

می‌باشند. بنابراین عرض (یک مقدار تنها و بدون ترکیب) نمی‌تواند یک شئ داده‌ای معتبر باشد، اما ابعاد (ترکیب ارتفاع، عرض و طول) را می‌توان به‌عنوان یک شئ تعریف نمود.

یک شئ داده‌ای می‌تواند دارای موجودیت خارجی باشد (به‌عنوان مثال هر چیزی که بتواند اطلاعاتی را به‌وجود آورد و یا از اطلاعات موجود استفاده کند یک شئ داده‌ای محسوب می‌شود). همچنین یک شئ داده‌ای می‌تواند یک شی (به‌عنوان مثال گزارش یا تصویر)، یک واقعه (به‌عنوان مثال زنگ زدن تلفن) یا اتفاق (به‌عنوان مثال اعلان خطر)، یک نقش (به‌عنوان مثال فروشنده)، یک واحد سازمانی (به‌عنوان مثال بخش حسابداری)، یک مکان (به‌عنوان مثال انبار) یا یک ساختار (به‌عنوان مثال فایل) باشد. به‌عنوان مثال یک ماشین یا یک فرد (شکل۱۲-۲) را می‌توان به‌عنوان یک شئ داده‌ای درنظر گرفت، بدین مفهوم که این اشیاء داده‌ای را می‌توان برحسب مجموعه‌ای از صفات خاصه توصیف نمود. در توصیف شئ داده‌ای باید شئ داده‌ای و تمام ویژگی‌های آن را با یکدیگر ادغام نمود.

شکل ۱۲ - ۲ اشیاء داده ای ، صفات خاصه , رابطه ها

اشیاء داده‌ای (که به صورت پررنگ نشان داده شده‌اند) در ارتباط با یکدیگر هستند. به‌عنوان مثال یک فرد می‌تواند مالک یک خودرو باشد که در این مثال رابطهٔ مالکیت می‌تواند "رابطهٔ" خاصی میان فرد و خودرو برقرار سازد. روابط موجود میان اشیاء داده‌ای همیشه در بافت مسائلی که مورد تجزیه و تحلیل قرار می‌گیرند، تعریف می‌شوند.

یک شئ داده‌ای فقط داده‌ها را احاطه می‌کند و هیچ ارجاعی، در شئ داده‌ای در ارتباط با عملیاتی که بر روی داده‌ها انجام می‌شود، وجود ندارد.[1] بنابراین می‌توان شئ داده‌ای را به‌صورت جدولی که در شکل ۱۲-۳ آمده است، نشان داد. سر تیترهای موجود در این جدول صفات خاصه شئ مورد نظر را نشان می‌دهند. در این مثال، ماشین برحسب کارخانهٔ سازنده، مدل، شماره شناسایی یا پلاک (ID#) نوع بدنه،

۱. این تفکیک، اشیاء داده ای را از کلاس یا شئ تعریف شده در پارادایم شئ گرا (که در بخش چهارم این کتاب توضیح داده شده است) جدا می سازد.

رنگ و مالک آن تعریف می‌شود. بخش اصلی این جدول مثال‌های خاصی از شئ داده‌ای مورد نظر را ارائه می‌کند، به‌عنوان مثال هوندا آکورد یک نمونه از شئ داده‌ای خودرو می‌باشد.

صفات خاصه. صفات خاصه، مشخصات یک مصداق شئ داده‌ای را تعریف می‌کنند و می‌توانند یکی از سه مشخصهٔ متفاوت را دارا باشند. صفات خاصه را می‌توان برای (۱) نام‌گذاری نمونه‌ای از اشیاء داده‌ای (۲) توصیف نمونهٔ مفروض و یا (۳) ارجاع دادن به نمونه‌ای دیگر در جدول دیگری استفاده نمود. به‌علاوه یک یا چند صفت از این صفات خاصه باید به‌عنوان شناسه[1] تعریف شوند ـ یعنی شناسه صفت خاصه هنگام جستجو برای یافتن نمونه‌ای از یک شئ داده‌ای یک ویژگی کلیدی محسوب می‌شوند. در بعضی موارد مقادیر و ارزش‌های شناسه منحصربه‌فرد می‌باشند، با این وجود این مسأله به‌عنوان یک الزام نمی‌باشد. به‌عنوان مثال در نمونه شئ داده‌ای ماشین، شناسهٔ قابل قبول و منطقی برای خودرو ID# (شماره پلاک)می‌باشد.

اطلاعات مفید در خصوص مدل سازی داده ها را در آدرس زیر بیابید :
www.datamodel.org

مجموعهٔ صفات خاصه متناسب با شئ داده‌ای مفروض را می‌توان از طریق درک نمودن بافت مسأله تعیین کرد. صفات خاصه توضیح داده شده در قسمت بالا برای خودرو می‌توانند برای استفادهٔ وزارتخانه از خودروهای موتوری سودمند و مفید واقع شوند؛ اما شاید همین ویژگی‌ها برای یک شرکت خودروسازی که به نرم‌افزار کنترل ساخت نیاز دارد، مفید و سودمند نباشند. در شرکت خودروسازی صفات خاصه ذکر شده برای خودرو باید شامل شناسه (ID#)، نوع بدنه و رنگ باشد و علاوه بر این ویژگی‌های اضافی (هم‌چون کد داخلی، نوع انتقال قدرت، نوع محرک و ...) نیز باید به صفات خاصه موجود اضافه شود تا ماشین در بافت کنترل ساخت، به یک موجودیت (شی) با مفهوم و معنی‌دار تبدیل شود.

صفات خاصه، یک شئ داده را نام گذاری می کند، توصیف می کند و در برخی موارد ارتباطی با دیگر اشیاء برقرار می سازند.

1.identifier

ارتباط دهنده یک شیء داده ای به دیگری
(در اینجا مالک)

صفات مشخصه مرتبط با نام

صفات خاصه ارجاعی

اسم شناسه

صفات خاصه توصیفی

مصداق

سازنده	مدل	شماره پلاک	نوع بدنه	رنگ	مالک
Lexus	LS400	AB123...	Sedan	White	RSP
Chevy	Corvette	X456...	Sports	Red	CCD
BMW	750iL	XZ765	Coupe	White	LJL
Ford	Taurus	Q12A45	Sedan	Blue	BLF

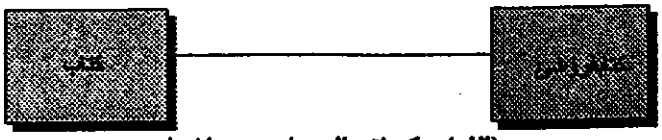
شکل ۱۲ - ۳ بازنمایی اشیاء داده ای در قالب جدول

رابطه ها مشخص می کنند که چگونه اشیاء داده ای به یکدیگر "متصل" می شوند.

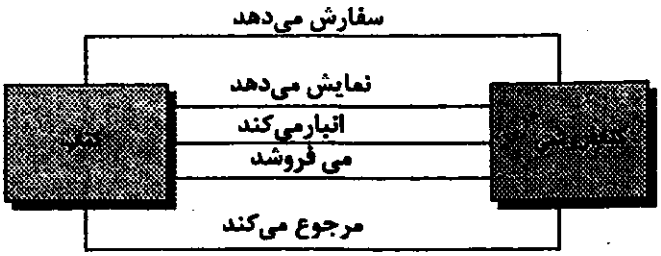
روابط. اشیاء داده‌ای به اشکال و شیوه‌های مختلف به یکدیگر مرتبط شده‌اند. به‌عنوان مثال اشیاء داده‌ای کتاب و کتاب فروشی را در نظر بگیرید. می‌توان با استفاده از نشان‌گذاری ساده‌ای که در شکل ۱۲-۴ الف نشان داده شده است. این اشیاء ها (یعنی کتاب و کتاب فروشی) را نمایش داد. میان کتاب و کتاب فروشی یک خط متصل‌کننده رسم شده است به‌دلیل آن‌که میان این دو شئ ارتباط وجود دارد. اما رابطهٔ میان این دو چه چیزی می‌باشد؟ برای پاسخ‌گویی به این سؤال باید نقش کتاب‌ها و کتاب‌فروشی‌ها را در بافت نرم‌افزاری که قرار است ساخته شود، در نظر بگیریم. می‌توانیم مجموعه‌ای از جفت‌های شئ - رابطه که روابط مربوطه را تعریف و مشخص می‌کند، تعریف کنیم. به‌عنوان مثال:

- یک کتاب فروش سفارش خرید کتاب‌هایی را صادر می‌کند.
- یک کتاب فروشی کتاب‌های خود را در معرض نمایش می‌گذارد.
- یک کتاب فروشی کتاب‌های خود را انبار می‌کند.
- یک کتاب فروشی به فروش کتاب‌ها مشغول می‌شود.
- یک کتاب فروشی کتاب‌ها را پس می‌دهد.

روابط سفارش دادن، در معرض نمایش گذاشتن، انبار کردن، فروختن و پس دادن روابط مربوطه میان کتاب و کتاب فروشی را مشخص می‌کنند. (شکل ۱۲-۴ ب) این جفت‌های شئ - رابطه‌ای را به‌صورت نمودار گرافیکی نشان می‌دهد.



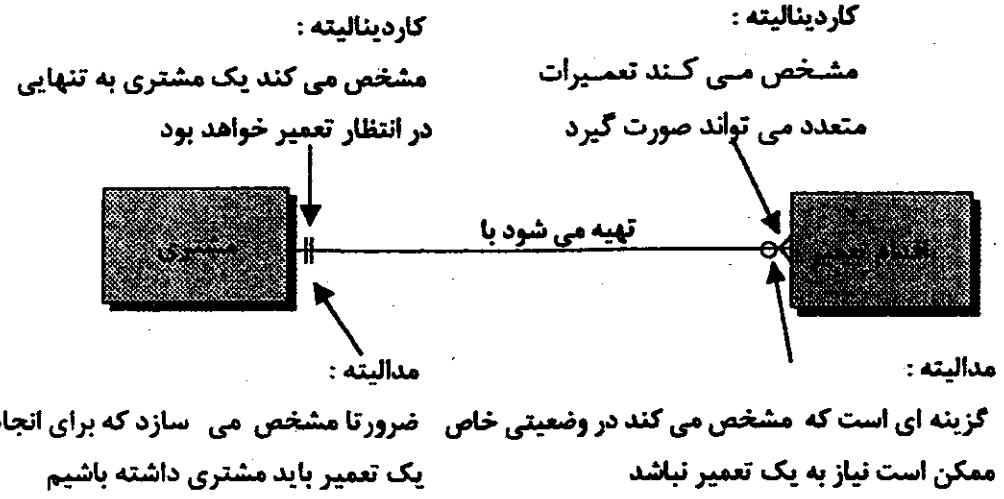
(الف) یک اتصال ساده بین اشیاء



(ب) رابطه هایی بین اشیاء

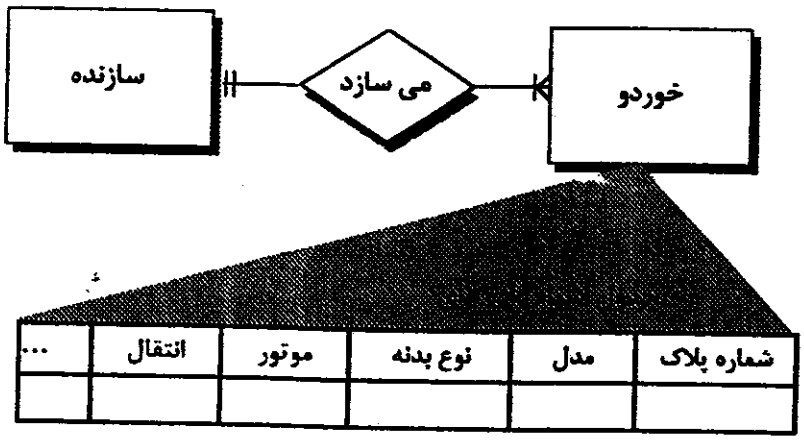
شکل ۱۲ـ۴ رابطه ها

همچنین توجه به این نکته نیز حائز اهمیت است که جفت‌های شئ ـ رابطه‌ای دو سویه می‌باشند. یعنی می‌توان این جفت‌ها را در هر دو جهت خواند. به‌عنوان مثال یک کتاب فروشی سفارش خرید کتاب را می‌دهد و کتاب‌ها توسط کتاب فروشی سفارش داده می‌شوند.[1]



شکل ۱۲ـ۵ کاردینالیته و مدالیته

۱ برای اجتناب از ابهام، نحوه برچسب خوردن رابطه ها باید مورد توجه قرار گیرد. برای مثال، اگر متنی برای یک رابطه دوسویه (دوجهته) در نظر گرفته نشده است. شکل ۱۲-۴ ب ممکن است به اشتباه تفسیر شود : که کتابها، کتابفروش را سفارش می دهند! در چنین مواردی کلمات را باید عوض نمود



شکل ۱۲ـ۶ یک ERD (نمودار رابطه/موجودیت) ساده و جدول اشیاء داده ای
(توجه شود که در این ERD رابطه "*می سازد*" با یک لوزی مشخص شده)

۱۲-۳-۲ کاردینالیته و مدالیته

عناصر مدل سازی داده‌ها ـ اشیاء داده‌ای، صفات خاصه و روابط ـ مبنا و پایه و اساسی را برای درک ـ میدان اطلاعاتی مربوط به یک مسأله فراهم می‌کنند. با این وجود اطلاعات اضافی مربوط به این عناصر پایه‌ای نیز باید به‌صورت کامل درک شود.

ما مجموعه‌ای از اشیاء را تعریف کرده‌ایم و جفت‌های شیئ / رابطه که میان اشیاء مختلف ارتباط ایجاد می‌کنند را نشان داده‌ایم. اما یک جفت ساده که نشان می‌دهد شیئ **X** با شیئ **Y** در ارتباط است، نمی‌تواند اطلاعات کافی برای اهداف مهندسی نرم‌افزار ارائه دهد. ما باید به این مسأله پی ببریم که چه تعداد از شیئ **X** ما چه تعداد از شیئ **Y** در ارتباط می‌باشد. و این امر به طرح نمونه‌سازی داده‌ها که تعداد مدخل‌ها در یک رابطه نامیده می‌شود، منتهی می‌گردد.

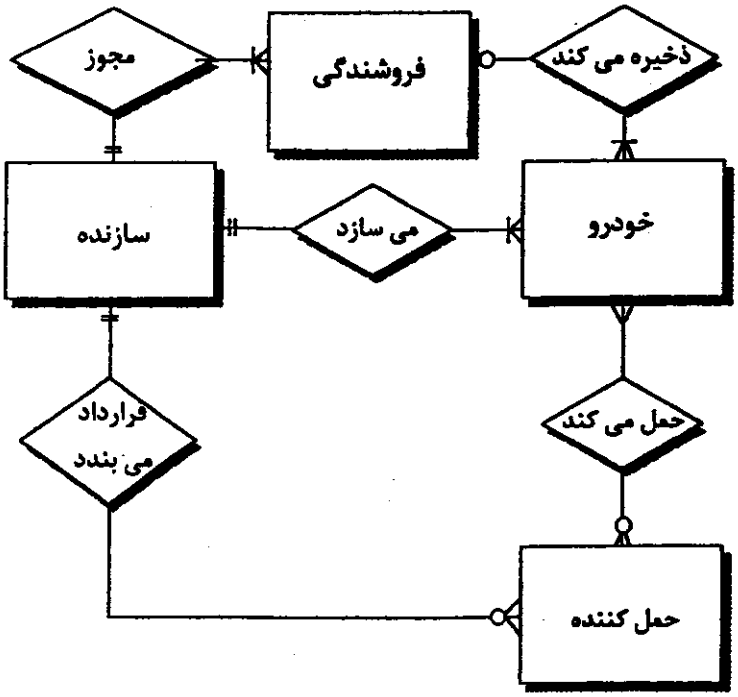
تعداد مدخل‌ها در یک رابطه (کاردینالیته)[1]: مدل داده‌ها باید بتواند تعداد وقوع اشیاء موجود در یک رابطهٔ مفروض را نشان دهد. تیلمن[TIL93][2] تعداد مدخل‌های در یک جفت رابطه ـ شیئ را به‌صورت زیر تعریف می‌کند:

تعداد مدخل‌ها مشخص‌کنندهٔ تعداد وقوع یک شیئ است که می‌تواند با تعداد وقوع شیئ دیگر در ارتباط باشد. تعداد مدخل‌ها معمولاً با "یک" یا "بسیار" نشان داده می‌شود. به‌عنوان مثال هر شوهری فقط می‌تواند یک زن داشته باشد (البته در بسیاری از فرهنگ‌ها) و این در حالی است که یک زن و شوهر

1.cardinality
2.Tillmann ,G.

می‌توانند فرزندان بسیاری داشته باشند. با در نظر گرفتن تمام ترکیبات احتمالی "یک" و "بسیار" دو شئ را می‌توان به‌صورت زیر به یکدیگر مرتبط نمود:

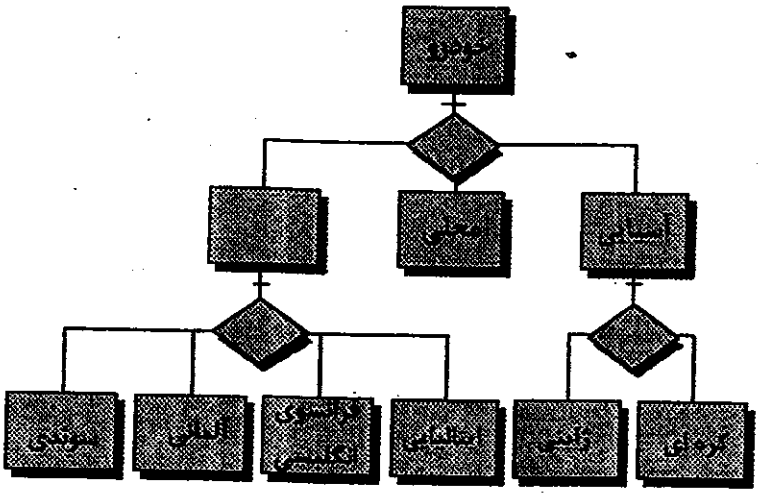
- **ارتباط یک به یک (1:1)** - تعداد وقوع شئ "A" را در این حالت فقط می‌توان به یکی از موارد شئ و "B" را فقط می‌توان به یکی از موارد شئ "A" مرتبط نمود.
- **ارتباط یک به بی‌شمار (1:N)** - یک مورد از شئ "A" را می‌توان به تعداد زیادی از موارد شئ "B" مرتبط نمود، اما یک مورد از شئ B را فقط می‌توان به یک مورد از A مرتبط نمود. به‌عنوان مثال یک مادر می‌تواند فرزندان بی‌شماری داشته باشد، اما یک کودک فقط می‌تواند یک مادر داشته باشد.
- **ارتباط بی‌شمار به بی‌شمار (M:N)** - یک مورد از "A" را می‌توان به یک مورد از B و یا تعداد بی‌شمار از موارد B و یک مورد از B را می‌توان به یک یا تعداد بی‌شماری از موارد A مرتبط نمود. به‌عنوان مثال یک نفر دایی می‌تواند تعدادی بی‌شماری خواهرزاده داشته باشد و یک خواهرزاده نیز می‌تواند تعداد بی‌شماری دایی داشته باشد.



شکل ۱۲ـ۷ یک ERD (نمودار رابطه / موجودیت) توسعه یافته

تعداد مدخل‌ها (کاردینالیته) می‌تواند "حداکثر تعداد روابط اشیاء شرکت‌کننده در یک رابطه" را مشخص و تعیین می‌کند. با این وجود تعداد مدخل‌ها نمی‌تواند مشخص‌کنندهٔ این مسأله باشد که آیا یک شئ داده‌ای خاص می‌تواند در یک رابطه شرکت کند و یا خیر. برای مشخص کردن این اطلاعات، مدل داده‌ای، مدالیته را به جفت رابطه- شئ اضافه می‌کند.

مدالیته[1]: اگر وجود رابطه میان دو شئ مختلف به‌صورت انتخابی باشد و یا نیازی برای برقراری ارتباط وجود نداشته باشد، مدالیته رابطه صفر خواهد بود. برای روشن شدن این مطلب نرم‌افزاری را در نظر بگیرید که یک شرکت محلی تعمیر تلفن برای مرتب کردن تقاضاهای موجود برای تعمیر تلفن در محل از آن استفاده می‌کند. یکی از مشتریان اظهار می‌کند که مشکلی وجود دارد. اگر این مشکل به راحتی تشخیص داده شود، کارهای ساده‌ای نیز برای تعمیر تلفن متعلق به آن مشتری صورت می‌گیرد. اما اگر مشکل به‌وجود آمده در سیستم تلفن این مشتری پیچیده باشد باید کارهای مختلف و چندگانه‌ای برای تعمیر آن انجام شود. شکل ۱۲-۵ رابطه، تعداد مدخل‌ها (روابط) و مدالیتهٔ موجود میان اشیاء داده‌ای یعنی مشتری و کارهای تعمیراتی را نشان می‌دهد.



شکل ۱۲ـ۸ سلسله مراتب نوع اشیاء داده‌ای

با توجه به این شکل مشخص می‌شود که رابطهٔ تعداد مدخل‌ها در این مثال ۱ به بی‌شمار (1:N) می‌باشد. یعنی یک مشتری می‌تواند کارهای تعمیراتی صفر و یا بی‌شماری را تقاضا کند. نمادهای موجود در زمینه اتصال رابطه‌ای نزدیک به مستطیل‌های شئ داده‌ای نشانگر تعداد مدخل‌های رابطه‌ای می‌باشد. ستون عمودی وجود یک رابطه و انشعاب ۳ اتصالی نیز روابط بی‌شماری را نشان می‌دهد. مدالیته نیز با

1.modality

علایمی که دورتر از مستطیل‌های شئ داده‌ای قرار دارند، مشخص می‌گردد. دومین ستون عمودی در سمت چپ نشانگر این مطلب است که برای به وقع پیوستن کارهای تعمیری وجود یک مشتری ضروری و الزامی می‌باشد. دایرهٔ سمت راست نشانگر این مسأله است که شاید هیچ‌گونه کار تعمیری برای نوع مشکل گزارش شده از سوی مشتری مورد نیاز نباشد.

۱۲-۳-۳ نمودارهای موجودیت / رابطه

مقصود اصلی نمودار رابطه/موجودیت (ERD) بازنمایی موجودیت ها (اشیاء داده ای) و رابطه میان هریک از آنها می باشند.

جفت رابطه ـ شئ (که در قسمت ۱۲-۳-۱ مورد بحث و بررسی قرار گرفت) پایه و اساس مدل داده‌ای می‌باشد. این جفت‌ها را می‌توان با استفاده از نمودار رابطه- موجودیت به‌صورت گرافیکی نشان داد. نمودار رابطه- موجودیت [1] (ERD) در ابتدا توسط پیترچن* برای طراحی سیستم‌های پایگاه داده رابطه‌ای پیشنهاد شد و تکمیل و توسعهٔ این نمودار توسط سایر افراد انجام شده است. مجموعه‌ای از عناصر اولیه برای ERD مشخص شده است. این عناصر عبارتند از:اشیاء داده‌ای، صفات خاصه، روابط و انواع مختلف شاخص‌ها. هدف اصلی ERD نشان دادن اشیاء داده‌ای و روابط آنها می‌باشد.

طرح اولیهٔ ERD در بخش ۱۲-۳ ارائه شده است. اشیاء داده‌ای به‌وسیلهٔ مستطیل‌های برچسب‌دار نشان داده می‌شود. در بعضی از تغییرات ERD، خط متصل‌کنندهٔ اشیاء به یکدیگر دارای یک لوزی است که نوع رابطه در داخل آن ذکر شده است. اتصالات موجود میان اشیاء داده‌ای و روابط، با استفاده از نمادهای خاص متنوع که نشانگر تعداد مدخل‌های رابطه‌ای و مدالیته هستند (به بخش ۱۲-۳-۲ مراجعه شود)، به‌وجود می‌آیند.

رابطهٔ موجود میان اشیاء داده‌ای یعنی ماشین و کارخانهٔ سازنده (در این مثال) را می‌توان به‌صورت نمودار ۱۲-۶ نمایش داد. در این مثال یک کارخانهٔ اتومبیل‌سازی یک یا تعداد بسیاری ماشین را می‌سازد.

با توجه به بافت پیشنهاد شده در ERD، مشخصات شئ داده‌ای خودرو (جدول شئ داده‌ای در نمودار ۱۲-۶) اساساً متفاوت از مشخصات اولیه خواهد بود (نمودار ۱۲-۳). با بررسی کردن نمادها در قسمت پایانی خط اتصال موجود در میان اشیاء، مشخص می‌شود که مدالیتهٔ هر دو وقوع الزامی است (خطوط عمودی).

با تکمیل کردن و گسترش دادن این مدل یک ERD بسیار ساده (شکل ۱۲-۷) از عناصر توزیعی کار اتومبیل‌سازی ارائه خواهیم داد. اشیاء جدید داده‌ای یعنی حمل‌کننده و واسطه حمل و نقل نیز مطرح

1.entity/relationship diagram

* Peter chen

می‌شوند. به‌علاوه روابط جدید ـ یعنی حمل و نقل[1]، قراردادها[2]، جواز[3] و انبار[4] مربوط به هر یک از اشیاء داده‌ای موجود در ERD باید مطابق با قوانین معرفی شده در ابتدای این فصل تکمیل شود.

علاوه بر طرح اولیه ERD که در نمودار ۱۲-۶ و ۱۲-۷ معرفی شد، فرد تحلیل‌گر می‌تواند سلسله مراتب نوع شئ داده‌ای[5] را نیز نشان دهد. در بسیاری از موارد، شئ داده‌ای در واقع می‌تواند نشانگر یک کلاس یا طبقهٔ اطلاعات باشد. به‌عنوان مثال، شئ داده‌ای خودرو را می‌توان به ماشین‌های داخلی، اروپایی یا آسیایی تقسیم‌بندی نمود. طرح ERD نشان داده شده در نمودار ۱۲-۸ این طبقه‌بندی را به شکل سلسله مراتبی نشان می‌دهد.

همچنین طرح ERD مکانیسمی را فراهم می‌کند که نشانگر روابط موجود میان اشیاء می‌باشد. همان‌طور که در نمودار ۱۲-۹ نشان داده شده است در طرح ERD شئ داده‌ای انجمنی[6] نیز ارائه می‌گردد. در این نمودار هر یک از اشیاء داده‌ای مدل‌سازی‌کنندهٔ هر یک از سیستم‌های فرعی با شئ داده‌ای خودرو در ارتباط هستند.[ROS85] [7]

مدل‌سازی داده‌ای و نمودار موجودیت - رابطه، تحلیلی را همراه با طرح دقیق برای مورد بررسی قرار دادن داده‌ها در بافت کاربرد نرم‌افزاری فراهم می‌کنند. در بسیاری از موارد روش مدل‌سازی داده‌ای برای ایجاد بخشی از مدل تحلیلی مورد استفاده قرار می‌گیرد، اما می‌توان از این روش برای طراحی پایگاه داده‌ای‌ها و حمایت کردن از سایر نیازمندیها شیوه‌های تحلیلی استفاده نمود.

۱۲-۴ مدل سازی کارکردی و جریان اطلاعات

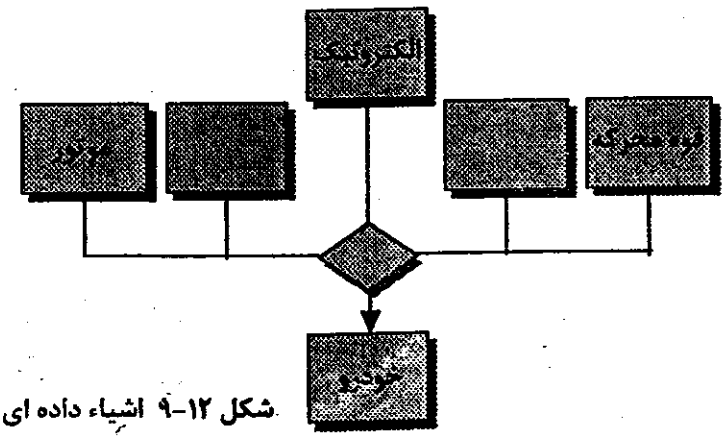
اطلاعات هنگام جریان یافتن در یک سیستم کامپیوتری انتقال داده می‌شوند. این سیستم ورودی‌ها را به شکلهای مختلف می‌پذیرد و قطعات سخت‌افزاری، نرم‌افزاری و عناصر انسانی را برای انتقال داده‌های ورودی به‌کار می‌برد و خروجی را به اشکال مختلف به‌وجود می‌آورد. ورودی ممکن است به‌صورت یکی از موارد زیر باشد:

سیگنال‌های کنترل مخابره شده توسط مبدل، مجموعه‌ای از اعداد تایپ شده توسط اپراتور انسانی، بسته‌ای از اطلاعات انتقال یافته به خط ارتباطی شبکه‌ای و یا یک فایل داده‌ای حجیم که از حافظهٔ ثانویه بازیابی شده است. تبدیلات ممکن است فقط از یک مقایسهٔ منطقی ساده، مجموعه‌ای از الگوریتم‌های

1.transport
2.contracts
3.licenses
4.stocks
5.data object type hierarchies
6.associative data object
7.Ross, D.

عددی پیچیده یا رهیافت استنتاج قواعد سیستم مبتنی بر قاعده تشکیل شده باشد. خروجی ممکن است فقط یک LED را روشن کند و یا یک گزارش ۲۰۰ صفحه‌ای را به‌وجود آورد. در عمل ما می‌توانیم صرف‌نظر از اندازه و پیچیدگی سیستم کامپیوتری مدل جریان[1] را برای هر یک از سیستم‌های کامپیوتری به‌وجود آوریم.

تحلیل ساخت یافته به‌عنوان تکنیک مدل‌سازی جریان اطلاعاتی آغاز شد. همان‌طور که در نمودار ۱۲-۱۰ نشان داده شده است، می‌توان یک سیستم کامپیوتری را به‌عنوان تبدیل اطلاعات نمایش داد. در این نمودار از یک مستطیل برای نشان دادن موجودیت (هستی) خارجی[2]، یعنی یک عنصر سیستمی (همانند سخت‌افزار، یک فرد، یا یک برنامهٔ دیگر) یا سیستم دیگر که اطلاعاتی را برای تبدیل شدن توسط نرم‌افزار به‌وجود می‌آورد یا اطلاعات به‌وجود آمده توسط نرم‌افزار را دریافت می‌کند، استفاده شده است. یک دایره (که در بعضی مواقع حباب[3] نامیده می‌شود) در این نمودار وجود دارد که نشانگر پردازش[4] یا تبدیل داده‌ها[5] و تغییراتی است که بر روی داده‌ها انجام می‌گیرد. در این نمودار از یک پیکان برای نشان دادن یک یا چند قلم داده‌ای[6] (شئ داده‌ای) استفاده شده است. تمام پیکان‌های موجود در نمودار جریان داده‌ها باید نام‌گذاری شده باشند. خط دو سویه نیز ذخیرهٔ اطلاعاتی ـ یعنی اطلاعات ذخیره شده‌ای که توسط نرم‌افزار مورد استفاده قرار می‌گیرند ـ را نشان می‌دهد. سادگی طرح DFD یکی از دلایل استفاده گسترده از فنون تحلیل ساخت یافته می‌باشد.



شکل ۱۲-۹ اشیاء داده ای شرکت پذیر

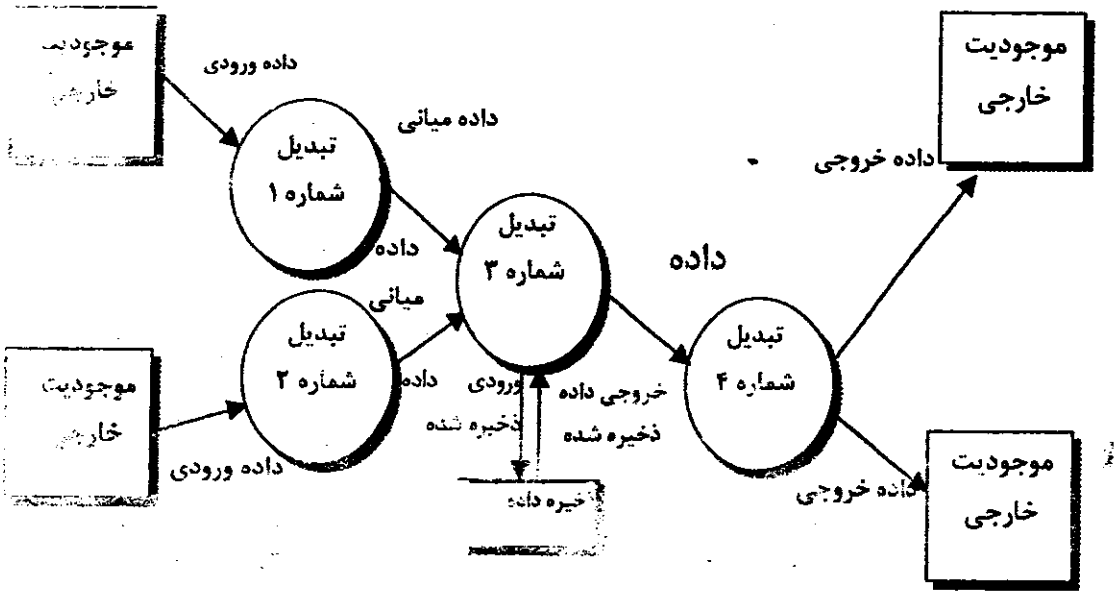
توجه به این نکته نیز حائز اهمیت است که هیچ‌گونه علامت مشخصی در ارتباط با توالی پردازشی یا منطق شرطی در این نمودار ارائه نمی‌شود. ممکن است روش یا توالی پردازشی به‌صورت ضمنی در این

1 flow model
2 [illegible]
3 [illegible]

نمودار وجود داشته باشد. اما ارائهٔ جزییات منطقی آشکار در ارتباط با توالی پردازشی تا هنگام طراحی نرم‌افزاری به تعویق می‌افتد. اشتباه نگرفتن DFD با نمودار جریان نیز از جملهٔ مسایل مهم می‌باشد.

۱۲-۴-۱ نمودارهای جریان داده ها

با حرکت کردن اطلاعات در نرم‌افزار، این اطلاعات توسط مجموعه‌ای از تبدیلات تغییر داده می‌شوند. نمودار جریان داده‌ها[1] (DFD) نمایش گرافیکی است که جریان اطلاعات و تبدیلاتی که به هنگام حرکت داده‌ها از ورودی به خروجی در مورد این اطلاعات اعمال شده است را نشان می‌دهد. شکل اصلی نمودار جریان داده‌ها که به‌عنوان گراف جریان داده‌ها[2] یا نمودار حبابی[3] نیز شناخته می‌شود در نمودار ۱۲-۱۰ نمایش داده شده است.



شکل ۱۲-۱۰ مدل جریان اطلاعات

جزییات کاربردی هستند، تقسیم نمود. بنابراین DFD، مکانیسمی را برای مدل‌سازی کاربردی و مدل‌سازی جریان اطلاعات ارائه می‌دهد. و DFD با انجام این نوع مدل‌سازی‌ها دومین اصل تحلیل عملیاتی را (یعنی ایجاد یک مدل کارکردی) که در فصل ۱۱ مورد بررسی قرار گرفت، تأمین می‌کند.

سطح صفر نمودار گردش داده‌ها، که مدل اصلی سیستم[1] یا مدل بافتی[2] (زمینه‌ای) نیز نامیده می‌شود، کل عناصر نرم‌افزار را به شکل یک حباب نشان می‌دهد که داده‌های ورودی و خروجی در این حباب با پیکان‌هایی که به‌ترتیب جهت آنها به سمت داخل و به سمت خارج است، مشخص می‌گردد. پردازش‌های اضافی و مسیرهای اضافی جریان اطلاعات هنگام تقسیم‌بندی سطح 0 نمودار گردش داده‌ها برای نشان دادن جزییات کامل امور نمایان می‌گردند. به‌عنوان مثال سطح ۱ نمودار جریان داده‌ها ممکن است دارای ۵ یا ۶ حباب باشد که با پیکان‌هایی به یکدیگر متصل شده‌اند. انجام هر یک از فرآیندهای پردازشی موجود در سطح ۱، وظیفهٔ فرعی کل سیستم ذکر شده در مدل بافتی می‌باشند.

همان‌طور که در قسمت‌های قبلی نیز ذکر شد، هر یک از لایه‌ها ممکن است برای نشان دادن جزییات بیشتر و کامل‌تر، پالایش و یا لایه‌بندی شوند. در شکل (نمودار) ۱۲-۱۱ این طرح نشان داده شده است. مدل اصلی سیستم F نشان می‌دهد که ورودی اولیه A است و خروجی نهایی B می‌باشد. در این‌جا مدل F را به توابع f_1 تا f_7 تبدیل نموده‌ایم. توجه داشته باشید که پیوستگی جریان اطلاعات[3] باید حفظ شود یعنی با ایجاد هرگونه پالایش ورودی و خروجی، باید این ورودی و خروجی به شکل سابق خود باقی بمانند. در پاره‌ای از موارد این طرح که موازنه نامیده می‌شود برای تکمیل مدل‌های سازگار ضروری هستند. تغییرات بیشتر در F_4 جزییات بیشتری را به شکل توابع f_{41} تا f_{45} نشان خواهد داد. در این حالت و با وجود ایجاد این تغییرات ورودی (X,Y) و خروجی (Z) بدون تغییر باقی خواهند ماند.

طرح اصلی استفاده شده برای تکمیل یک DFD به تنهایی برای توصیف نیازمندیهای نرم‌افزار کافی نمی‌باشد. به‌عنوان مثال پیکان استفاده شده در یک DFD نشان‌گر شئ داده‌ای است که وارد یک فرآیند پردازشی می‌شود و یا از آن خارج می‌گردد. ذخیرهٔ داده‌ای نیز نشانگر جمع‌آوری سازمان یافتهٔ داده‌ها می‌باشد: اما محتویات داده‌ای که توسط پیکان یا با ذخیره کردن داده‌ها مشخص می‌گردد، شامل چه چیزهایی می‌باشد؟ اگر پیکان مجموعه‌ای از اشیاء را نشان دهد، آنها چه چیزهایی هستند؟ با به‌کار بردن یکی دیگر از عناصر طرح اصلی تحلیل ساخت یافته ـ یعنی واژه‌نامهٔ داده‌ها[4] ـ می‌توان به این سؤالات پاسخ داد. استفاده از واژه‌نامهٔ داده‌ها در قسمت‌های بعدی همین فصل مورد بررسی قرار خواهد گرفت.

1.fundamental system model

2.context model

3.information flow continuity

4.data dictionary

طرح گرافیکی DFD باید با متن توصیفی تکمیل شود. مشخصات پردازشی[1] را می‌توان با مشخص کردن جزییات پردازشی ذکر شده در بخش حبابی DFD مورد استفاده قرار داد. مشخصات پردازشی، ورودی یک تابع، الگوریتم به‌کار برده شده برای تبدیل ورودی و خروجی به‌وجود آمده را تعیین و مشخص می‌کنند. علاوه بر آن مشخصات پردازشی (PSPEC) قیود و محدودیت‌ها و موانع موجود در کار پردازش، ویژگی‌های عملکردی مربوط به پردازش و محدودیت‌های طراحی مؤثر بر روش اجرای پردازش را نشان می‌دهد.

۱۲-۴-۲ توسعه سیستمهای مبتنی بر زمان واقعی (Real - time)

بسیاری از کاربردهای نرم‌افزاری وابسته به زمان و فرآیند پردازشی هستند که همانند داده‌ها کنترل اطلاعاتی بر روی آنها انجام می‌شود. سیستم بلادرنگ باید در تعامل با جهان واقعی و در تعامل با چارچوب زمانی دیکته شده توسط جهان واقعی قرار بگیرد. آویونیک[2] هواپیما، کنترل فرآیند ساخت، محصولات مصرف‌کننده و وسایل صنعتی فقط نمونه‌های معدودی از موارد کاربردی نرم‌افزار بلادرنگ هستند.

برای تطابق دادن تحلیل نرم‌افزار بلادرنگ تعدادی از کارهای تکمیلی در زمینهٔ طرح اولیهٔ تحلیل ساخت یافته تعریف شده است.[WAR85][3] [HAT87][4]

۱۲-۴-۳ فعالیتهای تکمیلی "وارد" و "ملور"

وارد و ملور[5] علائم اولیهٔ تحلیل ساخت یافته را تکمیل نموده‌اند تا نیازهای تحمیل شده توسط سیستم بلادرنگ را به‌شرح زیر تطابق دهند و هماهنگ نمایند:

- گردش اطلاعات جمع‌آوری شده یا به‌وجود آمده بر مبنای استمرار زمانی
- اطلاعات کنترل از کل سیستم و فرآیند کنترل مربوطه عبور می‌کنند.
- در بعضی مواقع نمونه‌های چندگانه‌ای (مصادیق چندگانه) از یک تبدیل مشترک در موقعیت‌های چندوظیفه‌ای مشاهده می‌شوند.
- حالت‌های سیستم و مکانیسمی که موجب به‌وجود آمدن تبدیل و تغییر حالت‌ها می‌شود.

در درصد قابل ملاحظه‌ای از موارد کاربردی سیستم بلادرنگ، این سیستم باید اطلاعات مستمر زمانی به‌وجود آمده به‌وسیله بعضی از فرآیندهای جهان واقعی را کنترل کند. به‌عنوان مثال آزمون بلادرنگ کنترل‌کنندهٔ سیستم موتورهای گازی توربین برای کنترل سرعت توربین، دمای مشعل و انواع مختلف

1.process specification

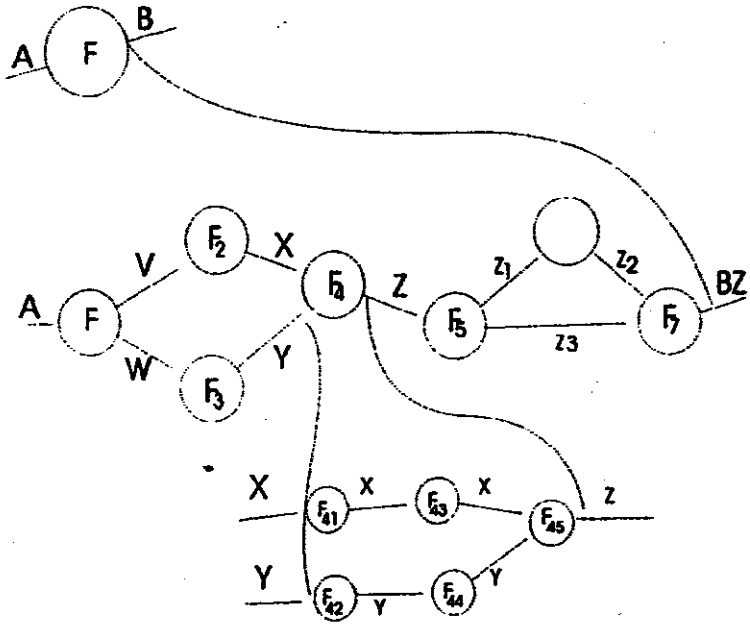
2. arionics

3.Ward ,P.T.and S.J.

4.Hatley , D.J. and I.A.

5.Ward ,P.T.and S.J.

کنترل فشار بر مبنای مستمر مورد نیاز خواهد بود. یکی از کارهای تکمیلی مربوط به طرح اولیهٔ تحلیل ساخت یافته که در شکل (نمودار) ۱۲-۱۲ نشان داده شده است. مکانیسمی را برای نشان دادن جریان مستمر داده‌ها[1] فراهم کرده است. پیکان دو سر برای نشان دادن گردش مستمر و پیکان یک سر برای نشان دادن گردش منقطع داده‌ها مورد استفاده قرار می‌گیرد. در این شکل دمای کنترل شده به‌صورت مستمر اندازه‌گیری می‌شود و این در حالی است که برای نقطهٔ تنظیم دما فقط یک مقدار واحد ارائه شده است. فرآیند نشان داده شده در این شکل مقدار اصلاح شدهٔ خروجی مستمر را به‌وجود می‌آورد.



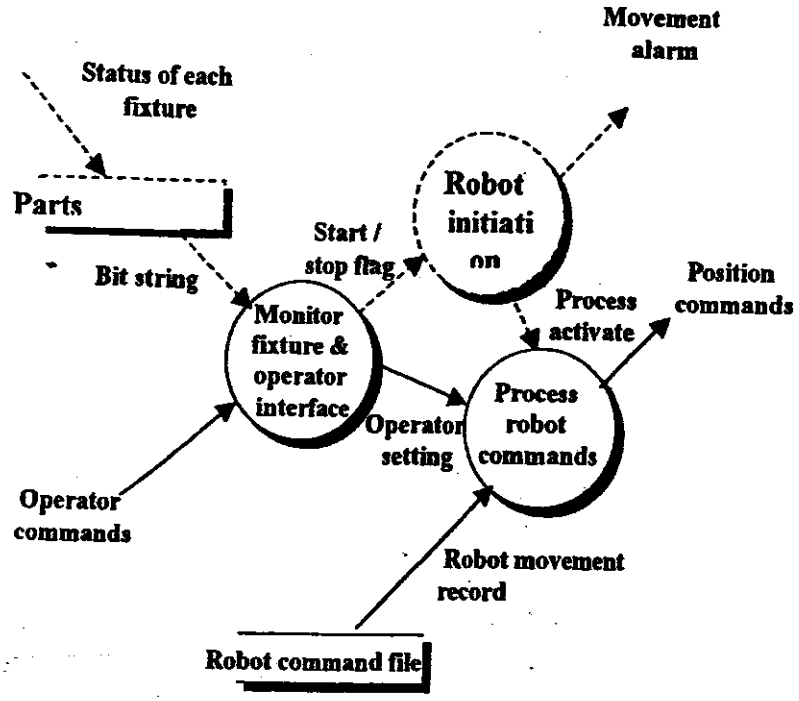
شکل ۱۲ - ۱۱ پالایش جریان اطلاعات

ایجاد تمایز میان گردش داده‌های مستمر و منقطع برای مهندسین سیستم و طراحان نرم‌افزاری حائز اهمیت است و از اهمیت کاربردی بالایی برخوردار می‌باشد. در طول ایجاد مدل سیستم، مهندس سیستم بهتر می‌تواند فرآیندهایی را که دارای حساسیت عملکردی هستند، متمایز نماید (اغلب این احتمال وجود دارد که داده‌های مستمر ورودی و خروجی نسبت به عملکرد موجود حساسیت داشته باشند)، بعد از به‌وجود آمدن مدل واقعی (فیزیکی) و اجرایی، طراح باید مکانیسمی را برای جمع‌آوری داده‌های مستمر زمانی ایجاد کند. کاملاً واضح و مشخص است که سیستم دیجیتالی با استفاده از فنونی هم‌چون نمونه‌گیری سریع، داده‌ها را به شکل شبه مستمر جمع‌آوری می‌کند. این طرح نشان می‌دهد که در چه قسمت‌هایی

1.time-continuous data flow

سخت‌افزار آنالوگ یا دیجیتال مورد نیاز خواهد بود و کدام یک از تبدیلات به نرم‌افزاری با کارآیی بالا نیاز خواهند داشت.

در نمودارهای معمولی جریان داده‌ها، جریان کنترل و روند رویدادها به‌صورت واضح و آشکار نمایان نمی‌شوند. در حقیقت مهندس نرم‌افزار برای حذف کردن نشانهٔ مربوط به روند کنترل از نمودار جریان داده‌ها احتیاط می‌کند. این نوع حذف هنگامی که موارد کاربرد بلادرنگ مدنظر می‌باشند به‌طور کلی ممنوع است و به همین دلیل طرح خاصی برای نشان دادن روند رویدادها و فرآیند کنترل، توسعه یافته است. با ادامه دادن به اصول تصویب شده برای نمودارهای جریان داده، جریان داده با استفاده از یک پیکان یک تکه‌ای نشان داده خواهد شد. با این وجود روند (جریان) کنترل[1] با استفاده از یک پیکان هاشوردار یا نقطه‌چین شده نشان داده خواهد شد. فرآیندی که فقط بر روند کنترل نظارت دارد، و فرآیند کنترل[2] نامیده می‌شود با استفاده از دایره‌ها (حباب‌های) نقطه‌چین شده نمایش داده می‌شود.



شکل ۱۲-۱۳ جریان داده و کنترل
با استفاده از علائم وارد و ملور[WAR85]

جریان (روند) کنترل را می‌توان به‌صورت مستقیم به فرآیند قراردادی یا فرآیند کنترل مرتبط نمود. شکل ۱۲-۱۳ جریان کنترل و پردازش را مشابه با آنچه که از طریق طرح وارد و ملور نمایش داده می‌شود، نشان داده است. این نمودار تصویر سطح بالای داده‌ها و گردش (روند) کنترل را برای یک واحد کارخانه را نشان می‌دهد. بعد از قرار گرفتن اجزاء مونتاژ شده توسط ربات بر روی گیره‌های موردنظر، بیت وضعیت، در

1.Control flow
2.Control process

داخل میانگیر وضعیت بخش‌ها (حافظهٔ کنترل) که نشانگر وجود یا عدم وجود هر یک از اجزاء می‌باشد، تنظیم می‌گردد. اطلاعات مربوط به رویدادها که در میانگیر وضعیت قرار دارند به‌عنوان یک رشتهٔ بیتی به فرآیند گیرهٔ نمایشگر و رابط اپراتور[1] انتقال داده می‌شوند. این فرآیند فقط هنگامی می‌تواند فرمان اپراتور را بخواند که اطلاعات کنترل، یعنی رشتهٔ بیتی، نشان دهد که تمام گیره‌ها دارای اجزاء لازم هستند. پرچم رویدادها، یعنی پرچم شروع/ توقف، به بخش کنترل اولیهٔ ربات فرستاده می‌شود و این فرآیند کنترل امکان پردازش فرمان‌های بیشتر را فراهم می‌سازد. سایر جریان‌های داده‌ای در نتیجهٔ فرآیند فعال‌سازی فرآیند که برای پردازش فرمان‌های ربات فرستاده می‌شود، پدید می‌آیند.

در بعضی مواقع نمونه‌های چندگانهٔ کنترل مشابه یا فرآیند مشابه تبدیل داده‌ها ممکن است در یک سیستم بلادرنگ پدید آید. و این حالت در محیط چند کاری هنگامی که کارها (وظایف) در نتیجهٔ پردازش داخلی یا رویدادهای خارجی زیاد می‌شوند، به‌وجود می‌آید. به‌عنوان مثال تعدادی از میانگیرهای بخش وضعیت را می‌توان به‌طور هم‌زمان کنترل نمود، بنابراین می‌توان در زمان مناسب ربات‌های مختلفی را سیگنال‌دهی نمود. علاوه بر آن هر یک از ربات‌ها خود نیز دارای سیستم کنترل ربات می‌باشند. طرح وارد و ملور نمونه‌های چندگانهٔ معادل[2] را نشان می‌دهد که حباب‌های پردازشی را برای نشان دادن تعدّد و کثرت جایگزاری می‌کنند.

۱۲-۴-۴ فعالیتهای تکمیلی "هتلی" و "پیربهای"

هتلی و پیربهای[HAT87][3] در کارهایی که برای تکمیل کردن طرح اصلی تحلیل ساخت یافته انجام داده‌اند بر ایجاد نمادهای گرافیکی بیشتر تأکید کمتری داشته‌اند و بیشتر به مشخصات جنبه‌های کنترل شدهٔ نرم‌افزار توجه داشته‌اند. در این‌جا نیز از پیکان خط‌چین شده برای نشان دادن جریان رویدادها یا کنترل استفاده شده است. بر خلاف وارد و ملور پیشنهاد هتلی و پیربهای آن است که طرح‌های ممتد و نقطه‌چین شده باید به‌صورت مجزا نمایش داده شوند. بنابراین نمودار جریان کنترل[4] (CFD) تعریف می‌شود. CFD دارای فرآیندهایی مشابه با فرآیندهای DFD می‌باشد، اما به‌جای نشان دادن جریان داده‌ها به نمایش دادن فرآیند (جریان) کنترل می‌پردازد. در نمودار جریان کنترل (CFD) به‌جای نمایش دادن فرآیندهای کنترل به‌صورت مستقیم در مدل جریان از طرح مربوط به مشخصات کنترل (ستون ممتد) استفاده می‌شود، در صورت لزوم می‌توان ستون ممتد را از حالت یک پنجره به یک "بخش اجرایی" (مشخصات کنترل) تبدیل نمود که این بخش اجرایی فرآیندهای نمایش داده شده در DFD براساس

1.monitor fixture and operator interface

2.multiple equivalent instances

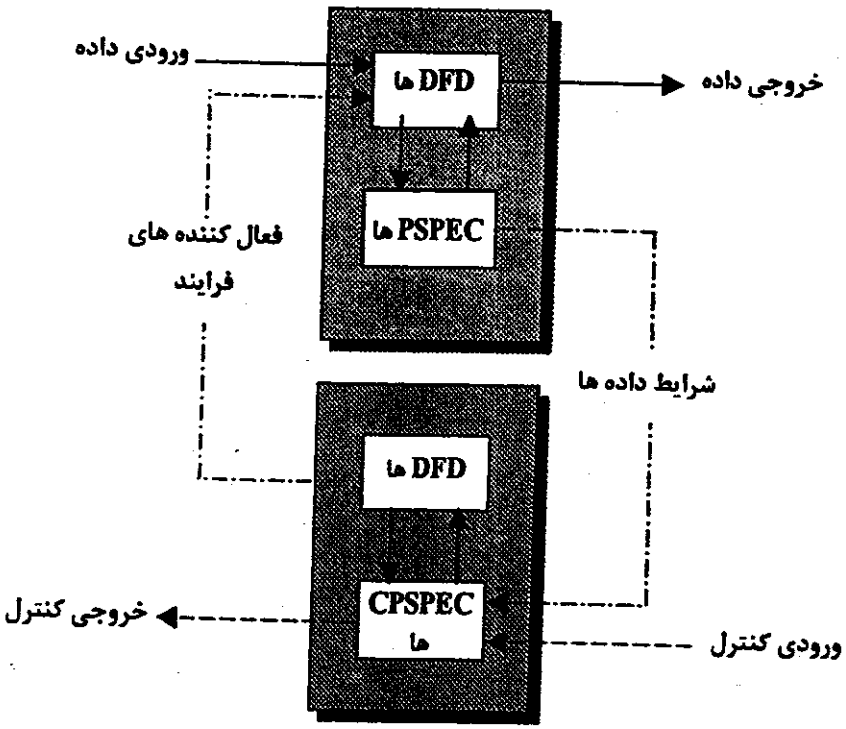
3.Hatley , D.J. and I.A.

4.control flow diagram

رویدادهایی که از طریق پنجره انتقال داده می‌شوند را کنترل می‌کند. مشخصات کنترل[1] (CSPEC)، که به‌صورت کامل در بخش ۱۲-۶-۴ توضیح داده شد، برای نشان دادن موارد زیر به‌کار برده می‌شوند:

۱- نحوهٔ عملکرد نرم‌افزار هنگام دریافت یک رویداد یا سیگنال کنترل

۲- تعیین نوع فرآیندهایی که در نتیجه به‌وقوع پیوستن یک رویداد ایجاد می‌شوند. مشخصات فرآیند برای توصیف کارهای داخلی فرآیند نمایش داده شده در نمودار جریان به‌کار برده می‌شوند.



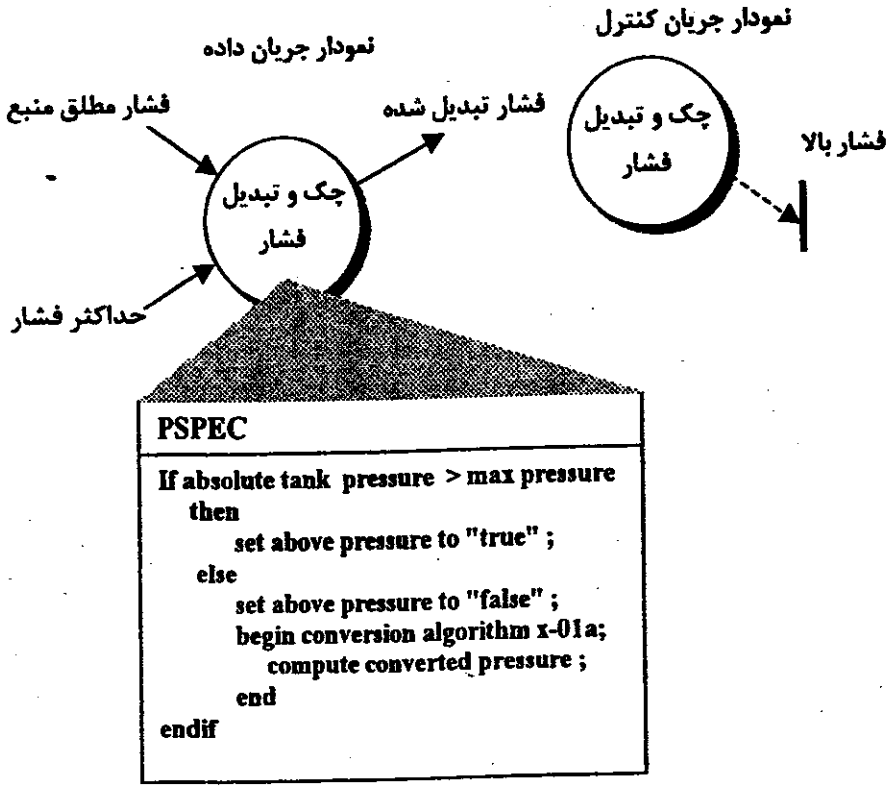
شکل ۱۲-۱۴ رابط میان مدلهای داده‌ای و کنترلی

هتلی و پیربهای با استفاده از طرح توضیح داده شده در نمودارهای ۱۲-۱۲ و ۱۲-۱۳ و استفاده از اطلاعات اضافی موجود در مشخصات کنترل (CSPEC) و مشخصات پردازشی (PSPEC) مدلی را در زمینهٔ سیستم بلادرنگ به‌وجود آورده‌اند. نمودارهای جریان داده‌ها برای نشان دادن داده‌ها و فرآیند به‌وجودآورندهٔ این داده‌ها مورد استفاده قرار می‌گیرند. نمودارهای جریان کنترل نیز نحوهٔ جریان رویدادها درمیان فرآیندهای مختلف و رویدادهای خارجی فعال‌کنندهٔ فرآیندهای متنوع را نشان می‌دهند. رابطهٔ داخلی موجود میان مدل‌های پردازشی و کنترل به‌صورت طرح کلی در نمودار ۱۲-۱۴ نشان داده شده است. مدل پردازشی از طریق شرایط داده‌ای به "مدل کنترل" متصل می‌شود و با آن ارتباط برقرار می‌کند.

1.control specification

مدل کنترل نیز از طریق فرآیند فعال‌سازی اطلاعات موجود در مشخصات کنترل با مدل پردازشی ارتباط برقرار می‌کند.

یک شرط داده‌ای[1] هنگام وارد شدن داده به فرآیند به‌وجودآورندهٔ خروجی کنترل، پدید می‌آید. این وضعیت در نمودار ۱۲-۱۵ نشان داده شده است. در این نمودار بخشی از مدل جریان سیستم کنترل و نظارت خودکار مجراهای فشار در تصفیهٔ روغن به تصویر کشیده شده است. کنترل فرآیند و تبدیل فشار[2]، الگوریتم توصیف شده در طرح شبه برنامهٔ مشخصات پردازشی را پیاده می‌کنند. اگر فشار کل مخزن بیشتر از حداکثر مقدار مجاز آن باشد، رویداد فشار بالا نیز به‌وجود می‌آید. توجه داشته باشید که هنگام استفاده از طرح هتلی و پیربهای گردش داده‌ها به‌عنوان بخشی از DFD نمایش داده می‌شود و این در حالی است که گردش کنترل به‌صورت مجزا و به‌عنوان بخشی از نمودار جریان کنترل نمایان می‌شود. همان‌طور که در قسمت‌های قبلی نیز توضیح داده شد ستون عمودی ممتد که رویداد فشار بالا به سمت آن جریان می‌یابد اشاره‌گر مشخصات کنترل (CSPEC) می‌باشد.



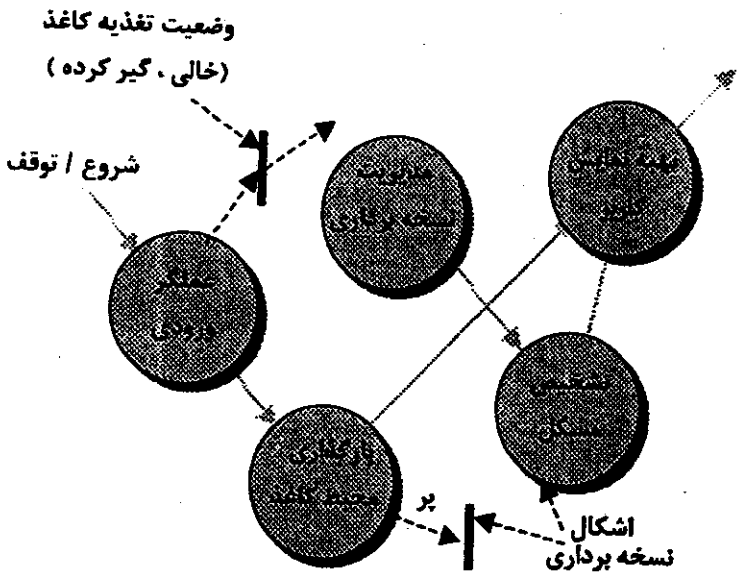
شکل ۱۲-۱۵ شرایط داده

بنابراین برای تعیین این مسأله که هنگام به‌وقوع پیوستن این رویداد چه چیزی اتفاق می‌افتد باید مشخصات کنترل (CSPEC) را چک کنیم.

1.data condition

2.check and convert

مشخصات کنترل حاوی تعدادی از ابزار مهم مدل‌سازی هستند. از جدول فعال‌سازی فرآیند[1]، (پردازشی) برای نشان دادن فرآیندهایی به‌واسطهٔ رویداد مفروض فعال شده‌اند، استفاده می‌شود. به‌عنوان مثال جدول فعال‌سازی فرآیند (PAT) برای نمودار ۱۲-۱۵ می‌تواند نشانگر این امر باشد که رویداد **فشار بالا** به‌وجودآورندهٔ فرآیند کاهش فشار منبع می‌باشد، علاوه بر PAT (جدول فعال‌سازی فرآیند)، مشخصات کنترل (CSPEC) حاوی نمودار تغییر حالت (STD) هستند. نمودار تغییر حالت (STD) نوعی مدل رفتاری است که به تعریف مجموعه‌ای از حالات سیستم وابسته است و تعریف این نمودار در بخش‌های بعدی ارائه خواهد شد.



شکل ۱۲ـ۱۶ سطح یک CDF (نمودار جریان کنترل) برای نرم افزار نسخه برداری

۱۲-۵ مدل سازی رفتاری

مدل‌سازی رفتاری[2] به‌عنوان یک اصل مهم عملیاتی در تمام شیوه های تحلیل نیازمندی‌ها مطرح می‌باشد. اما در حال حاضر فقط نسخه‌های تکمیل شدهٔ تحلیل ساخت یافته([WAR85] [3] و [HAT87][4]) علائمی را برای این نوع مدل‌سازی، یعنی مدل‌سازی رفتاری فراهم کرده است. نمودار تغییر حالت با نشان دادن حالات سیستم و رویدادهایی که موجب تغییر حالت سیستم می‌شوند، رفتار یک

عکس العمل نرم افزار در قبال رویدادهای خارجی را چگونه باید مدل کنیم؟

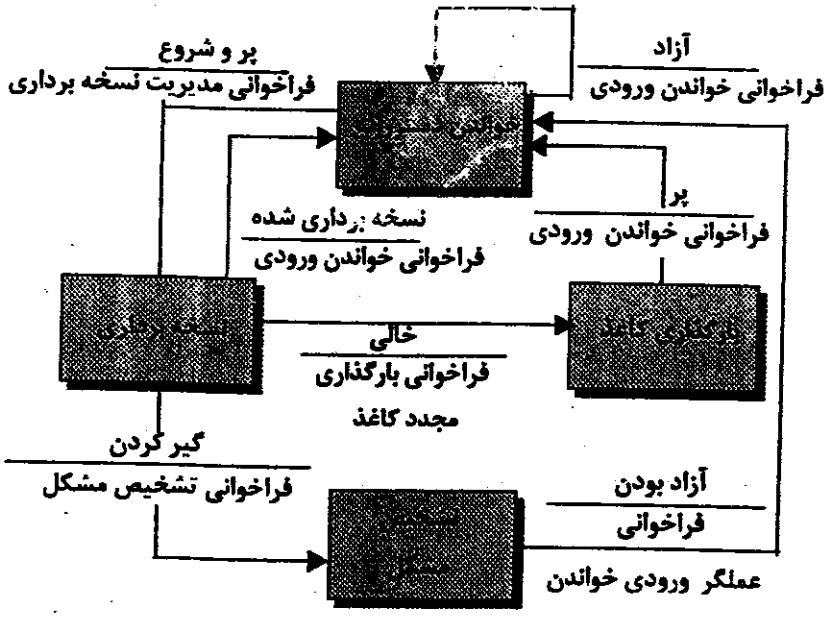
1.process activation
2.Behavioral modeling
3.Ward,P.T. and S.J.
4.Hatley , D.J. and I.A.

سیستم را نشان می‌دهد. علاوه بر آن نمودار تغییر حالت (STD) نشان می‌دهد که در صورت وقوع یک رویداد خاص باید چه اقداماتی انجام شود.

حالت سیستم عبارت است از هرگوه حالت رفتاری قابل مشاهده، به‌عنوان مثال حالات مربوط به سیستم کنترل و نظارت برای مجراهای فشار که در قسمت ۱۲-۴-۴ توضیح داده شده عبارتند شامل حالت نظارت، حالت اعلام‌نظر، حالت تخلیهٔ فشار و غیره باشد. هر یک از این حالات می‌توانند نشانگر وضعیتی از رفتار سیستم باشند. نمودار تغییر حالت نحوهٔ برخورد سیستم در هنگام تغییر از حالتی به حالت دیگر را نشان می‌دهد.

برای نشان دادن موارد استعمال طرح‌های رفتاری و کنترلی تکمیل شده توسط هتلی و پیربهای، نرم‌افزاری را در نظر بگیرید که در داخل دستگاه فتوکپی یک اداره قرار داده شده است. شکل سادهٔ جریان کنترل در نرم‌افزار کارگذاری شده در دستگاه فتوکپی در نمودار ۱۲-۱۶ نشان داده شده است. در این نمودار برای گویاتر شدن مسأله پیکان‌های جریان داده‌ها به‌صورت کمرنگ نقطه‌چین شده‌اند، اما واقعیت آن است که این پیکان‌ها هرگز به‌عنوان بخشی از نمودار گردش کنترل نشان داده نمی‌شوند.

جریان‌های کنترل به‌صورت فرآیندهای مجزای ورودی و خروجی نشان داده می‌شوند و ستون عمودی نیز نشانگر "پنجره" مشخصات کنترل می‌باشد. به‌عنوان مثال حالت کاغذرسانی و رویدادهای شروع/ توقف کار به سمت ستون مشخصات کنترل (CSPEC) جریان می‌یابند. و این امر بدین مفهوم است که هر یک از این رویدادها می توانند موجب فعال شدن برخی از فرآیندهای نمایش داده شده در CFD شوند. اگر ما بخواهیم بخش‌های داخلی مشخصات کنترل را مورد بررسی قرار دهیم، رویداد شروع/ توقف را می‌توانیم برای فعال کردن/ غیرفعال کردن فرآیند کنترل کپی نشان دهیم. همچنین رویداد ازکارافتادن دستگاه کپی (بخشی از حالت کاغذرسانی، تشخیص دادن مشکل عملکردی را فعال خواهد کرد). باید توجه داشت که تمام ستون‌های عمودی در داخل CFD نشانگر مشخصات کنترل یکسان می‌باشند. گردش رویداد را می‌توان به صورت مستقیم وارد یک فرآیند (پردازشی) کرد که این مسأله در مورد خطای تکثیر نشان داده شده است. با این وجود، این جریان نمی‌تواند فعال‌کنندهٔ فرآیند مذکور باشد، اما می‌تواند اطلاعات کنترل را برای فرآیند الگوریتم فراهم کند.



شکل ۱۲ـ۱۷ نمودارانتقال وضعیت برای نرم افزار نسخه برداری (فتوکپی)

نمودار ساده شدهٔ تغییر حلات برای نرم‌افزار کارگذاری شده در دستگاه فتوکپی در شکل ۱۲-۱۷ نشان داده شده است. مستطیل‌های موجود در این شکل نمایانگر حالات سیستم و پیکان‌ها، نمایانگر تغییر حالات هستند. هر یک از این پیکان‌ها نیز با یک عبارت قاعده‌مندی نام‌گذاری شده‌اند. مقادیر بالایی نشانگر رویدادهایی هستند که موجب به‌وجود آمدن تغییر شده‌اند. مقادیر پایینی در این جدول نشانگر اقداماتی است که در نتیجهٔ اتفاق افتادن یک رویداد پدید می‌آیند. بنابراین هنگامی که برگه‌دان دستگاه فتوکپی پر است و دکمهٔ شروع (start) برای انجام عمل فتوکپی فشار داده می‌شود حالت سیستم از حالت خواندن فرمان به حالت کپی گرفتن تغییر می‌کند. توجه داشته باشید که حالات با فرآیندهای موجود در تناظر یک به یک نمی‌باشند. به‌عنوان مثال حالت کپی گرفتن شامل فرآیندهای کنترل کار کپی و تولید کپی می‌باشد که در نمودار ۱۲-۱۶ نشان داده شده است.

۱۲-۶ مکانیک تحلیل ساخت یافته

در قسمت قبلی طرح‌های ابتدایی و تکمیلی تحلیل ساخت یافته را مورد بررسی قرار دادیم. برای آن‌که بتوان این طرح را به‌صورت مؤثری در تحلیل نیازمندی‌های نرم‌افزاری مورد استفاده قرار داد باید این طرح را با مجموعه‌ای از مسایل ذهنی که مهندسین نرم‌افزار را قادر می‌سازند تا یک مدل تحلیلی مناسبی را به‌وجود آورند، ادغام کنید. برای نشان دادن موارد مصرف این مسایل ذهنی از نسخهٔ تطبیق یافته کارهای تکمیلی هتلی و پیربهای در مورد طرح اولیهٔ تحلیل ساخت یافته در بقیه قسمت‌های باقی‌مانده از این فصل استفاده خواهد شد.

در قسمت‌های بعدی این فصل مراحل لازم برای تکمیل مدل‌های کامل و مناسب با استفاده از تحلیل ساخت یافته مورد بررسی قرار خواهد گرفت. در طول این بحث و بررسی از طرح توضیح داده شده در قسمت ۱۲-۴ استفاده خواهد شد و سایر طرح‌هایی که در قسمت‌های قبلی به‌صورت خلاصه به آنها اشاره شده است به‌صورت کامل در بخش‌های بعد مورد بررسی قرار خواهند گرفت.

۱۲-۶-۱ایجاد یک نمودار موجودیت / رابطه

پرسش

برای ساخت یک نمودار موجودیت / رابطه (ERD) چه گامهایی باید برداشته شود؟

نمودار موجودیت / رابطه به مهندسین نرم‌افزاری این امکان را می‌دهد که اشیاء داده‌ای ورودی و خروجی سیستم را به‌طور کامل مشخص و تعیین کنند. همچنین این نمودار امکان تعیین ویژگی‌های مشخص‌کنندهٔ این اشیاء و روابط آنها را برای مهندسین نرم‌افزار میسر می‌سازد. ERD نیز همانند بسیاری از عناصر مدل تحلیلی به روش تکراری به‌وجود آمده است. روش‌های زیر برای به‌وجود آمدن ERD طی شده است:

۱- در طول فراخوانی نیازمندی‌ها از مشتریان خواسته می‌شود تا از لیستی از مواردی را که در روند کار وجود دارد تهیه نمایند. این موارد نیز به لیستی از اشیاء داده‌ای ورودی و خروجی و موجودیت‌های خارجی تولیدکننده یا مصرف‌کنندهٔ اطلاعات تبدیل می‌شود.

۲- مشتری و فرد تحلیل‌گر با در نظر گرفتن اشیاء و استفاده از آنها در یک زمان مشخص تعیین می‌کنند که آیا رابطه‌ای (مرحلهٔ بدون نام) میان شئ داده‌ای و سایر اشیاء وجود دارد یا خیر؟

۳- در هر زمان و هر مرحله‌ای که رابطه‌ای وجود داشته باشد، مشتری و تحلیل‌گر یک یا چند جفت رابطه ـ موجودیت را ایجاد می‌کنند.

۴- و برای هر یک از جفت‌های رابطه- شئ، تعداد مدخل‌های رابطه‌ای و مدالیته مشخص می‌شود.

۵- مراحل ۲ تا ۴ به‌صورت متناوب و تکراری انجام می‌شوند تا تمام جفت‌های رابطه ـ شئ به‌طور کامل تعیین شوند. با ادامه یافتن این روند این احتمال وجود دارد که موارد حذف شده نیز مشخص و آشکار شود. با افزایش تعداد تکرارها، اشیاء و روابط جدیدی به‌صورت تغییرناپذیر اضافه خواهند شد.

۶- صفات خاصه هر یک از موجودیت‌ها (اشیاء) را باید تعریف و مشخص نمود.

۷- نمودار موجودیت / رابطه رسمی شده و مجدداً مورد بررسی قرار می‌گیرد.

۸- مراحل ۱ تا ۷ تکرار می‌شوند تا آن‌که مدل‌سازی داده‌ها تکمیل شود.

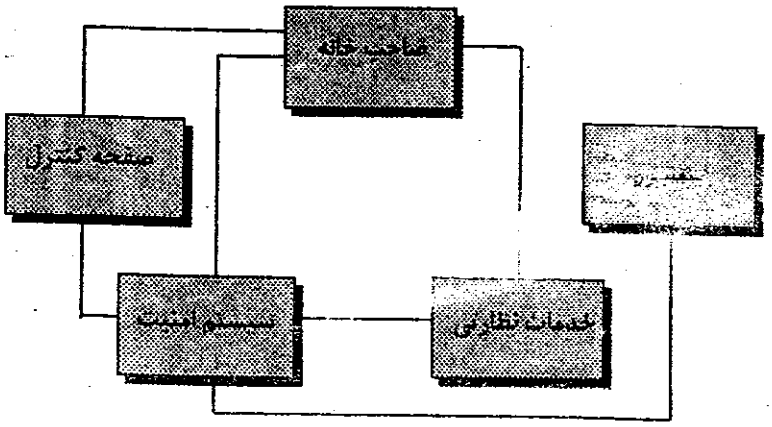
برای نشان دادن موارد مصرف و کاربرد این دستورالعمل‌های اولیه از مثال سیستم‌های امنیتی خانهٔ امن، که در فصل ۱۱ مورد بررسی قرار داده شده، استفاده خواهد شد. با مراجعه کردن به روش روایت پردازش برای خانهٔ امن (به قسمت ۱۱-۳-۳ مراجعه شود) لیستی از "موارد" ذیل با این مسأله در ارتباط هستند:

- صاحب‌خانه (home owner)
- صفحهٔ کنترل (contol panel)
- حسگرها (sensors)
- سیستم‌های حفاظتی(security systems)
- خدمات نظارت‌کننده(monitoring service)

با در نظر گرفتن این "موارد" و موارد فوق‌الذکر به‌صورت همزمان، روابط موجود مشخص خواهد شد. برای تکمیل نمودن این کار باید هر یک از اشیاء را رسم نمود و خطوط ارتباطی اشیاء را مشخص کرد. به‌عنوان مثال با مراجعه به نمودار ۱۲-۱۸ مشخص خواهد شد که یک رابطهٔ مستقیم میان صاحب‌خانه و صفحهٔ کنترل، سیستم امنیتی و خدمات نظارت و کنترل وجود دارد. و یک رابطه نیز میان سنسورها و سیستم‌های امنیتی مشخص خواهد شد.

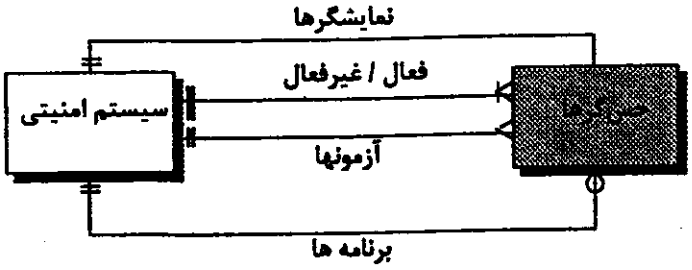
بعد از تعریف کردن تمام روابط موجود، یک یا جفت رابطه - شئ برای هر رابطه مشخص خواهد شد. به‌عنوان مثال وجود رابطه میان سنسورها و سیستم‌های امنیتی مشخص می‌کند که این رابطه دارای جفت‌های رابطه - شئ به‌شرح زیر می‌باشد.

- سیستم‌های امنیتی بر سنسورها نظارت دارند.
- سیستم‌های امنیتی باعث فعال شدن و یا عدم فعالیت سنسورها می‌گردند.
- سیستم‌های امنیتی ، سنسورها را مورد آزمون قرار می‌دهند.
- سیستم‌های امنیتی برنامه‌ریزی‌کننده سنسورها هستند.



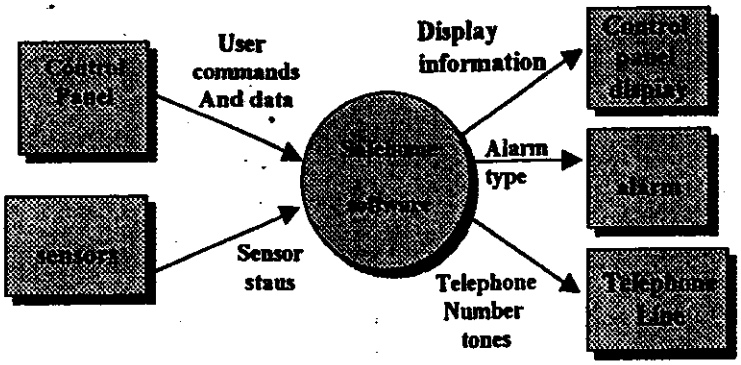
شکل ۱۲-۱۸ برقراری ارتباطات

هر یک از جفت‌های رابطه ـ شئ فوق‌الذکر برای تعیین تعداد مدخل‌های رابطه‌ای و مدالیته مورد تحلیل قرار می‌گیرند. به‌عنوان مثال با در نظر گرفتن یک جفت رابطه ـ شئ مانند موردی که سیستم‌های حفاظتی بر سنسورها نظارت دارند، تعداد مدخل‌های رابطه‌ای موجود میان سیستم‌های امنیتی و سنسورها، ۱ به بی‌شمار (۱:N) خواهد بود. مدالیته نیز شامل یکبار وقوع سیستم‌های امنیتی و حداقل یکبار وقوع سنسورها همان‌طور که در نمودار ۱۲-۱۹ نشان داده شده است، تغییر خواهد یافت و اصلاح می‌شود. تحلیل‌های مشابهی در مورد سایراشیاء داده‌ای اعمال خواهد شد.

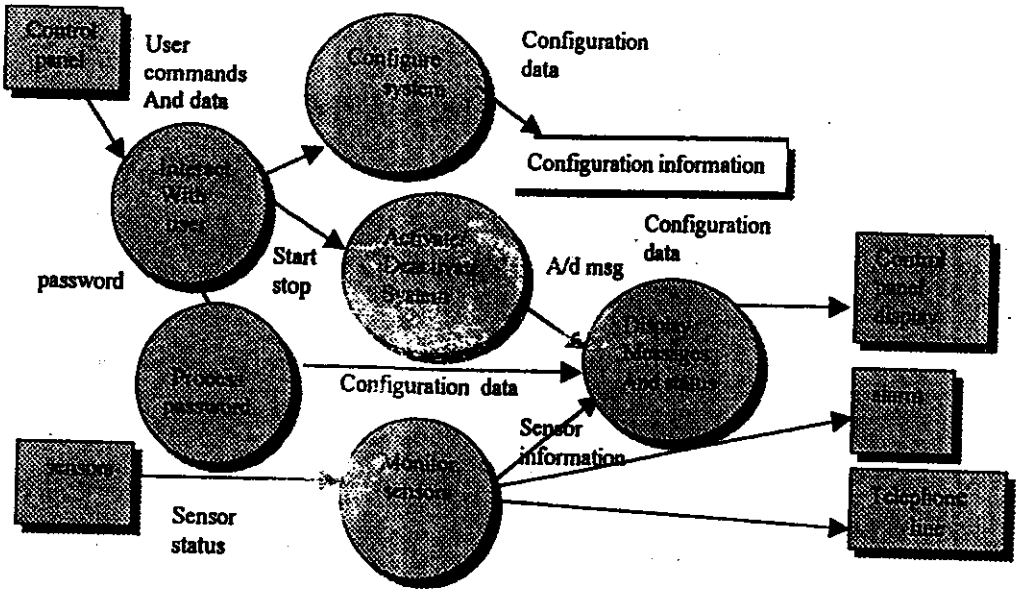


شکل ۱۲-۱۹ ساخت و ارتباطات و کاردینالیته / مدالیته

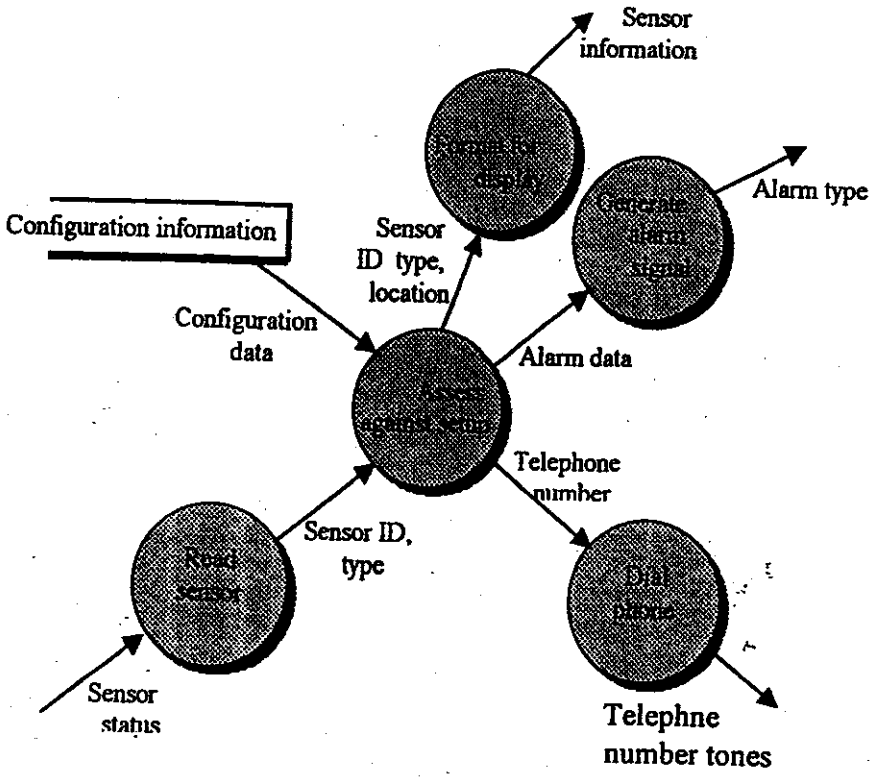
برای تعیین صفات خاصه هر یک از اشیاء، این اشیاء به صورت مجزا مورد بررسی قرار خواهند گرفت. از آن‌جایی‌که ما برای این کار نرم‌افزار حمایت‌کننده از خانهٔ امن را درنظر می‌گیریم. بنابراین صفات خاصه نیز باید برداده‌هایی تأکید داشته باشند که باید برای فراهم کردن امکان راه‌اندازی سیستم ذخیره شوند. به‌عنوان مثال شئ حس‌گر (سنسور) باید دارای صفات خاصه ذیل باشد: نوع حس‌گر، شمارهٔ شناسایی داخلی، نشانی منطقه و سطح اعلان خطر.



شکل ۱۲ـ۲۰ نمودار DFD سطح متن



شکل ۱۲ـ۲۱ سطح DFD (نمودار جریان داده) برای خانه امن



شکل ۱۲-۲۲ سطح دوم DFD (نمودار جریان داده ها) که فرآیند ناظر بر حس گرها را پالایش می‌کند

۱۲-۶-۲ ایجاد یک مدل جریان داده ها

پرسش

آیا رهنمودهای مناسبی برای ساخت DFD وجود دارند؟

نمودار گردش داده‌ها (DFD) برای مهندسین نرم‌افزار این امکان را فراهم می‌آورد که مدل‌های اطلاعاتی و حوزه کارکرد را به‌طور همزمان تکمیل کنند. پس از آن‌که نمودار جریان داده‌ها به سطوح و بخش‌های جزیی‌تری تغییر یافت، تحلیل‌گر کار تجزیهٔ ضمنی وظایف سیستم را انجام می‌دهد و بدین‌ترتیب اصل چهارم تحلیل عملیات را برای وظایف موجود تکمیل می‌کند. همزمان با انجام این کار تکمیلی پالایش انجام شده در DFD موجب به‌وجود آمدن پالایش متناظر در داده‌ها می‌شود. چند دستورالعمل ساده می‌تواند در طی اشتقاق نمودار جریان داده‌ها مفید و مؤثر واقع شود. این دستورالعمل‌ها عبارتند از:

۱- سطح 0 نمودار جریان داده‌ها باید نرم‌افزار/ سیستم را به‌صورت تک حبابی نشان دهد؛ ۲- ورودی خروجی اصلی باید به دقت مورد بررسی قرار گیرند؛ ۳- پالایش باید با جداسازی فرآیندهای کاندیدا، اشیاء داده‌ای و با جداسازی برنامه‌های ذخیره شده‌ای که قرار است در سطح بعدی ظاهر شوند، آغاز گردد؛ ۴- تمام پیکان‌ها و حباب‌ها باید به‌وسیله اسامی با مفهوم و پرمعنا نام‌گذاری شوند؛ ۵- استمرار جریان اطلاعات باید از سطحی به سطح دیگر حفظ شود؛ ۶- یک حباب باید در زمان مناسب پالایش شود. نوعی گرایش طبیعی برای پیچیده‌سازی نمودار جریان داده‌ها وجود دارد. و این حالت زمانی اتفاق می‌افتد که فرد تحلیل‌گر سعی دارد جزییات بیش از حد را زودتر از موعد مقرر نشان دهد و یا سعی دارد جنبه‌های رویه‌ای (عملی) نرم‌افزار را به‌جای نمودار گردش اطلاعاتی نمایش دهد.

با در نظر گرفتن مجدد محصول خانهٔ امن ، سطح 0 نمودار جریان داده‌ها برای این سیستم در نمودار ۱۲-۲۰ نمایش داده می‌شود. موجودیت‌ها (اشیاء) اولیهٔ داخلی، اطلاعات را برای مصرف سیستم به‌وجود می‌آورند و از اطلاعات تولید شده توسط سیستم استفاده می‌کنند. پیکان‌های نام‌گذاری شده نمایانگر اشیاء داده‌ای یا سلسله مراتب نوع اشیاء داده‌ای هستند. به‌عنوان مثال فرمان‌های کاربر و داده‌ها می‌توانند تمام فرمان‌های پیکربندی، تمام فرمان‌های فعال‌سازی/ غیرفعال‌سازی، تمام فعل و انفعالات متفرقه، و تمام داده‌هایی را پوشش دهند که برای واجدالشرایط ساختن و یا تکمیل یک فرمان وارد سیستم شده‌اند.

اکنون سطح 0 نمودار گردش داده‌ای به مدل سطح ۱ ارتقاء یافته است، اما میزان پیشرفت ما چگونه بوده است؟ هنوز نیز یک روش ساده و مؤثر در این زمینه آن است که یک "تجزیهٔ دستوری" در گزارش پردازشی توصیف‌کنندهٔ حباب سطح متنی انجام شود. یعنی در این تجزیه ما تمام اسم‌ها (و عبارات اسمی) و فعل‌ها (گروه‌های فعلی) را در گزارش خانهٔ امن ، که در فصل ۱۱ توضیح داده است، از یکدیگر مجزا کنیم. برای نشان دادن این تجزیه ما مجدداً گزارش پردازشی را ایجاد می‌کنیم و در این گزارش زیر اولین وقوع تمام اسم‌ها خط می‌کشیم (اولین وقوع تمام فعل‌ها را با نوشتن این فعل‌ها به‌صورت مورب نشان می‌دهیم.[1]

---

۱ باید توجه داشت که اسمها و فعلهای که مترادفند یا معنای مشابهی در مدل سازی فرآیند دارند، در نظر گرفته نخواهند شد.

نرم‌افزار خانهٔ امن صاحب دستگاه را قادر می‌سازد تا سیستم‌های امنیتی را پس از نصب شدن آنها پیکربندی کند، تمام سنسورها (حس‌گرها) مربوط به سیستم امنیتی را کنترل کند و بر آنها نظارت داشته باشد، و با استفاده از صفحه کلید کمکی و کلیدهای تابع موجود در صفحهٔ کنترل نرم‌افزار خانهٔ امن که در نمودار ۱۱-۲ نشان داده شده است با صاحب سیستم تعامل داشته باشد.

در طول کار نصب صفحهٔ کنترل خانهٔ امن برای برنامه‌نویسی و پیکربندی سیستم مورد استفاده قرار می‌گیرد. به هر یک از سنسورها (گیرنده‌ها) یک شماره و یک نوع نسب ـ داده می‌شود، یعنی یک اسم رمز در برنامه‌ریزی برای فعال کردن و غیرفعال کردن سیستم تعیین می‌شود. شماره تلفن‌ها نیز برای برقرار تماس به هنگام وقوع رویداد گیرنده‌ای، وارد سیستم شده‌اند.

هنگام رخ دادن یک رویداد گیرنده‌ای و شناخته شدن آن توسط سیستم ن نرم‌افزار آژیر وصل شده به سیستم را به صدا درمی‌آورد. بعد از یک وقفهٔ کوتاه که مدت آن توسط صاحب سیستم در طول کارهای پیکربندی مشخص شده است، نرم‌افزار با شماره تلفن بخش خدمات نظارتی و کنترل ماس برقرار می‌کند و اطلاعاتی را در مورد موقعیت سیستم در اختیار این بخش قرار می‌دهد و در مورد موجودیت رویداد شناخته شده توسط سیستم گزارش می‌دهد. شمارهٔ تلفن هر ۲۰ ثانیه یکبار گرفته می‌شود تا آن‌که تماس تلفنی برقرار شود.

تمام ارتباطات موجود به کمک نرم‌افزار خانهٔ امن به‌وسیلهٔ سیستم فرعی ارتباط کاربر کنترل می‌شود این سیستم اطلاعات ورودی فراهم شده از طریق صفحه کلید فرعی و کلیدهای تابع را می خواند، پیغام‌های پر اهمیت را بر روی صفحهٔ LCD نمایش می‌دهد، و وضعیت اطلاعاتی سیستم را بر روی صفحهٔ LCD نمایش می‌دهد، ارتباط از طریق صفحه کلید به‌صورت زیر حاصل می‌شود:

با مراجعه کردن به "تجزیهٔ دستوری" یک الگو به‌وجود می‌آید که در این الگو تمام افعال حاصل فرآیندهای خانهٔ امن هستند، یعنی این افعال ممکن است در نهایت به‌صورت یک حباب در نمودارهای بعدی جریان داده‌ها ظاهر شوند و تمام افعال به‌صورت موجودیت‌ها (جعبه‌های) داخلی، اشیاء داده‌ای یا کنترلی (پیکان‌ها) و یا به‌صورت ذخیرهٔ داده‌ای (خطوط موازی) هستند. توجه داشته باشید که می‌توان افعال و اسامی موجود در این تجزیه را به یکدیگر متصل نمود (به‌عنوان مثال به هر یک از گیرنده‌ها(سنسورها) اسم و نوع خاصی نسبت داده می‌شود). بنابراین با انجام تجزیهٔ دستوری در گزارش پردازشی برای تعیین حباب‌ها در هر یک از سطوح نمودار جریان داده‌ها، می‌توان اطلاعات مفید بیشتری را در مورد نحوهٔ پیش رفتن به سطح بعدی با انجام اصلاحات لازم، به‌وجود آورد. با استفاده از این اطلاعات می‌توان سطح ۱ نمودار گردش داده‌ها را همان‌طور که در نمودار ۱۲-۲۱ نشان داده شده است، نمایش داد. جریان سطح متنی که در نمودار ۱۲-۲۰ نشان داده شده است به هفت فرآیند جریان نشأت گرفته از بررسی تجزیهٔ دستوری تقسیم شده است. هم‌چنین جریان اطلاعات میان فرآیندهای سطح ۱، از این تجزیه نشأت گرفته است.

باید توجه داشت که استمرار جریان اطلاعات میان سطح ۱ و سطح ۰ حفظ می‌شود. توضیح مربوط به محتویات ورودی و خروجی در سطوح ۰ و ۱ نمودار جریان داده‌ها به بخش ۱۲-۷ موکول خواهد شد.

فرآیندهای مشخص شده در سطح ۱ نمودار جریان داده‌ها را می‌توان به دو سطح پایین‌تر پالایش نمود. به‌عنوان مثال فرآیند سنسورهای نظارت‌کننده را می‌توان همان‌طور که در نمودار ۱۲-۲۲ نشان داده شده است به سطح ۲ نمودار جریان داده‌ها تبدیل نمود. توجه داشته باشید که استمرار جریان اطلاعات، میان سطوح حفظ می‌شود.

۱۲-۶-۳ ایجاد یک مدل جریان کنترل

در بسیاری از انواع موارد کاربردی پردازش داده‌ها، مدل داده‌ای و نمودار جریان داده‌ها دقیقاً همان موارد ضروری برای دستیابی به بینش پر معنا در مورد نیازمندی‌های نرم‌افزاری هستند. اما همان‌طور که قبلاً نیز به آن اشاره شد انواع مختلفی از موارد کاربردی وجود دارند که از رویدادها منتج می‌شوند و نه از داده‌ها و این حالت موجب به‌وجود آمدن اطلاعات کنترل به‌جای گزارشات می‌شود و این اطلاعات پردازشی به دقت و عملکرد بیشتری نیاز دارند. چنین موارد کاربردی علاوه بر مدل‌سازی جریان داده‌ها به مدل‌سازی جریان کنترل نیز نیاز خواهند داشت.

طرح گرافیکی مورد نیاز برای ایجاد نمودار جریان کنترل در بخش ۱۲-۴-۴ ارائه شده است. برای بررسی اجمالی روش موجود برای ایجاد نمودار جریان کنترل (CFD) می‌توان گفت که مدل جریان داده‌ها از مجموع تمام پیکان‌های گردش داده‌ای حاصل می‌شود. سپس رویدادها و موارد کنترل (پیکان‌های خط‌چین شده) نیز به این نمودار اضافه می‌شود و یک پنجره (یک ستون عمودی) نیز به مشخصات کنترل نشان داده شده، اضافه می‌گردد. اما چگونه می‌توان رویدادها را انتخاب نمود؟

ما کاملاً به این مسأله واقف هستیم که یک رویداد یا مورد کنترل به شکل ارزش دوارزشی (به‌عنوان مثال درست یا غلط، خاموش یا روشن، ۱ یا ۰) یا به شکل لیست ناپیوسته‌ای از شرایط (خالی یا پر) اجرا می‌شود. برای انتخاب بهترین رویداد کاندید، دستورالعمل‌های زیر توصیه می‌شوند:

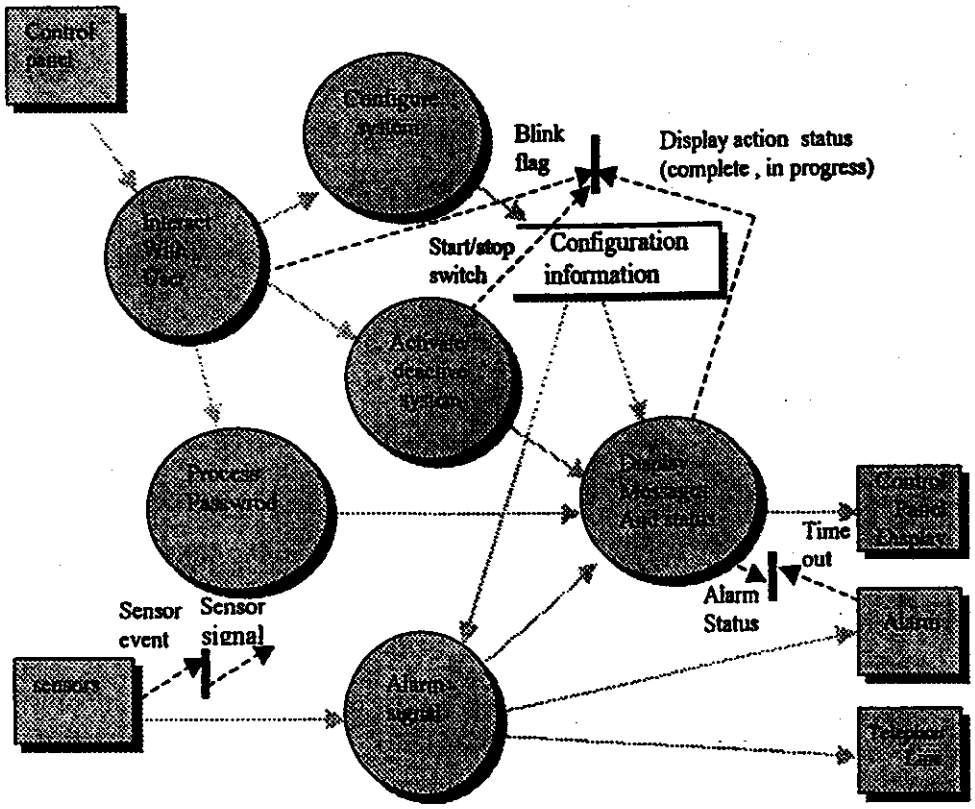
- لیستی از تمام سنسورها (گیرنده‌هایی) که توسط نرم‌افزار خوانده می‌شوند، تهیه کنید.
- لیستی از تمام شرایط وقفه تهیه کنید.
- لیستی از تمام کلیدهایی که توسط نرم‌افزار خوانده می‌شوند، تهیه کنید.
- لیستی از تمام کلیدهایی که توسط اپراتور به کار می‌افتند را تهیه کنید.
- مجدداً از تجزیهٔ فعلی ـ اسمی که در مورد گزارش پردازشی به‌کار برده شد، استفاده کنید. تمام "موارد کنترل" و ورودی‌ها و خورجی‌های CSPEC (مشخصات کنترل) را تا حد ممکن مورد بررسی قرار دهید.

- با مشخص کردن حالات سیستم، رفتار آن را مشخص کنید. نحوهٔ به‌وجود آمدن هر یک از حالات را مشخص کنید و تغییر حالات موجود را تعریف کنید.
- بر هر یک از موارد احتمال حذف تأکید کنید و به آنها توجه داشته باشید ـ یعنی برخطای متعارف در کار تعیین کنترل توجه داشته باشید. به‌عنوان مثال این سؤال را مطرح کنید. "آیا روش دیگری برای رسیدن به این حالت و خارج شدن از این حالت وجود دارد یا خیر؟"

سطح ۱ نمودار گردش کنترل برای نرم‌افزار خانهٔ امن در نمودار ۱۲-۲۳ نشان داده شده است. از میان رویدادها و موارد کنترل ذکر شده می‌توان به **رویداد گیرنده (سنسور)**، **پرچم چشمک‌زن** (علامتی برای چشمک زدن صفحه نمایش LCD) و **کلید شروع/ توقف** (علامتی برای روشن یا خاموش کردن سیستم) اشاره نمود. هنگامی که رویدادی از محیط خارج به پنجرهٔ مشخصات کنترل (CSPEC) وارد می‌شود، بدین مفهوم خواهد بود که مشخصات کنترل یک یا چند فرآیند پردازشی نشان داده شده در نمودار جریان کنترل را فعال خواهند کرد. حالتی که در آن یکی از موارد کنترل از یک فرآیند پردازشی حاصل می‌شود و به سمت پنجره مشخصات کنترل جریان می‌یابد. نشان‌گر این امر است که کنترل و فعال‌سازی سایر فرآیندها و موجودیت (اشیاء) خارجی نیز اعمال شده است.

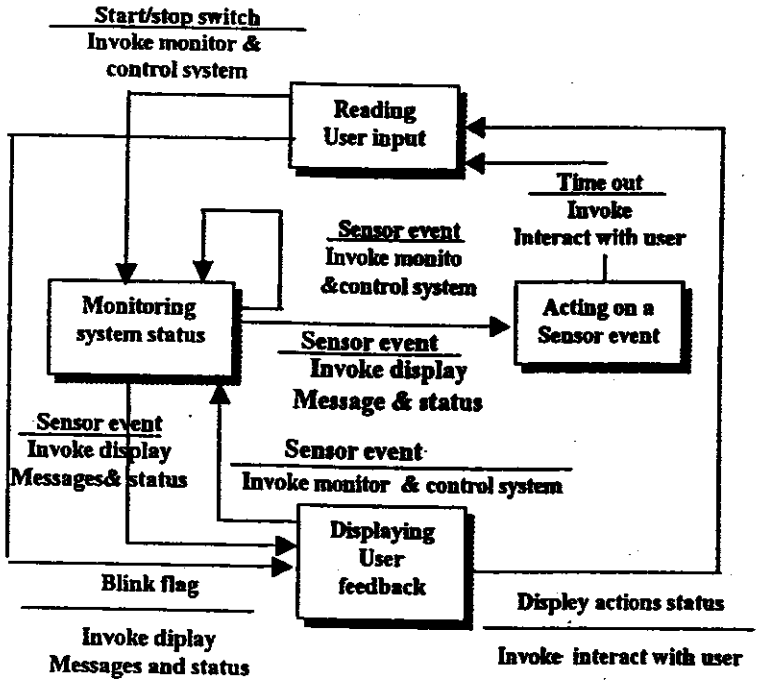
۱۲-۶-۴ تعیین مشخصات کنترل

مشخصات کنترل (CSPEC) نشان‌گر رفتار سیستم (در سطح نشأت گرفته از آن) به دو روش متفاوت هستند. مشخصات کنترل حاوی نمودار تغییر حالت (STD) هستند که این نمودار نشان‌گر مشخصات ترتیبی رفتار می‌باشد. همچنین مشخصات کنترل می‌توانند دارای جدول فعال‌سازی برنامه‌ای ـ یعنی مشخصات ترکیبی رفتار ـ هستند. ویژگی‌های مهم و برجستهٔ مشخصات کنترل در بخش ۱۲-۴-۴ معرفی شدند. حالا وقت آن رسیده است که مثالی از این طرح مهم مدل‌سازی را برای تحلیل ساخت یافته مطرح کنیم.



شکل ۱۲ـ ۲۳سطح یک CFD (نمودار روند کنترل) خانه امن

نمودار ۱۲-۲۴ نمودار تغییر حالت را برای سطح ۱ مدل گردش کنترل در نرم‌افزار خانهٔ امن نشان می‌دهد. پیکان‌های نام‌گذاری شدهٔ تغییر حالت، نحوهٔ پاسخ‌گویی سیستم به رویدادها را در هنگامی نشان می‌دهد که سیستم هر چهار حالت تعریف شده در این سطح را پشت سر گذاشته است. مهندسین نرم‌افزار با مورد مطالعه قرار دادن STD می‌توانند رفتار سیستم را تعیین کنند و مهم‌تر از آن این‌که مهندسین با انجام این مطالعه می‌توانند تعیین کنند که آیا شکاف یا وقفه‌ای در رفتار مشخص شده وجود دارد یا خیر.



شکل ۱۲ـ۲۴ نمودار انتقال وضعیت برای خانه امن

حالت متفاوت نمایش رفتاری شامل جدول فعال‌سازی فرآیند می‌باشد. جدول فعال‌سازی فرآیند (PAT) اطلاعات موجود در STD را در بافت فرآیندها نشان می‌دهد و نه در بافت حالات مختلف یعنی این جدول نشان می‌دهد که کدام‌یک از فرآیندهای (حباب‌های) موجود در مدل جریان هنگام روی دادن یک رخداد فراخوانده می‌شوند. یک طراح که باید طرح اجرایی کنترل‌کنندهٔ فرآیندهای موجود در این سطح را ایجاد کند می‌تواند از این جدول PAT (جدول فعال‌سازی فرآیند) به‌عنوان راهنما استفاده کند. جدول فعال‌سازی فرآیند برای سطح ۱ مدل جریان نرم‌افزار خانهٔ امن ، در جدول ۱۲-۲۵ نشان داده شده است.

مشخصات CSPEG به توصیف رفتار سیستم می‌پردازد، اما این مشخصات هیچ‌گونه اطلاعاتی در مورد کار داخلی فرآیندهای فعال شده در نتیجهٔ این رفتار را در اختیار ما قرار نمی‌دهد. طرح مدل‌سازی که فراهم‌کنندهٔ این اطلاعات است در بخش بعدی مورد بررسی قرار خواهد گرفت.

۱۲-۶-۵ تعیین مشخصات فرآیند

از مشخصات فرآیند (PSPEC) برای توصیف تمام فرآیندهای مدل جریان که در سطح پایانی پالایش ظاهر می‌شوند، استفاده می‌گردد. محتویات مشخصات فرآیند عبارتند از: متن روایتی، زبان طراحی برنامه[1] (PDL)، توصیف و توضیح الگوریتم فرآیند، معادلات ریاضی، جداول، نمودارها یا چارت‌ها. مهندسین نرم‌افزار با فراهم کردن مشخصات کنترل به‌همراه هر یک از حباب‌های موجود در مدل جریان

1.Program Design Language

توانسته‌اند "حداقل مشخصات" را ایجاد کنند که این مشخصات نیز می‌توانند به‌عنوان اولین گام در ایجاد مشخصات نیازمندی‌های نرم‌افزاری[1] و به‌عنوان دستورالعملی برای طراحی جزء نرم‌افزاری پیاده‌سازی‌کنندهٔ فرآیند، عمل کنند.

برای واضح‌تر شدن موارد کاربرد مشخصات فرآیند (PSPEC)، فرآیند اسم رمز ذکر شده در مدل جریان نرم‌افزار خانهٔامن را درنظر بگیرید (نمودار ۱۲-۲۱) مشخصات فرآیندی این تابع می‌تواند به شکل زیر باشد:

PSPEC : پردازش رمز عبور

فرآیند رمز عبور تمام تصدیق‌ها و تأییدات مربوط به اسم رمز را برای سیستم خانهٔ امن اجرا می‌کند. فرآیند رمز عبور با برقراری ارتباط با تابع کاربر، یک اسم رمزی ۴ رقمی را دریافت می‌کند. این اسم رمز ابتدا با اسم رمز اصلی ذخیره شده در داخل سیستم مورد مقایسه قرار می‌گیرد. اگر این اسم رمز جدید با اسم رمز اصلی هماهنگی داشته باشد، (پیغام تصدیق = درست) به تابع نمایش پیغام و وضعیت انتقال داده می‌شود. اگر این اسم رمز با رمز اصلی هماهنگی نداشته باشد، این رمز ۴ رقمی با جدول اسم رمز ثانویه مورد مقایسه قرار خواهد گرفت. اگر این اسم رمز با یکی از ورودی‌های موجود در جدول اسم رمز ثانویه هماهنگی داشته باشد (پیغام تصدیق = درست) به تابع نمایش پیغام و وضعیت انتقال داده می‌شود. اما اگر هیچ‌گونه هماهنگی وجود نداشته باشد (پیغام تصدیق = غلط) به تابع نمایش پیغام و وضعیت انتقال داده می‌شود.

رویدادهای ورودی						
Sensor event	0	0	0	0	1	0
Blink flag	0	0	1	1	0	0
Start stop switch	0	1	0	0	0	0
Display action status						
Complete	0	0	0	1	0	0
In-progress	0	0	1	0	0	0
Time out	0	0	0	0	0	1
خروجی						
Alarm signal	0	0	0	0	1	0
فعال سازی فرایندها						
Monitor and control system	0	1	0	0	1	1
Active/deactivate system	0	1	0	0	0	0
Display messages and status	1	0	1	1	1	1
Interact with user	1	0	0	1	0	1

شکل ۱۲-۲۵ جدول فعال سازی فرآیند خانه امن

1.Software Requirements Specification

اگر جزییات الگوریتمی اضافی در این مرحله مورد نیاز باشد، نمایش زبان طراحی برنامه (PDL) نیز به‌عنوان بخشی از مشخصات فرآیند (PSPEG) درنظر گرفته خواهد شد. اما به اعتقاد برخی از افراد نسخهٔ PDL باید تا هنگام شروع طراحی اجزاء به تعویق بیافتد.

۱۲-۷ فرهنگ داده ها (واژه نامه داده ها)

مدل تحلیلی می‌تواند بازنمایی اشیاء داده‌ای، کاری و کنترلی را پوشش دهد. در هر یک از موارد، اشیاء داده‌ای و موارد کنترل نقشی را ایفا می‌کنند. بنابراین لازم است یک رهیافت سازمان یافته برای نشان دادن ویژگی‌های هر یک از اشیاء داده‌ای و موارد کنترل ارائه شود. این کار با استفاده از واژه‌نامهٔ داده‌ها تکمیل می‌شود.

فرهنگ یا واژه‌نامهٔ داده‌ها به‌عنوان دستور شبه‌رسمی برای توصیف محتویات اشیاء تعریف شده در طول تحلیل ساختیافته مطرح شده است. این علائم مهم در زمینهٔ مدل‌سازی، به‌شکل زیر تعریف شده است [YOU89][1].

واژه‌نامهٔ داده‌ها[2] لیستی سازمان یافته از تمام عناصر داده‌ای است که در ارتباط با سیستم می‌باشد. در این لیست تعاریف دقیق و مشخصی گنجانده شده است به‌گونه‌ای که کاربرد و تحلیل‌گر سیستم هر دو درک مشترکی از ورودی، خروجی، اجزاء ذخیره شده و حتی محاسبات میانی خواهند داشت.

امروزه واژه‌نامهٔ داده‌ها به‌عنوان بخشی از CASE " ابزار طراحی و تحلیلی ساخت یافته " پیاده‌سازی می‌شود. اگر چه شکل این واژه‌نامه‌ها از ابزاری به ابزار دیگر متفاوت است، اما بیشتر این واژه‌نامه‌ها دارای اطلاعات ذیل می‌باشند:

- **نام** ـ نام اولیهٔ داده‌ها و موارد کنترل و داده‌های ذخیره شده یک موجودیت و موارد (اشیاء) خارجی.
- **نام مستعار** ـ سایر نام‌هایی که برای ورودی اولیه مورد استفاده قرار گرفته‌اند.
- **محل استفاده و نحوهٔ استفاده** ـ لیستی از فرآیندهایی که از داده‌ها و موارد کنترل استفاده کرده‌اند و نحوهٔ استفاده از این داده‌ها و موارد کنترل (به‌عنوان مثال به‌صورت ورودی فرآیند، به‌صورت خروجی فرآیند، به‌صورت داده‌های ذخیره شده و به‌صورت موجودیت خارجی).
- **توضیح محتویات** ـ طرحی برای نشان دادن محتویات
- **اطلاعات تکمیلی** ـ سایر اطلاعات موجود در زمینهٔ نوع داده‌ها، مقادیر از پیش تنظیم شده، محدودیت‌ها یا موانع و مواردی از این قبیل.

1. Yourdon , E..N.
2. data dictionary

پس از وارد کردن نام اشیاء داده‌ای یا موارد کنترل و پس از وارد کردن نام مستعار آنها در واژه‌نامهٔ داده‌ها، می‌توان سازگاری نام‌گذاری این موارد را بررسی نمود. یعنی اگر اعضا تیم تحلیل تصمیم بگیرند که یک مورد اطلاعاتی جدید را xyz بنامند، اما xyz از قبل در واژه‌نامهٔ داده‌ها وجود داشته باشد. وسیله پشتیبانی‌کننده از این واژه‌نامه در CASE اخطاری را مبنی بر وجود نام تکراری نشان می‌دهد. و این حالت سازگاری مدل‌های تحلیلی را بهبود می‌بخشد و به کاهش خطا کمک می‌کند.

اطلاعات مربوط به "محل استفاده/ نحوهٔ استفاده" به‌صورت خودکار از روی مدل‌های جریان ضبط می‌شوند. هنگامی که ورودی واژه‌نامه ایجاد می‌گردد، ابزار CASE از تمام DFDها و CFDها (نمودار جریان داده‌ها و نمودار جریان کنترل) پویش می‌کند تا مشخص کند که کدام‌یک از فرآیندها از اطلاعات داده‌ای و اطلاعات کنترل استفاده کرده‌اند و نحوهٔ استفادهٔ آنها چگونه بوده است. اگر چه ممکن است این مسأله بی‌اهمیت به‌نظر برسد، اما در واقع این کار یکی از مهم‌ترین مزیت‌های این واژه‌نامه‌ها محسوب می‌شود. در طول تحلیل معمولاً جریان تغییرات به‌صورت مستمر و بدون وقفه دنبال می‌شود و در پروژه‌های بزرگ اغلب تعیین تأثیر تغییرات امری بسیار مشکل است. اکثر مهندسین نرم‌افزار این سؤالات را مطرح می‌سازند "محل استفادهٔ این اشیاء داده‌ای کجاست؟ و با تغییر این اشیاء داده‌ای چه چیزهای دیگری باید تغییر داده شوند؟ تأثیر کلی این تغییرات شامل چه چیزهایی می‌باشد؟" از آن‌جایی‌که واژه‌نامهٔ داده‌ها را می‌توان به‌عنوان یک بانک اطلاعاتی به‌کار برد. بنابراین تحلیل‌گران می‌توانند "سؤالات مربوط به محل و نحوه استفاده" را مطرح کنند و به پاسخ سؤالات مطرح شده در قسمت بالا برسند. طرح استفاده شده برای تکمیل توضیح محتویات در جدول ذیل گنجانده شده است :

Data Construct	Notation	Meaning
	=	is composed of
Sequence	+	and
Selection	[I]	either-or
Repetition	{}n	repetitions of
	()	optional data
	...	delimits comments

این طرح مهندسین نرم‌افزار را قادر ساخته است تا داده‌های مرکب را با استفاده از یکی از سه راه اساسی موجود برای ساخت این داده‌ها، ارائه دهند. این سه راه اساسی عبارتند از:

۱- بصورت توالی اقلام داده‌ای.

۲- بصورت انتخاب از میان مجموعه‌ای از اقلام داده‌ای.

۳- بصورت گروه تکراری اقلام داده‌ای. هر یک از ورودی‌های اقلام داده‌ای که به‌عنوان بخشی از توالی، انتخاب، تکرار مشخص می‌شود می‌تواند به تنهایی یک قلم داده‌ای مرکب دیگر باشد که این قلم داده‌ای نیز به تغییرات بیشتر در داخل این واژه‌نامه نیاز دارد.

برای نشان دادن موارد مصرف واژه‌نامهٔ داده‌ای به سطح دوم نمودار جریان داده‌ها در فرآیند سیستم نظارتی نرم‌افزار خانهٔ امن ، که در نمودار ۱۲-۲۲ نشان داده شده است، بازمی‌گردیم. با توجه به این نمودار می‌توان دریافت که در این نمودار قلم داده‌ای یعنی شمارهٔ تلفن به‌عنوان ورودی مشخص شده است. اما شماره تلفن دقیقاً چه چیزی می‌باشد؟ یک شماره تلفن می‌تواند یک شمارهٔ محلی ۷ رقمی، تلفن داخلی ۴ رقمی و یا یک شمارهٔ ۲۵ رقمی برای برقراری ارتباط در مسافت‌های طولانی باشد. واژه‌نامهٔ داده‌ها می‌تواند تعریف دقیقی از شمارهٔ تلفن مورد بحث در نمودار جریان داده‌ها، در اختیار ما قرار دهد. به‌علاوه این واژه‌نامه محل و نحوهٔ استفاده از این اقلام داده‌ای و هرگونه اطلاعات تکمیلی مربوط به این داده‌ها را نشان می‌دهد. ورودی واژه‌نامهٔ داده‌ها به‌صورت زیر آغاز می‌شود:

نام: شماره تلفن

نام مستعار: ---

محل استفاده/ نحوهٔ استفاده: ارزیابی تنظیم (خروجی) شماره‌گیری (ورودی)

توضیحات :

شمارهٔ تلفن = [شماره محلی | شماره راه‌دور]

شماره محلی = پیش شماره + شماره دسترسی

شماره راه‌دور = ۱ + شماره ناحیه + شماره محل

شماره ناحیه = [۸۰۰ | ۸۸۸ | ۵۶۱]

پیش شماره = *یک شماره سه رقمی که هرگز با ۰ یا ۱ آغاز نخواهد شد*

شماره دسترسی = *هر رشته چهار رقمی*

شماره تلفن همچون 01327 546381 را می‌توان با استفاده از واژه‌نامهٔ فوق‌الذکر توصیف نمود. در سیستم‌های بزرگ کامپیوتری، اندازه و پیچیدگی واژه‌نامهٔ داده‌ای به سرعت رشد می‌کند و توسعه می‌یابد. در حقیقت حفظ و گسترش یک واژه‌نامه به‌صورت دستی بسیار مشکل است. به‌همین دلیل باید برای گسترش دادن این واژه‌نامه از ابزار CASE استفاده نمود.

۱۲-۸ دیگر شیوه های تحلیل سنتی

در طول این سال‌ها بسیاری از دیگر شیوه های تحلیل نیازمندی‌های نرم‌افزاری در تمام صنایع مورد استفاده قرار گرفته است. در حالی‌که تمام این شیوه ها از اصول تحلیل عملیاتی توضیح داده شده در فصل ۱۱ تبعیت می‌کنند، اما هر یک از آنها از یک طرح متفاوت و مجموعهٔ منحصربه‌فردی از اعمال ذهنی برای ایجاد مدل تحلیلی استفاده می‌کنند.

بررسی اجمالی سه شیوه تحلیلی مهم عبارتند از:

- توسعه سیستم‌های ساختیافته داده‌ای[1] (DSSD) [WAR81][2] و [ORR81][3]
- توسعه سیستم جکسون[4] (JSD)[JAC83] [5]
- تکنیک طراحی و تحلیل ساختیافته [ROS77][6] و [ROS85][7]

که این ۳، در سایت وب SEPA برای مطالعهٔ بیشتر افراد علاقه‌مند به اطلاعات بیشتر در زمینهٔ مدل‌سازی تحلیل ارائه شده است.

۱۲-۹ خلاصه

تحلیل ساختیافته، که به‌عنوان یکی از شیوه های مدل‌سازی نیازمندی‌ها به‌طور گسترده مورد استفاده قرار می‌گیرد، بر مدل‌سازی داده‌ها و مدل‌سازی جریان وابسته است تا بتواند مبنایی را برای مدل جامع تحلیلی ایجاد کند. مهندسین نرم‌افزار با استفاده از نمودارهای موجودیت/رابطه شمایی از تمام اشیاء داده‌ای مهم برای سیستم، ایجاد کرده‌اند. نمودارهای جریان کنترل داده‌ای به‌عنوان مبنایی برای نشان دادن تبدیل و تغییر داده‌ها و کنترل مورد استفاده قرار می‌گیرند. هم‌چنین این مدل‌ها برای ایجاد مدل کاربردی نرم‌افزار و فراهم کردن مکانیسمی برای انجام کار تقسیم‌بندی و تفکیک‌بندی مورد استفاده قرار می‌گیرند. می‌توان مدل رفتاری را با استفاده از نمودار تغییر حالت ایجاد نمود. محتویات داده‌ای نیز به کمک واژه‌نامهٔ داده‌ای تکمیل می‌گردد. مشخصات کنترل و فرآیند نیز می‌توانند جزییات بیشتری را مشخص کنند.

طرح اصلی و اولیهٔ تحلیل ساختیافته برای پردازش داده‌های متعارف تکمیل شده است. اما در حال حاضر گسترش و تکمیل این موجب شده است که امکان استفاده از این طرح در سیستم‌های بلادرنگ

1.Data Structured Systems Development

2.Warnier , J.D.

3.Orr,K.T.

4.Jackson System Development

5.Jackson , M.A.

6.Ross, D. and K.

7.Ross, D.

نیز فراهم شود. تحلیل ساختیافته به‌وسیلهٔ آرایه‌ای از ابزارهای CASE که به ایجاد هر یک از عناصر این مدل و به تضمین سازگاری و صحت این طرح کمک می‌کنند، حمایت می‌شود.

مسایل و نکاتی برای تفکر و تعمق بیشتر

۱۲-۱ حداقل سه مرجع مورد بحث در بخش ۱۲ـ۱ را مطالعه کرده، مقالهٔ کوتاهی بنویسید که در آن چگونگی تغییر نگرش به تحلیل ساختیافته با گذشت زمان مطرح شده باشد. به عنوان نتیجه گیری، راههایی را پیشنهاد کنید که گمان دارید در آینده شیوه را دستخوش تغییر می سازد.

۱۲-۲ از شما درخواست شده که یکی از سیستمهای زیر را بسازید:

الف. سیستم ثبتنام رشته مبتنی بر شبکه برای دانشگاه خود

ب. سیستم پردازش سفارشات مبتنی بر وب برای مغازه کامپیوتر

پ. سیستم ساده صدور فاکتور برای یک تجارتخانه کوچک

ت. نرم‌افزاری که جایگزین یک رولودکس در تلفن بی‌سیم خواهد شد.

ث. یک کتاب آشپزی خودکار که در داخل یک اجاق گاز الکتریکی یا میکروویو قرار می گیرد.

سیستم مورد علاقه خود را انتخاب کرده یک نمودار موجودیت رابطه بسازیدکه اشیای داده‌ای، روابط میان آنها و صفات خاصه آنها را توصیف کند.

۱۲-۳ چه اختلافی میان کاردینالیته و مدالیته وجود دارد؟

۱۲-۴ یک مدل در سطح بافت (DFD سطح صفر) برای یکی از پنج سیستم لیست شده در مسئله ۱۲ـ۲ رسم کنید. یک حکایت پردازش در سطح بافت برای سیستم انتخاب کنید.

۱۲-۵ با استفاده از DFD در سطح (متن) بافت که در مسئله ۱۲ـ۴ تهیه کرده‌اید، DFDهای سطح ۱ و سطح ۲ را توسعه دهید. برای شروع کار از "تجزیهٔ گرامری" حکایت پردازش در سطح بافت استفاده کنید. به‌خاطر داشته باشید که تمام جریان اطلاعاتی را با نشانه‌گذاری روی پیکان‌های میان حبابها مشخص کنید. برای هر یک از تبدیلات نامی معنا دار برگزینید.

۱۲-۶ برای سیستمی که در مسئله ۱۲ـ۲ انتخاب کرده‌اید، یک CFD، CSPEC، PSPEC و فرهنگ داده‌ها را بسازید. بکوشید مدل تا آنجا که امکان دارد کامل شود.

۱۲-۷ آیا مفهوم تسلسل جریان اطلاعات به این معناست که اگر یک پیکان جریان در سطح صفر به‌عنوان ورودی ظاهر شد، در آن‌صورت یک پیکان جریان باید به‌عنوان ورودی سطح بعدی ظاهر شود؟ پاسخ خود را تشریح کنید.

۱۲-۸ با استفاده از توسعه وارد و ملور، مدل جریان مربوط به شکل ۱۲ـ۱۶ را دوباره رسم کنید. شما چگونه مشخصه کنترلی را که در شکل ۱۲ـ۱۶ وجود دارد، تطبیق می دهید؟ وارد و ملور از این علائم استفاده نمی‌کنند.

۱۲-۹ با استفاده از توسعه هتلی و پیربهای، مدل مربوط به شکل ۱۲ـ۱۳ را دوباره رسم کنید. چگونه فرآیند کنترلی (دایره خط چین) را در شکل ۱۲ـ۱۳ وجود دارند، تطبیق می دهید؟ هتلی و پیربهای از این علائم استفاده نمی‌کنند.

۱۲-۱۰ به زبان خودتان جریان رویداد را شرح دهید.

۱۲-۱۱ یک مدل جریان کامل برای نرم‌افزار فتوکپی توضیح داده شده در بخش ۵-۱۲ توسعه دهید. شما می توانید از روش وارد و ملور یا هتلی و پیربهای استفاده کنید. اطمینان حاصل کنید که یک نمودار انتقال وضعیت تفصیلی برای سیستم توسعه داده اید.

۱۲-۱۲ حکایات پردازشی را برای مدل تحلیل نرم‌افزار خانه امن نشان داده شده در شکل ۱۲ـ۲۱ کامل کنید. مکانیک محاوره میان کاربر و سیستم را شرح دهید. آیا اطلاعات اضافی شما، مدل‌های جریان را که برای خانه امن ارائه شده است، تغییر می‌دهد؟ اگر اینطور است، چگونه؟

۱۲ـ۱۳ دپارتمان امور عامه در یک شهر بزرگ تصمیم دارد که یک سیستم PHTRS (سیستم تشخیص دست اندلز و رفع آن) که مبتنی بر وب باشد، تهیه نماید. توصیف کار به صورت زیر میباشد:

شهروندان می توانند از طریق اینترنت به سایت وب وصل شوند و مختصات چاله ها را اعلام نمایند. این مشخصات و مختصات به " سیستم تعمیر دپارتمان کارهای عامه" منتقل می شود. و یک شماره شناسایی به آن تعلق می گیرد و به همراه آدرس خیابان، اندازه چاله (مقیاس از ۱ تا ۱۰) ، محل (وسط خیابان، کنارآن و غیره)

با استفاده از علائم تحلیل ساختیافته، یک مدل تحلیلی کامل برای PHTRS توسعه دهید.

۱۲ـ۱۴ نرم افزارهای نسل بعد، برای سیستم پردازش کلمات در دست توسعه اند. ساعتی چند در خصوص حیطه کاربردی آن تحقیق کنید و یک نشست FAST (فصل ۱۱) با دانشجویانتان داشته باشید.(استادتان شما را در این هماهنگی یاری خواهد نمود). یک مدل نیازمندیهای سیستم را با استفاده از تحلیل ساختیافته بسازید.

۱۲ـ۱۵ نرم افزاری برای یک بازی ویدئویی در دست ساخت می باشد. مطابقه مسئله ۱۲ـ۱۴ عمل نمایید.

۱۲ـ۱۶ با چهار یا پنج فرشنده ابزارهای CASE برای تحلیل ساختیافته تماس حاصل نمایید. متن آنها را مورد مرور قرار داده و یک مقاله کوتاه در خصوص جمع بندی ویژگیهای عمومی ای که یک ابزار را از دیگری متمایز می کند، بنویسید.

فهرست منابع و مراجع

[BRU88] Bruyn, W. et al., "ESML: An Extended Systems Modeling Language Based on the Data Flow Diagram," *ACM Software Engineering Notes*, vol. 13, no. 1, January 1988, pp. 58-67.

[CHE77] Chen, P., *The Entity-Relationship Approach to Logical Database Design*, QED Information Systems, 1977.

[DEM79] DeMarco, T., *Structured Analysis and System Specification*, Prentice-Hall, 1979.

[GAN82] Gane, T. and C. Sarson, *Structured Systems Analysis*, McDonnell Douglas, 1982.

[HAT87] Hatley, D.J. and I.A. Pirbhai, *Strategies for Real-Time System Specification*, Dorset House, '87.

[JAC83] Jackson, M.A., *System Development*, Prentice-Hall, 1983.

[ORR81] Orr, K.T., *Structured Requirements Definition*, Ken Orr & Associates, Inc., 1981.

[PAG80] Page-Jones, M., *The Practical Guide to Structured Systems Design*, Yourdon Press, 1980.

[ROS77] Ross, D. and K. Schoman, "Structured Analysis for Requirements Definition," *IEEE Trans. Software Engineering*, vol. SE-3, no. 1, January 1977, pp. 6-15.

[ROS85] Ross, D. "Applications and Extensions of SADT," *IEEE Computer*, vol. 18, no. 4, April 1984, pp. 25-35.

[STE74] Stevens, W.P., G.J. Myers, and L.L. Constantine, "Structured Design," *IBM Systems journal*, vol. 13, no. 2,1974, pp. 115-139.

[TIL93] Tillmann, G., *A Practical Guide to Logical Data Modeling*, McGraw-Hill, 1993.

[WAR81] Warnier, J.D., *Logical Construction of Systems*, Van Nostrand-Reinhold, 1981.

[WAR85] Ward, P.T. and S.J. Mellor, *Structured Development for Real-Time Systems* (three volumes), Yourdon Press, 1985.

[YOU78] Yourdon, E.N. and Constantine, L.L., *Structured Design*, Yourdon Press, 1978.

[YOU89] Yourdon, E.N., *Modern Structured Analysis*, Prentice-Hall, 1990.

خواندنیهای دیگر و منابع اطلاعاتی

Dozens of books have been published on structured analysis. All cover the subject adequately, but only a few do a truly excellent job. DeMarco's book [DEM79] remains a good introduction to the basic notation. Books by Hoffer et al. (*Modern Systems Analysis and Design*, Addison-Wesley, 2nd ed., 1998). Kendall and Kendall (*Systems Analysis and Design*, 2nd ed., Prentice-Hall, 1998), Davis and Yen (*The Information System Consultant's Handbook: Systems Analysis and Design*, CRC Press, 1998), Modell (*A Professional's Guide to Systems Analysis*, 2nd ed., McGraw-Hill, 1996), Robertson and Robertson (*Complete Systems Analysis*, 2 volumes, Dorset House, 1994), and Page-Jones (*The Practical Guide to Structured Systems Design*, 2nd ed., Prentice-Hall, 1988) are worthwhile references. Yourdon's book on the subject [YOU89] remains among

the most comprehensive coverage published to date.
For an engineering emphasis [WAR85] and [HAT87] are the books of preference. However, Edwards *(Real-Time Structured Methods: Systems Analysis*, Wiley, 1993) also covers the analysis of real-time systems in considerable detail, presenting a number of useful examples drawn from actual applications.
Many variations on structured analysis have evolved over the last decade. Cutts *(Structured Systems Analysis and Design Methodology*, Van Nostrand-Reinhold, 1990) and Hares *(SSADM for the Advanced Practitioner*, Wiley, 1990) describe SSADM, a variation on structured analysis that is widely used in the united Kingdom and Europe.
Flynn et al *(Information Modeling: An International Perspective*, Prentice-Hall, 1996), Reingruber and Gregory *(Data Modeling Handbook*, Wiley, 1995) and Tillman [TIL93] present detailed tutorials for creating industry-quality data models. Kim and Salvatore ("Comparing Data Modeling Formalisms," *Communications of the ACM*, June 1995) have written an excellent comparison of data modeling methods. An interesting book by Hay *(Data Modeling Patterns*, Dorset House, 1995) presents typical data model "patterns" that are encountered in many different businesses. A detailed treatment of behavioral modeling can be found in Kowal *(Behavior Models: Specifying Users Expectations*, Prentice-Hall, 1992).
A wide variety of information sources on structured analysis and related subjects is available on the Internet. An up-to-date list of World Wide Web references that are relevant to analysis concepts and methods can be found at the SEPA Web site:
http://www.mhhe.com/engcs/compsci/pressman/resources/reqm-analysis.mhtml