فهرست مطالب

پیشگفتار

فصل اول - مقدمه‌ای بر SSADM

تاریخچه SSADM ۱۲

جایگاه SSADM ۱۲

SSADM چیست ۱۵

مزایای SSADM ۱۵

فصل دوم - ساختار کلی SSADM

ساختار SSADM ۱۸

ماجول صفر (امکان سنجی) ۲۰

ماجول تجزیه و تحلیل نیازها ۲۱

ماجول مشخص‌سازی نیازها (مرحله مشخص‌سازی نیازها) ۲۵

ماجول طراحی منطقی ۲۷

ماجول طراحی فیزیکی (مرحله طراحی فیزیکی) ۲۹

فصل سوم - امکان سنجی

گام ۱۰ آمادگی جهت امکان‌سنجی ۳۲

گام ۲۰ تعریف مسئله ۳۳

گام ۳۰ تعیین راه‌حل‌های موجود ۳۴

فصل چهارم - مدل فعالیت تجاری

بررسی مقدماتی ۳۶

مدل فعالیت تجاری چیست ۳۶

ایجاد مدل فعالیت تجاری ۳۸

روش تجزیه و تحلیل عملکردها چیست ۳۹

نمودار سطح صفر ۴۳

فصل پنجم - تعیین مشخصات نیازمندیها

چگونگی تشخیص نیازمندیها ۴۶
ایجاد لیست کاربران ۴۹

فصل ششم - بررسی وضع موجود

گام ۱۳۰ بررسی وضعیت موجود ۵۲
گام ۱۴۰ بررسی و ایجاد ساختار منطقی داده‌ها ۵۴
رابطه بین موجودیتها ۵۵
زیرموجودیت ۵۸
مدل منطقی داده‌ها ۵۹
ترسیم نمودارهای سیستم رزواسیون متل ۵۹

فصل هفتم - ارائه دیدگاه منطقی از وضعیت موجود

مراحل مختلف منطقی سازی ۶۴
بررسی مجدد مدل فعالیت تجاری ۶۵
محل ذخیره داده ۶۷
ترسیم نمودار منطقی جریان داده‌ها ۶۷
قواعد ترسیم نمودار منطقی جریان داده‌ها ۷۲
ارزیابی مدل منطقی داده‌ها ۷۶

فصل هشتم - گزینه‌های موجود و راه‌حل انتخابی

گام ۲۱۰ گزینه‌های موجود ۷۸
گام ۲۲۰ تعیین گزینه ۷۹

فصل نهم - تعیین پردازشها و ایجاد ساختار منطقی داده‌ها

اهداف مرحله سوم ۸۲
گامهای ۳۱۰ و ۳۲۰ ..... ۸۳
سیستم رزواسیون متل ۸۴

فصل دهم - کارکردها

کارکرد چیست ۹۰
دسته‌بندی کارکردها ۹۳
مستندسازی کارکردها ۹۸
طراحی صفحات ورودی و خروجی ۱۰۰
حقوق کاربران ۱۰۳

فصل یازدهم- تجزیه و تحلیل منطقی اطلاعات

مقدمه‌ای بر منطقی سازی ۱۰۶
جداول ۱۰۷
کلیدها ۱۰۸
فرم نرمال از نوع سوم ۱۱۰
نمودار وابستگی ۱۱۰
ترسیم نمودار وابستگی ۱۱۱
سایر ملاحظات ۱۱۶
منطقی‌سازی ۱۱۷
ایجاد ساختار منطقی داده‌ها ۱۱۸
مقایسه دو ساختار ۱۲۱

فصل دوازدهم -نمونه سازی

انواع نمونه سازی ۱۲۶
جایگاه نمونه‌سازی در SSADM ۱۲۷
تعیین محدوده نمونه سازی ۱۲۸
مزایا و معایب ۱۲۸
ایجاد مستندات ۱۲۸

فصل سیزدهم - تاریخچه زندگی موجودیت

از کجا آغاز کنیم ۱۳۳
شناسایی رویدادها ۱۳۵
نمودار تاریخچه زندگی موجودیت ۱۴۰

تاثیرات رویداد بر نمونه ۱۴۲
خروج‌ها ۱۴۵
نقش موجودیت ۱۴۸
زیر زندگی ۱۵۲
اقلام وابسته، رویداد حذف و زیر موجودیت‌ها ۱۵۶
ایجاد فهرست رویدادها ۱۵۸

فصل چهاردهم - تعیین مسیر دسترسی
نمودار مسیر دسترسی چیست ۱۶۳
چگونگی رسم نمودار مسیر دسترسی ۱۶۵
نمودار تطابق اثر ۱۷۳

فصل پانزدهم - طراحی واسط گرافیکی
مدل‌سازی فعالیت ۱۸۳
مدل اشیا کاربر ۱۸۴
مدل پیمایش کارکردی ۱۸۵
مدل پیمایش پنجره‌ها ۱۸۶

فصل شانزدهم - آخرین گام در مرحله سوم
لیست کارکردها ۱۹۰
گام آخر ۱۹۱
جمع‌آوری مشخصات نیازمندیها ۱۹۲

فصل هفدهم - تعیین گزینه‌های فنی و انتخاب گزینه مناسب
گام ۴۱۰ تعیین گزینه‌های فنی سیستم ۱۹۷
گام ۴۲۰ انتخاب گزینه فنی ۱۹۹

فصل هیجدهم - طراحی منطقی
گام ۵۱۰ تعیین مشخصات پردازش‌های بروزرسانی ۲۰۳
گام ۵۲۰ تعریف پردازش‌های پرس و جو ۲۰۹

فصل نوزدهم - طراحی فیزیکی

گام ۶۱۰ ایجاد آمادگی جهت طراحی بانک اطلاعاتی ۲۲۱
گام ۶۲۰ طراحی اولیه بانک اطلاعاتی..... ۲۲۲
گام ۶۳۰ طراحی گفتگوهای سیستم..... ۲۲۶
گام ۶۴۰ پیاده‌سازی کارکردها ۲۳۲
گام ۶۵۰ تهیه مشخصات برنامه‌ها..... ۲۳۴
گام ۶۶۰ بهینه‌سازی بانک اطلاعاتی ۲۳۶
گام ۶۷۰ ایجاد واسط بین داده‌ها و پردازش‌ها ۲۳۷
گام آخر گردآوری نتایج ۲۳۸

فصل اول

مقدمه‌ای بر
SSADM

تاریخچه SSADM

روش تجزیه و تحلیل ساخت یافته (SSADM) ابتدا در سال ۱۹۸۱ توسط آژانس مرکزی حسابگری و ارتباطات انگلستان، به منظور استاندارد کردن روشهای توسعه و ایجاد سیستم‌ها در مراکز دولتی عرضه شد و قبل از آن روش مشخصی برای توسعه و عرضه سیستم‌های کاربردی وجود نداشت.

بدون وجود روشهای معین برای اندازه‌گیری و تخصیص منابع، تعریف دقیق و مشخص سیستم و ایجاد آن کاری بسیار مشکل و گاه ناشدنی می‌گردد و در نتیجه بسیاری از سیستم‌های کاربردی خیلی دیرتر از موعد مقرر عرضه گردیده و جوابگوی خواسته‌های کاربران نمی‌باشند. اینگونه سیستم‌ها حتی پس از تحویل نیز دارای مستندات لازم جهت آشنایی کاربران با وظایف خودشان نبوده و اطلاعات لازم جهت نگهداری سیستم نیز در اختیار آنان نمی‌باشد. بهمین دلیل پس از مدتی رها می‌شوند و جای خود را با سیستم دیگری عوض می‌کنند که در اغلب اوقات سیستم جدید نیز به راه سیستم قدیم می‌رود.

به هر حال از آن زمان تا کنون متدولوژی SSADM تغییرات بسیاری کرده است. از دلایل عمده این تغییرات یکی ضعف متدولوژی و دیگر تغییر تکنولوژی بوده است که باعث شد نسخه سوم آن در سال ۱۹۸۶ تکنیکهای جدیدی را در جهت ارتباط کاربران با سیستم مطرح نماید. سپس در سال ۱۹۹۰ نسخه 4.0 و نسخه 4.2 در سال ۱۹۹۵ عرضه شد و برای اولین بار تکنیک نمونه‌سازی (Prototyping) را مطرح کرد و آخرین نسخه آن نسخه 4.3که در سال ۱۹۹۶ عرضه گردیده است.

جایگاه SSADM

SSADM جزئی از چرخهٔ توسعهٔ سیستم می‌باشد همانطور که می‌دانید چرخه توسعهٔ سیستم از مراحلی تشکیل می‌شود که برای توسعهٔ سیستم ضروری می‌باشد. این مراحل عبارتند از:

۱. تعریف مسئله:

در این مرحله نیازهای کاربران مشخص می‌شود و وظایف سیستم آتی نیز تعیین می‌گردد.

۲. امکان سنجی:

در این مرحله مشخص می‌شود که آیا اصولاً می‌توان به نیازهای کاربران پاسخ گفت و اگر جواب مثبت است چه راه‌حلهایی وجود دارد.

۳. تجزیه و تحلیل سیستم:

شامل بررسی و تجزیه و تحلیل سیستم موجود و بررسی نیازهای کاربران می‌باشد.

۴. طراحی سیستم پیشنهادی:

این مرحله شامل ایجاد و طراحی سیستم جدید برای رفع نیاز کاربران می‌باشد (در SSADM نام این مرحله مشخص نمودن نیازها است).

۵. طراحی منطقی:

این مرحله شامل حذف دوباره کاریها یا چندباره کاریها و رفع ایرادات منطقی سیستم و تعیین جزئیات و وظایف هر کارکرد (Function) می‌باشد.

۶. طراحی فیزیکی:

شامل طراحی بانک اطلاعاتی و تهیه مشخصات و برنامه‌ها می‌باشد.

۷. توسعه:

در این مرحله برنامه‌های مربوطه با توجه به استانداردها و شرح آنها نوشته می‌شود.

۸. پیاده‌سازی:

این مرحله شامل پیاده‌سازی، راه‌اندازی و عملیاتی کردن سیستم جدید می‌باشد.

۹. تست و آزمایش:

در این مرحله تست‌ها و آزمون‌های مختلف از برنامه‌ها و ارتباطات آنها توسط طراحان و کاربران ،جهت اطمینان از عملکرد درست سیستم انجام می‌شود.

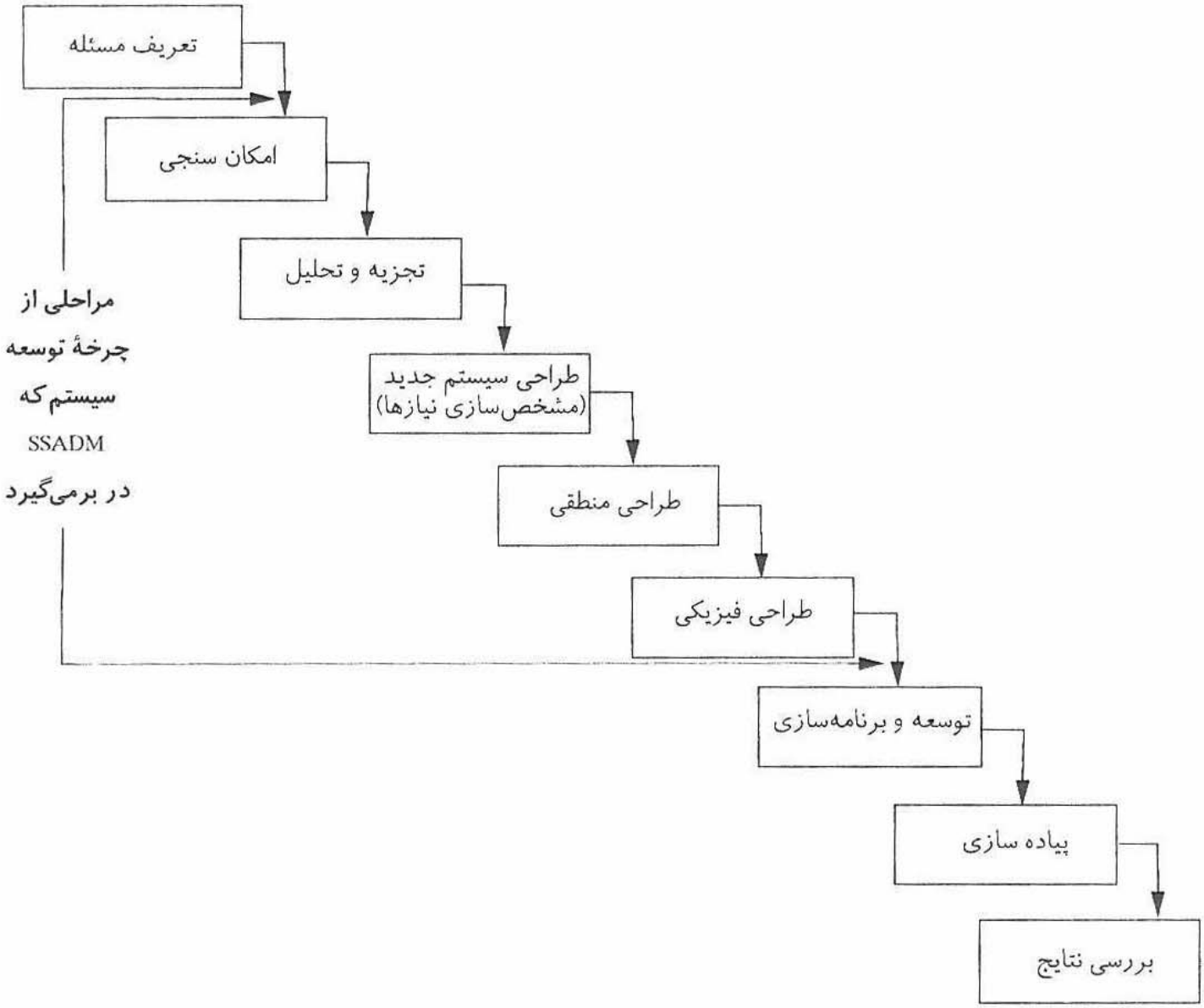
۱۰. بررسی سیستم:

این بررسی نشان می‌دهد که سیستم تا چه اندازه در رسیدن به اهداف خود موفق بوده است.

۱۱. نگهداری:

که شامل رفع ایرادات جزئی و اشتباهات می‌باشد.

شکل ۱.۱ مراحل مختلف چرخه توسعهٔ سیستم و قسمت‌هایی را که SSADM در بر می‌گیرد نشان می‌دهد.



شکل ۱.۱ مراحل چرخهٔ توسعه سیستم

SSADM چیست

SSADM مجموعه‌ای بهم پیوسته از استانداردها و خطوط راهنما برای تجزیه و تحلیل و طراحی می‌باشد. SSADM مراحل ساخت سیستم را در قالب کارها و وظایف مشخصی که به روشنی تعریف شده‌اند و رابطه معینی با یکدیگر دارند ارائه می‌دهد و گروه توسعه سیستم را راهنمائی می‌کند تا بدانند کی و چگونه از چه ابزارها و روش‌هایی استفاده کنند. SSADM روشهائی برای ثبت و مستندسازی فعالیت‌ها ارائه داده و مشخص می‌نماید که در هر مرحله از توسعه و ایجاد سیستم چه مستنداتی تولید شود.

مزایای SSADM

مزایای استفاده از SSADM عبارتند از:

- کنترل بهتر پروژه
- تحویل به موقع سیستم
- تحویل سیستم مطابق نیازهای کاربران و منطبق با اهداف سازمان
- استفاده بهتر از امکانات و مهارت‌ها
- انعطاف‌پذیری بیشتر
- افزایش بهره‌وری
- عدم وابستگی به نرم‌افزار یا سخت افزارهای خاص
- امکان استفاده از ابزارهای نرم‌افزاری (Cose tools) جهت توسعهٔ سیستم

فصل دوم

ساختار کلّی

SSADM

اهداف فصل

در این فصل بخشهای مختلف SSADM بصورت کلی بیان می‌شود همانطور که در فصل اوّل گفتیم، آخرین نسخه SSADM نسخه 4.3 می‌باشد که سعی شده است ساختاری که توضیح داده می‌شود منطبق بر آن باشد.

آنچه در پی می‌آید ساختار کلی SSADM است که هر مدیر پروژه می‌تواند بنا به ضرورت قسمتهایی از آن را حذف و یا تغییر دهد و روشهایی را انتخاب کند که مناسب امکانات او باشد.

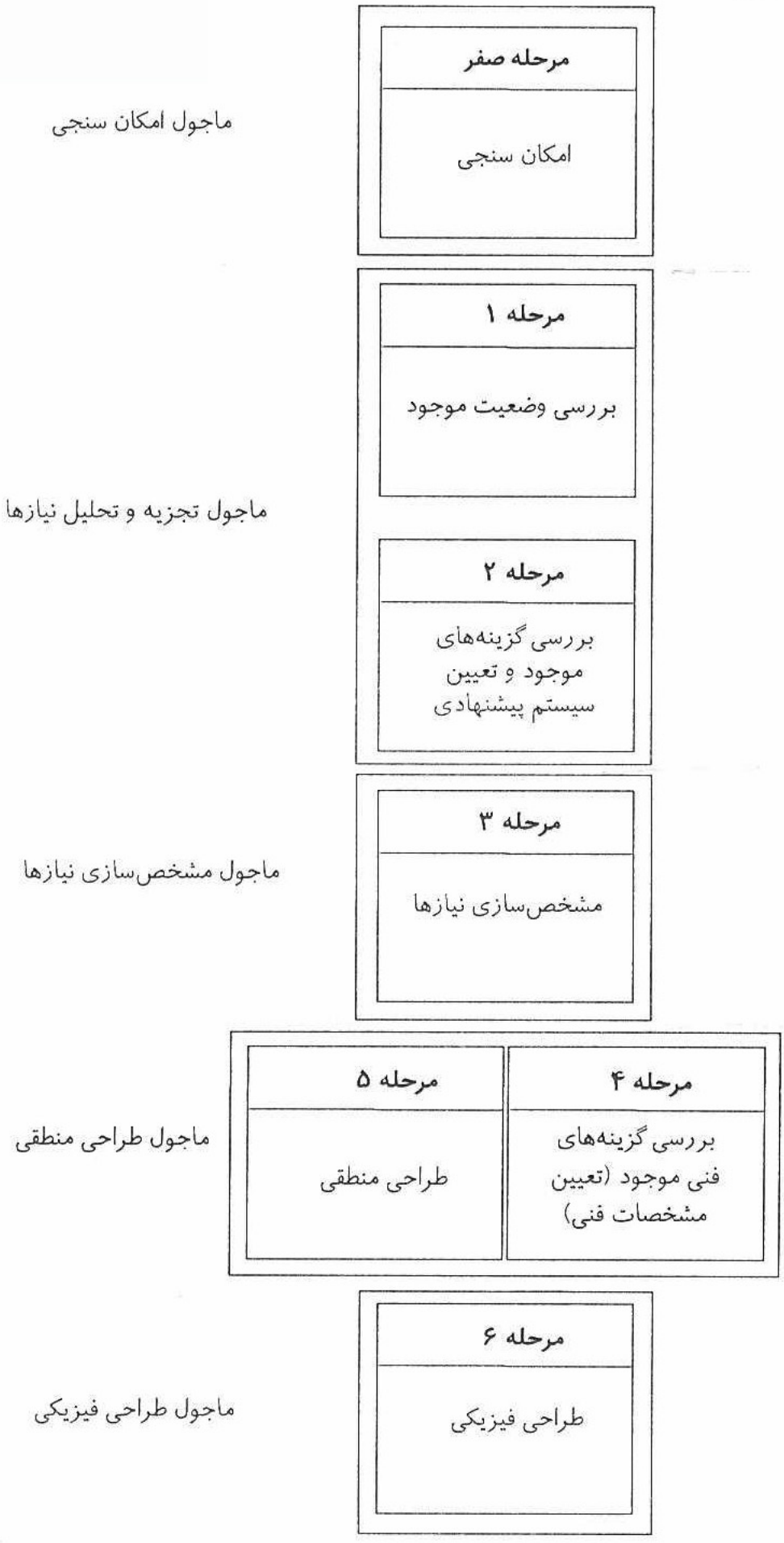
ساختار SSADM

در SSADM پنج ماجول (MODULE) وجود دارد و هر ماجول از چند مرحله (Stage) و هر مرحله از چندین گام (Step) و هر گام از چندین کار (Task) تشکیل شده است. در این بخش هدف آنست که این ساختار بصورت کلی تا سطح گام بیان گردد در فصول بعدی هر یک از اجزا همراه با مثالهای مناسب تشریح می‌گردد.

ماجولهای SSADM عبارتند از:

- امکان سنجی
- تجزیه و تحلیل نیازها
- تعیین و مشخص نمودن نیازها
- طراحی منطقی
- طراحی فیزیکی

شکل ۲.۱ ماجولهای مختلف را به همراه مراحل هر ماجول نشان می‌دهند.



شکل ۲.۱ ماجول‌های SSADM

ماجول صفر (امکان سنجی)

این ماجول شامل یک مرحله امکان سنجی می‌باشد و هدف از آن پاسخ به این سؤال است که آیا وضعیت موجود می‌تواند در قالب یک سیستم کامپیوتری پیاده‌سازی گردد و اگر سیستم موجود کامپیوتری است آیا سیستم جدید می‌تواند به اهداف مورد نظر دست یابد یا خیر. سیستم جدید با چه امکانات و محدودیتهایی می‌تواند پیاده‌سازی شود و در نهایت سازمان چه هزینه‌ای را بایستی متحمل شود. این مرحله اختیاری است و در پروژه‌هایی که از ابعاد کوچک یا متوسط برخور دارند و یا از لحاظ زمانی محدودیت دارند، می‌تواند حذف گردد و این ماجول شامل گامهای زیر است.

ـ گام ۱۰ آمادگی برای شروع امکان سنجی

در این گام مستندات آغاز پروژه بررسی و براساس آن منابع و مراجع و محدودهٔ پروژه و محدودیتهای موجود مشخص می‌شود.

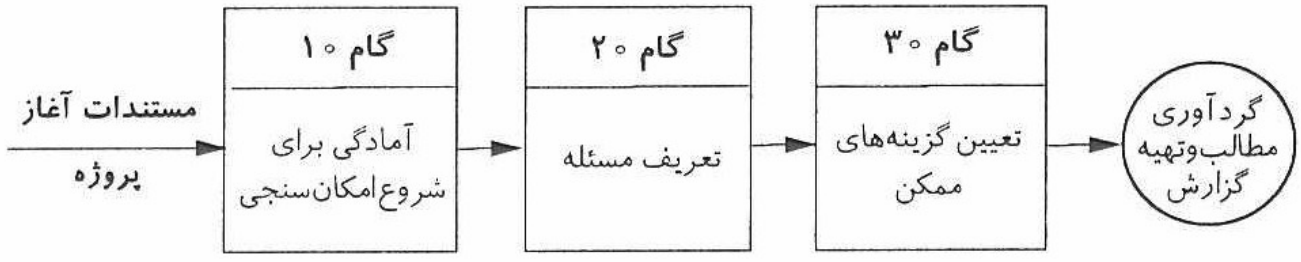
ـ گام ۲۰ تعریف مسئله

در این گام سیستم موجود با استفاده از ابزارهایی نظیر نمودار جریان داده‌ها (DFD) و نمودار ساختار منطقی داده‌ها LDS شناسایی و تعریف می‌شود. در این گام نیازها شناسائی شده و کاربران سیستم مشخص می‌گردند.

ـ گام ۳۰ تعیین گزینه‌های ممکن (سیستم‌های پیشنهادی)

در این گام راه‌حلهای مختلف ارائه شده و روشهای متفاوتی که سیستم جدید براساس آن می‌تواند شکل بگیرد بصورت یک گزارش در اختیار کمیته بررسی کننده قرار می‌گیرد.

در نسخه‌های قبلی SSADM (قبل از نسخه ۴) گام شماره ۴۰ نیز در نظر گرفته شده بود که شامل گردآوری مطالب و تهیه گزارش امکان‌سنجی می‌شد ولی در نسخه ۴ این گام با گام شماره ۳۰ ادغام گردیده است.



شکل ۲.۲ نمودار گامهای مرحله امکان سنجی

ماجول تجزیه و تحلیل نیازها

این مافوق می‌تواند مرحله آغاز پروژه باشد. در صورتی که گزارش امکان سنجی تهیه نشده باشد یا پس از مرحله امکان سنجی شروع شود، ماجول شامل دو مرحله (Stage) زیر است:

ـ مرحله اول بررسی وضعیت موجود

این فعالیت برای بررسی و شناخت وضعیت موجود صورت می‌گیرد و براساس اطلاعات جمع‌آوری شده مدل منطقی داده‌ها (Logical Data Model) و پردازش‌های موجود سیستم مشخص می‌شود. این مرحله شامل گامهای زیر است:

ـ گام ۱۱۰ تعیین محدودهٔ مورد بررسی

در این گام محدودهٔ سیستم مورد شناسایی قرار می‌گیرد.

ـ گام ۱۱۵ ایجاد مدل تجاری سیستم

در این گام قواعد وضعیت موجود بررسی و تعیین می‌گردد. بطور خلاصه در این گام به پرسشهایی از این قبیل، پاسخ داده می‌شود:

- علت وجود سیستم چیست؟
- عملکرد سیستم چیست؟
- عملیات به چه صورت و در چه زمان صورت می‌گیرد؟

ـ گام ۱۲۰ تعیین مشخصات نیازمندیها

در این گام مشکلات وضعیت موجود بررسی شده و نیازهای جدید مشخص می‌شود و لیست نیازها (Requirement Catalogue) با در نظر گرفتن جزئیات آماده می‌شود. بطور خلاصه نیازهایی که سیستم آتی بایستی پاسخگوی آنها باشد و همچنین کاربران احتمالی سیستم جدید شناسایی می‌شوند.

ـ گام ۱۳۰ شناسایی و بررسی پردازشهای سیستم

در این گام پردازش‌های اصلی وضعیت موجود در بررسی قرار گرفته و نقاط ضعف این پردازش‌ها مشخص می‌شود و در لیست نیازها قرار می‌گیرد.

ـ گام ۱۴۰ بررسی و ایجاد مدل منطقی داده‌ها

در این گام ساختار منطقی داده‌ها برای وضعیت موجود تهیه می‌گردد و سعی می‌شود که موجودیتها (Entities) و اقلام اطلاعاتی آنها در وضعیت موجود

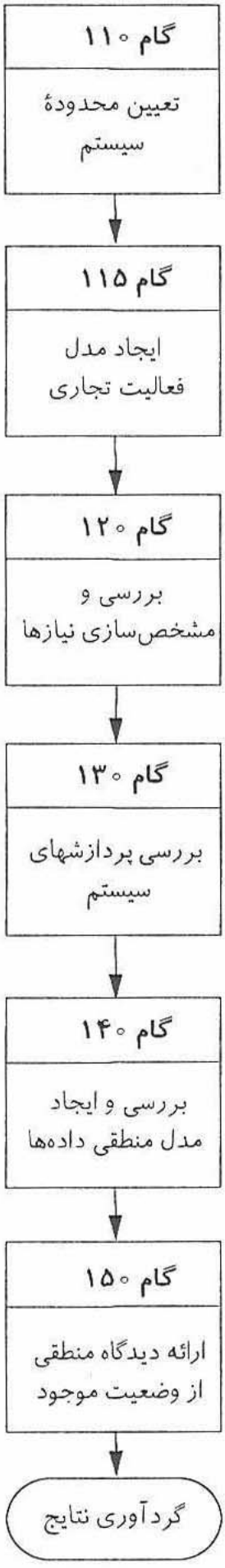
شناسایی گردد.

ـ گام ۱۵۰ ارائه دیدگاه منطقی از وضعیت موجود

در این گام با توجه به مدل منطقی داده‌ها و نمودار جریان داده‌ها در وضع موجود دوباره کاریها و اشکالات موجود در روش‌ها و رویه‌ها و نقاط کور آنها مشخص می‌شود.

گردآوری نتایج

اطلاعات بدست آمده در گامهای قبلی بررسی شده و پس از اطمینان از سازگاری آنها با یکدیگر اشکلات احتمالی موجود رفع می‌گردد.



شکل ۲.۳ نمودار گامهای مرحله بررسی وضعیت موجود (گامهای ۱۲۰ و ۱۳۰ و ۱۴۰ می‌توانند به صورت موازی انجام شوند)

مرحله دوم تعیین گزینه‌های تجاری موجود

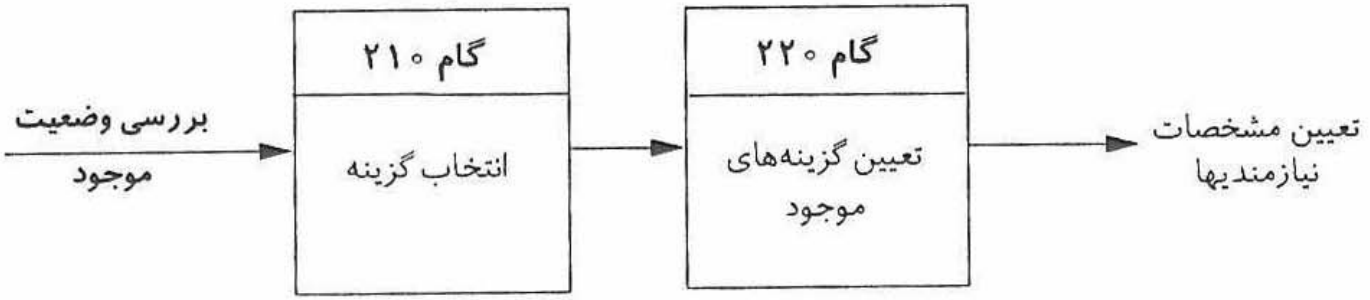
در این مرحله حالتهای مختلفی از سیستم پیشنهادی برای جوابگویی به کاملترین وضعیت تا ساده‌ترین وضعیت مورد بررسی قرار گرفته و بهترین آن با توجه به امکانات انتخاب می‌شود. این مرحله شامل گامهای زیر است:

ـ گام ۲۱۰ تعیین گزینه‌های موجود

در این گام حالت‌های ممکن سیستم پیشنهادی مورد بررسی قرار گرفته و از این میان حداکثر شش حالت با کمک کاربران انتخاب می‌شود و به کمیته بررسی کننده ارائه می‌گردد.

ـ گام ۲۱۰ انتخاب گزینه مورد نظر

از بین گزینه‌های موجود بهترین حالت با توجه به امکانات موجود انتخاب می‌شود.



شکل ۲.۴ نمودار گامهای مرحله دوم (تعیین گزینه‌های تجاری موجود)

ماجول مشخص‌سازی نیازها (مرحله مشخص‌سازی نیازها)

این ماجول شامل یک مرحله است، در این مرحله براساس لیست نیازها و گزینه‌های انتخابی، سیستم پیشنهادی طراحی می‌گردد. نمودار جریان داده‌ها و ساختار منطقی داده‌ها مطابق گزینه انتخابی دوباره ترسیم می‌شود، کارکردها (Functions) شناسایی و کلید ورودیها و خروجیهای سیستم، کلیه رویدادها (Events) در سیستم مشخص شده و آثار آنها بررسی می‌گردد. این مرحله شامل گامهای زیر است:

ـگام ۳۱۰ تعریف پردازش‌ها

نمودار جریان داده‌ها مطابق گزینه انتخابی دوباره ترسیم می‌شود.

ـگام ۳۲۰ ایجاد ساختار منطقی داده‌ها

ساختار منطقی برای سیستم جدید براساس ساختار منطقی داده‌ها در وضعیت موجود ترسیم می‌شود.

ـگام ۳۳۰ شناسایی کارکردها

کارکردهای سیستم به کمک لیست نیازمندیها تعیین شده و لیست آنها تهیه می‌گردد.

ـگام ۳۴۰ تجزیه و تحلیل ساختار منطقی داده‌ها

اشکالات ساختار منطقی داده‌ها برای سیستم جدید مشخص شده و برطرف می‌شود.

ـگام ۳۵۰ نمونه سازی (Protoyping)

مدل سیستم جدید با ابعاد کوچکتر در این گام ایجاد شده و به همراه کاربران مورد بررسی قرار می‌گیرد.

ـگام ۳۶۰ بررسی کارکردها (Functions)

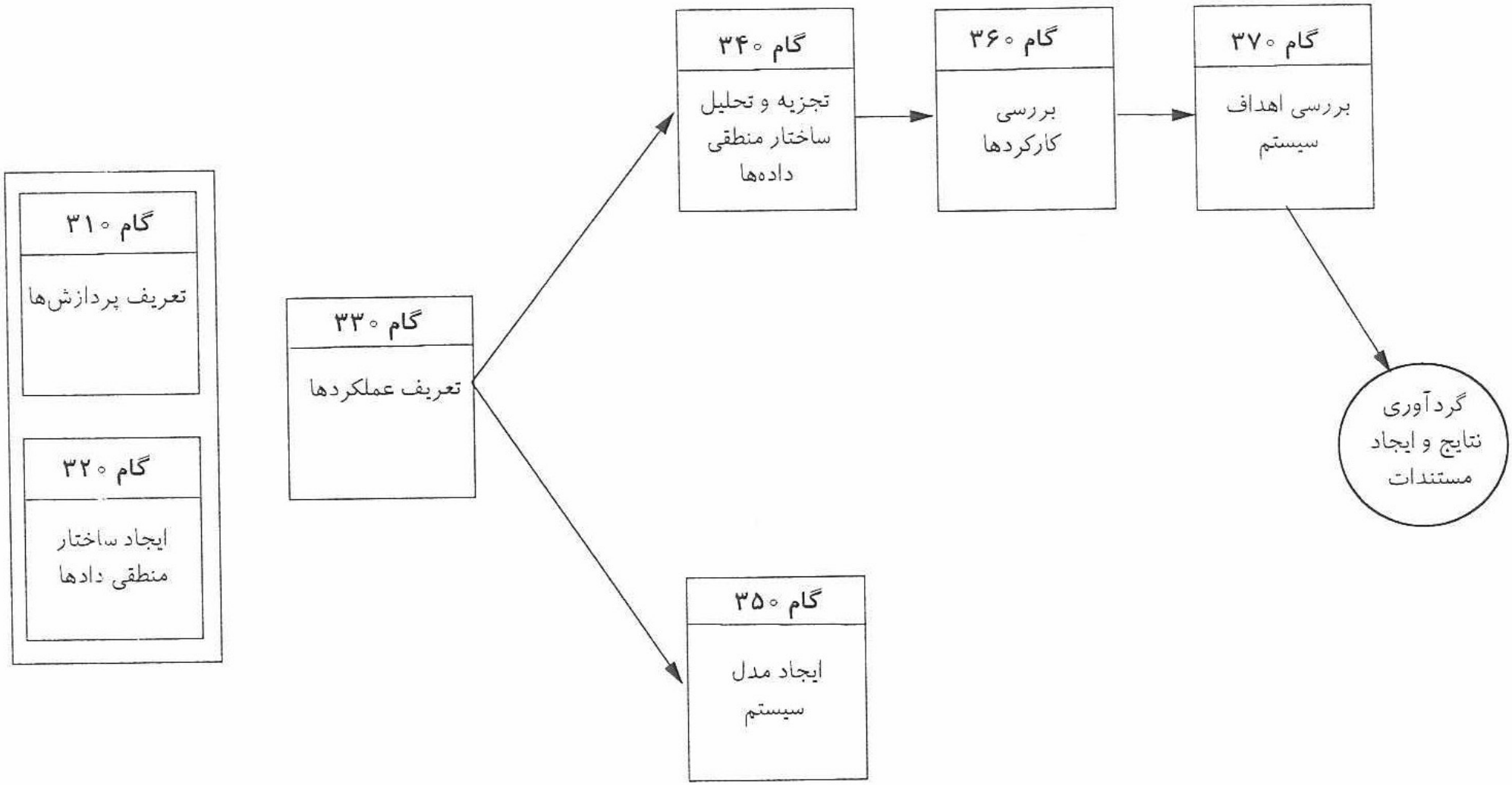
در این گام کارکردهایی که جا افتاده‌اند مشخص و شناسایی می‌شوند.

ـگام ۳۷۰ بررسی اهداف سیستم

در این گام لیست نیازمندیها بررسی می‌شود تا مشخص گردد آیا به کلیه آنها با توجه به استانداردهای مربوطه پاسخ داده شده است یا خیر؟

گردآوری نتایج

نتایج حاصله از این بررسی گردآوری می‌شود و پس از اطمینان از سازگاری و عدم مغایرت این نتایج با یکدیگر ثبت شده و مستند مشخصات نیازمندیها ارائه می‌شود.



شکل ۲.۵ نمودار گامهای مرحله تعیین مشخصات نیازمندیها

ماجول طراحی منطقی

این ماجول شامل دو مرحله بررسی گزینه‌های فنی و طراحی منطقی می‌باشد که در پی می‌آیند.

مرحله چهارم بررسی گزینه‌های فنی

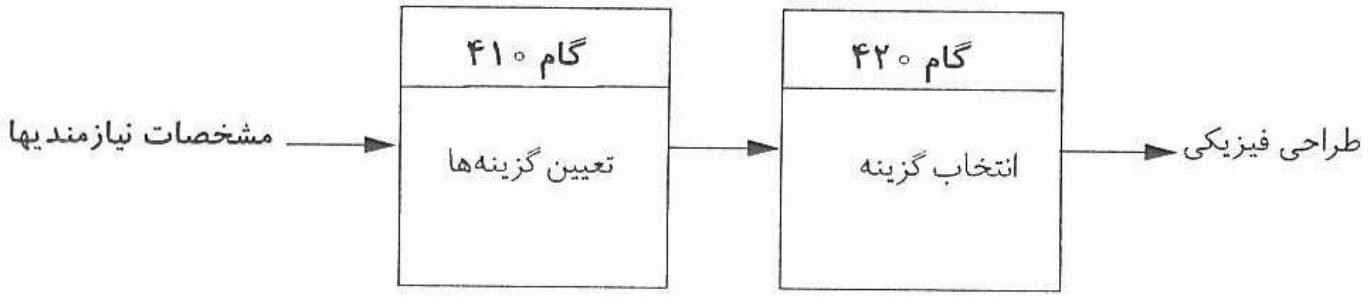
در این مرحله گزینه‌های سخت‌افزاری و نرم‌افزاری مربوطه تعیین و سپس با توجه به امکانات سازمان، بهترین گزینه انتخاب می‌شود. این مرحله شامل گامهای زیر است:

گام ۴۱۰ تعیین گزینه‌های فنی

در این مرحله گزینه‌های مختلف سخت‌افزاری و نرم‌افزاری مشخص می‌شود.

گام ۴۲۰ انتخاب گزینه فنی

براساس شرایط و امکانات سازمان و لیست نیازمندیها گزینه فنی مناسب انتخاب می‌شود.



شکل ۲.۶ نمودار گامهای مرحله چهارم (بررسی گزینه‌های فنی)

مرحله پنجم طراحی منطقی

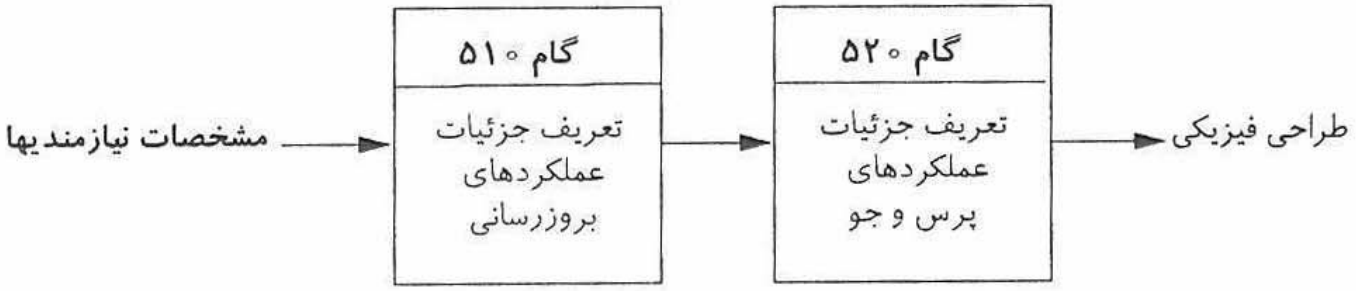
در این مرحله جزئیات هر یک از عملکردهای پرس‌وجو و یا بروز رسانی مشخص می‌شود و شامل گامهای زیر است:

گام ۵۱۰ تعریف مشخصات و جزئیات عملکردهای بروزرسانی

در این مرحله جزئیات بروز رسانی برای هر موجودیت و نشانگرهای مربوطه و مقادیر این نشانگرها (Flags) تعریف می‌شود.

گام ۵۲۰ تعریف مشخصات و جزئیات عملکردهای پرس و جو

در این مرحله جزئیات هر یک از عملکردهای پرس و جو تعیین می‌شود.



شکل ۲.۷ نمودار گامهای مرحله پنجم (طراحی منطقی)

ماجول طراحی فیزیکی (مرحله طراحی فیزیکی)

این ماجول شامل یک مرحله است، در این مرحله ساختار منطقی داده‌ها به طراحی پایگاه اطلاعاتی تبدیل می‌شود و مشخصات و وظیفه هرکارکرد (Function) به شرح برنامه تبدیل می‌شود و شامل گامهای زیر است:

گام ۶۱۰ آمادگی برای طراحی فیزیکی

در این گام استانداردهای طراحی فیزیکی از جمله استانداردهای نامگذاری و طراحی جداول مشخص می‌شود.

گام ۶۲۰ طراحی اولیه پایگاه اطلاعاتی

در این گام نمونه اولیه پایگاه اطلاعاتی براساس ساختار منطقی داده‌ها ایجاد می‌شود.

گام ۶۳۰ طراحی روش‌های ارتباط کاربران با سیستم (گفتگوهای سیستم)

در این گام براساس مشخصات هر عملکرد و نقش هر کاربر صفحات (Screen) ورودی و خروجی طراحی می‌گردد.

گام ۶۴۰ ایجاد جداول FCIM و پیاده‌سازی کارکردها

در این گام جدولی ایجاد می‌شود که در آن برنامه‌های مورد نیاز هر کارکرد مشخص می‌گردد.

گام ۶۵۰ ایجاد شرح برنامه‌ها و مشخصات آنها

در این گام وظایف و مشخصات هر برنامه به ریز مشخص می‌شود تا برنامه‌نویس مربوطه بتواند آنرا ایجاد کند.

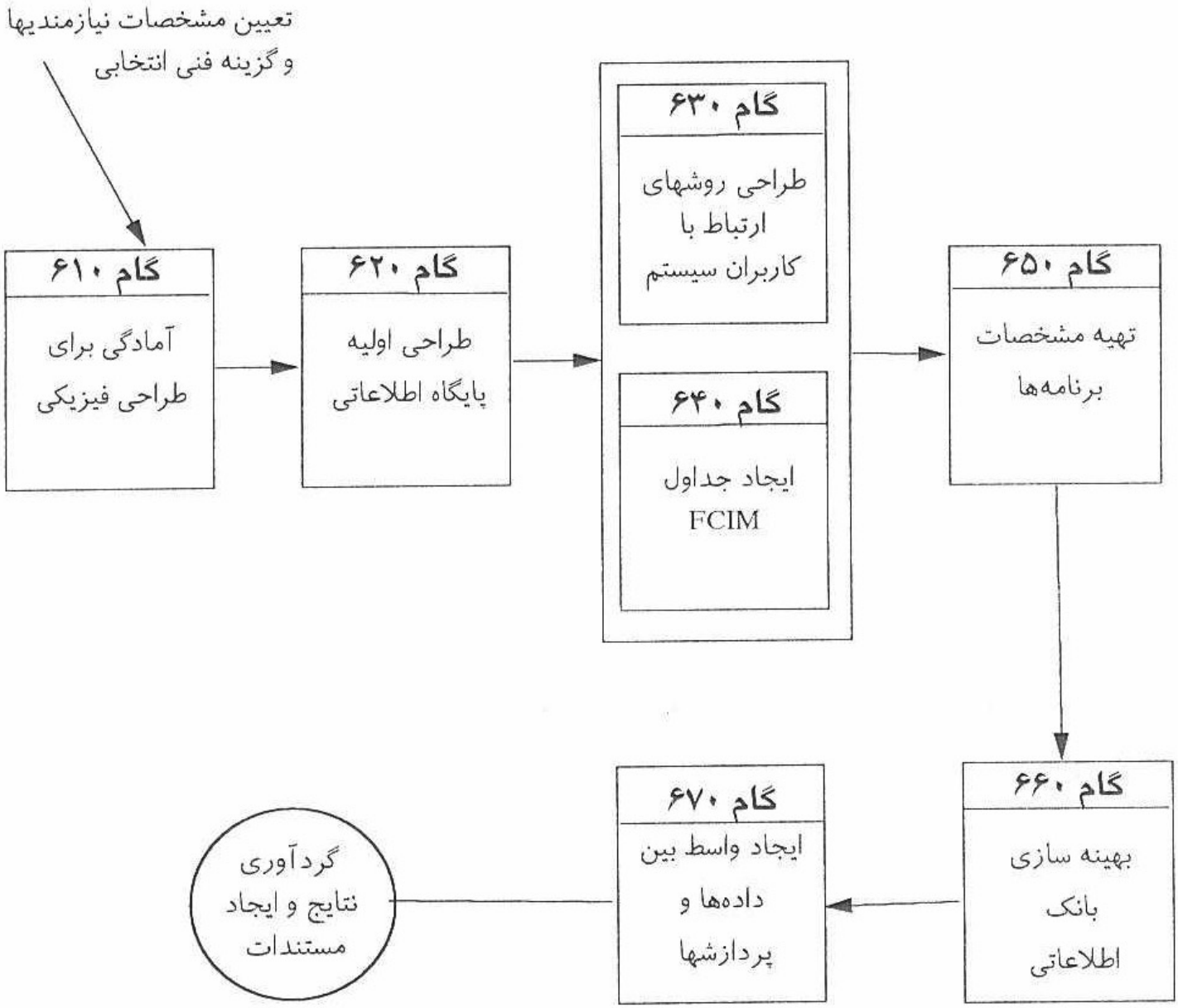
گام ۶۶۰ بهینه‌سازی بانک اطلاعاتی

مشکلات و نواقص احتمالی بانک اطلاعاتی طراحی شده در این مرحله برطرف می‌گردد.

گام ۶۷۰ ایجاد واسط بین داده‌ها و پردازش‌ها

در این گام یک واسط نرم‌افزاری جهت مطابقت مدیریت بانک اطلاعاتی سلسله مراتبی یا شبکه‌ای با ساختار منطقی داده‌ها ایجاد می‌گردد.

سپس نتایج حاصل گردآوری و مستند می‌شود.



شکل ۲.۸ گامهای مختلف طراحی فیزیکی

فصل سوم

امکان سنجی

اهداف فصل

هدف از مطالعهٔ امکان سنجی بررسی وضعیت موجود و سیستم اطلاعاتی آینده و توانائیهای آن و مطابقت دادن این توانائیها با نیازمندیهای سازمان می‌باشد و پس از بررسی مشخص میگردد که آیا چنین سیستمی قادر هست تا نیازمندیهای سازمان را پاسخگو باشد یا خیر.

مطالعه امکان سنجی معمولاً پیش‌نیاز یک مطالعهٔ کامل و جامع وضعیت موجود و طراحی سیستم آینده می‌باشد.

SSADM با مشخص کردن فعالیتهای این مرحله به شناسایی وضعیت سیستم فعلی و مشکلات موجود در آن کمک می‌کند و موجب می‌شود که ارائه راه‌حل‌های ممکن جهت حل مسائل و مشکلات موجود، آسان شود.

بطور کلی اهداف این مرحله به شرح زیر است:

۱. تعیین امکانات سیستم پیشنهادی و مقایسه آن با نیازهای سازمان

۲. بررسی امکان مطالعه جامع وضعیت موجود و طراحی سیستم اطلاعاتی مناسب

۳. کمک به کاربران برای تعیین بهترین راه‌حل پیشنهادی

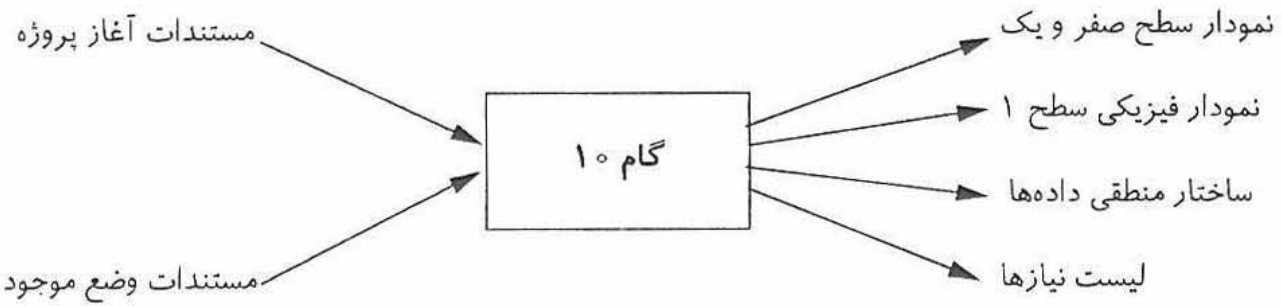
گامهای این مرحله به شرح زیر است:

گام ۱۰: آمادگی جهت امکان سنجی

در این گام پس از بررسی مستندات موجود در آغاز پروژه و سایر مستندات سیستم به شناسایی محدودهٔ سیستم می‌پردازیم و سپس نیازهای سیستم جدید را مشخص می‌کنیم. نیازها باید به روشنی تعریف شده و قابل دستیابی باشند. نباید در تعریف آنها هیچ‌گونه ابهامی وجود داشته باشد.

پس از شناسایی نیازمندیها، آنها را در لیستی به همین نام قرار می‌دهیم این لیست یکی از محصولات این گام می‌باشد سپس نمودار سطح صفر و سطح ۱ وضعیت موجود تهیه گردیده و ساختار منطقی داده‌ها نیز بطور کلی با همکاری کاربران تهیه می‌گردد.

و در پایان به شناخت منابع و مراجعی که می‌توانند اطلاعات مورد نیاز را در اختیار ما قرار دهند می‌پردازیم.



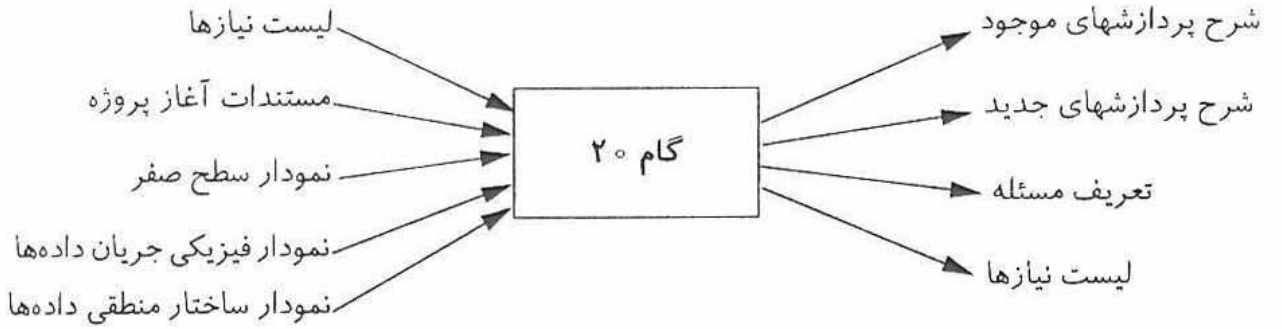
شکل ۳.۱ نمودار ورودیها و خروجیهای گام ۱۰

گام ۲۰ـ تعریف مسئله

در این گام وضعیت موجود شناسایی می‌گردد و پردازشهای مهم، پردازشهایی که در سیستم اطلاعاتی حتماً می‌بایستی وجود داشته باشند، مشخص می‌گردد و برای اینکار می‌توان از نمودار جریان داده‌ها و ساختار منطقی داده‌ها که در گام پیش ایجاد گردید استفاده کرد. سپس به شناخت نقاط ضعف و نقائص وضعیت موجود می‌پردازیم و بعد از آن پردازشهای جدیدی را که در سیستم اطلاعاتی باید موجود باشد، شناسایی می‌نمائیم.

سپس این پردازشها را به لیست نیازها اضافه کرده و کاربران سیستم جدید و نقش و وظایف آنها را تعیین می‌کنیم.

در پایان خلاصه نیازها را بر حسب اولویت در گزارشی تحت عنوان تعریف مسئله تهیه کرده و برای ارائه به کاربران آماده می‌کنیم تا از طرف آنها مورد بررسی قرار گیرد.

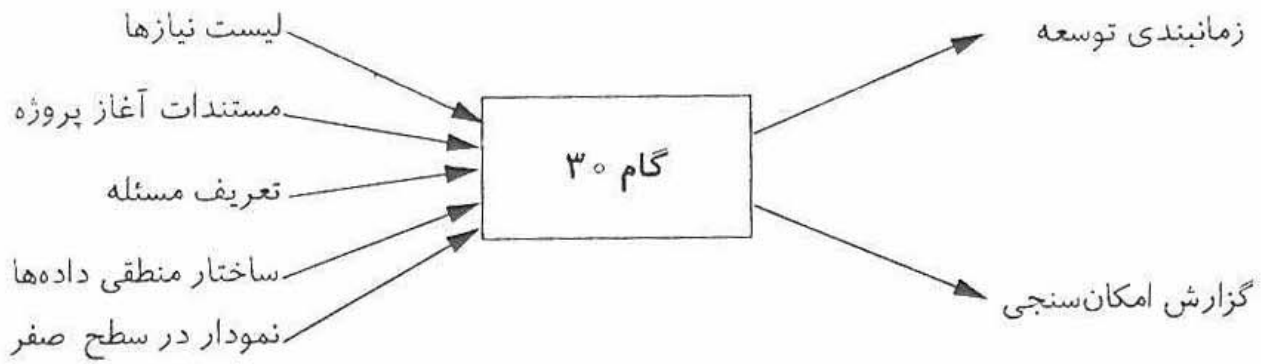


شکل ۳.۲ نمودار ورودیها و خروجیهای گام ۲۰

اما نکته اساسی این است که همواره به یاد داشته باشیم که در این مرحله نبایستی به جزئیات بپردازیم بلکه یک دیدگاه کلی کافی است چون این خطر وجود دارد که پرداختن به جزئیات سبب افزایش زمان و هزینه گردد.

گام ۳۰ـ تعیین راه‌حلهای موجود

پس از گام تعریف مسئله و شناسائی نیازهای اساسی واصلی کاربران در این گام راه‌کارهای موجود برای برآورد نیازهای کاربران پس از بررسی گزینه‌های فنی و تجاری ممکن باهمکاری کاربران مشخص می‌گردد هر راه‌کار، با در نظر گرفتن گزینه‌های فنی بصورت محدود تشریح می‌گردد و با همکاری کاربران حداکثر شش راه‌کار تجاری در نظر گرفته می‌شود و گزینه‌های فنی موجود نیز برای آنها بررسی می‌گردد و سپس راه‌کارهای بدست آمده برای بررسی نهایی به کمیته راهبری پروژه ارائه می‌گردد تا از بین آنها یکی را انتخاب کند.



هر گزینه می‌تواند از جهات زیر مورد بررسی قرار داد

- مزایای راه‌حل و معایب آن را نمایش داد.
- به کاربران نشان داد که پس از انتخاب گزینه چگونه اهداف سازمان محقق می‌شود
- استراتژی پیاده‌سازی هرگزینه را در نظر گرفت و شرح داد.

فصل چهارم

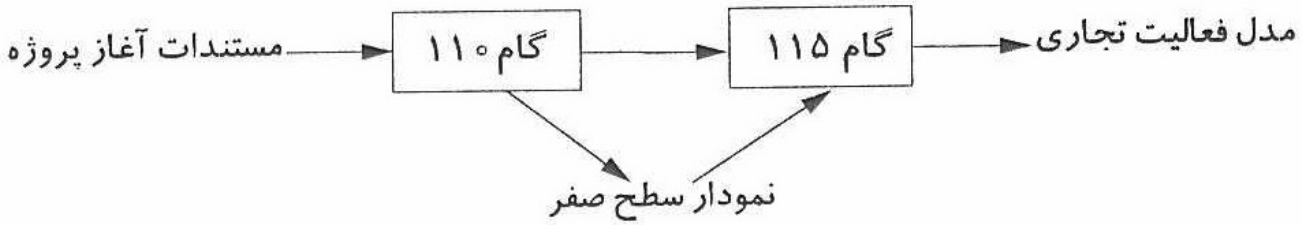
مدل فعالیت تجاری

بررسی مقدماتی

اولین گام برای شناخت وضعیت موجود گام ۱۱۰ می‌باشد در این گام تحلیلگر سعی می‌کند درک درستی از حوزهٔ عملیات سیستم و وظایف آن پیدا کند زیرا که در غیر اینصورت تحلیگر مسائلی را بررسی می‌کند که ربطی به سیستم ندارد و در حوزهٔ عملیات آن نیست و ممکن است در آینده باعث شکست سیستم و عدم مقبولیت آن توسط کاربران گردد. اهم فعالیتها در این گام عبارتند از:

- بررسی مستندات موجود (مستندات آغاز پروژه و مطالعات امکان سنجی) و تعیین نیازمندیهای اصلی و حصول اطمینان از بروز بودن این مستندات و بررسی این مطلب که آیا این نیازها همچنان به قوت خود باقی هستند یا خیر.
- تعیین حوزه عملیات براساس مستندات موجود
- بررسی و شناخت کاربران اصلی سیستم
- برنامه‌ریزی برای مراحل بعدی به عبارت دیگر تعیین اینکه چه مستنداتی، چطور و چگونه بایستی تولید شود. چه گامها و مراحلی از SSADM می‌تواند انجام شود و کدامیک می‌تواند حذف گردد و مکانیزم‌های کنترل پروژه چه باشد؟

در پایان این گام تحلیلگر بایستی اهداف سیستم را بخوبی بشناسد و بتواند نمودار سطح صفر جریان داده‌ها (Context Diagram) را ترسیم و فعالیتهای سیستم را بطور کلی تشریح کند.



شکل ۴.۱ نمودار ارتباط گامهای ۱۱۰ و ۱۱۵ و ورودیها و خروجیهای آنها

مدل فعالیت تجاری چیست

SSADM به فعالیتهای سازمان بیش از هر چیز توجه می‌کند. برای شناخت هر چه بهتر این فعالیتها مدل فعالیت تجاری را ایجاد می‌کند تا بوسیله این مدل فعالیتها را بررسی نماید. برای ایجاد مدل فعالیت تجاری بایستی به چهار سئوال اساسی زیر پاسخ داده شود.

۱. چه فعالیتهایی انجام می‌شود

در این قسمت فعالیتهای اصلی سیستم بررسی گردیده و به صورت زیر دسته‌بندی می‌شود.

- **فعالیتهای اصلی**

فعالیتهایی است که به منظور انجام وظایف اصلی سیستم صورت می‌گیرد بطور مثال صدور سند حسابداری یکی از فعالیتهای اصلی سیستم حسابداری می‌باشد.

- **فعالیتهای جانبی**

فعالیتهای درجه ۲ و کناری می‌باشد. مانند تهیه گزارشات موردی .

- **برنامه‌ریزی**

فعالیتهایی که به منظور برنامه‌ریزی صورت می‌گیرد را شامل می‌شود مانند تهیه گزارشهایی از وضعیت مصرف یک کالا که می‌تواند برای برنامه‌ریزی خرید آن کالا در سال بعد استفاده شود.

- **فعالیتهای کنترلی**

تهیه گزارشات مختلف کنترلی که نشان دهنده کارآیی سیستم می‌باشد در زمره فعالیتهای کنترلی است

۲. چرا این فعالیتها انجام می‌شود

جواب به این سؤال اهداف و وظایف اصلی را روشن می‌کند ممکن است پاسخ به این سئوال ساده نباشد. بنابراین سئوال را می‌توان از جوانب مختلف بررسی کرد. با اشخاص مختلف در میان گذاشت و در پایان با جمع‌بندی نظرات، اهداف سیستم را روشن ساخت.

۳. چه موقع این فعالیتها صورت می‌گیرد

در یک سیستم هر فعالیت می‌تواند سه منشاء داخلی، خارجی و زمانبندی شده داشته باشد. بعنوان مثال صدور صورتحساب مشتری براساس درخواست وی دارای منشاء خارجی (خارج از سیستم) می‌باشد و یا تصمیم به افزایش قیمیت یک کالا توسط مدیریت دارای منشاء داخلی به حساب می‌آید و گزارشاتی که بایستی در پایان روز تهیه گردد و براساس زمانبندی معین انجام می‌شود.

۴. چگونه این فعالیتها صورت می‌گیرد

جواب به این سئوال قواعد حاکم بر سیستم (Bussiness Rule) را مشخص می‌کند. این قواعد را می‌توان به دو دسته تقسیم کرد.

ـ محدودیتها، کنترلهایی که در سیستم موجود است در این قسمت قرار می‌گیرند فی‌المثل در یک سیستم تسهیلات نمی‌توان بیشتر از اعتبار یک مشتری به او تسهیلات پرداخت کرد مگر با تائید مدیریت.

ـ خط‌مشی‌ها، قواعدی که راه‌کارها را نشان می‌دهند در این قسمت قرار می‌گیرند فی‌المثل دستور العمل محاسبه مالیات و یا دستورالعمل محاسبه جریمه.

در SSADM خط‌مشی مشخصی برای ایجاد مدل تجاری ارائه نشده است و این مطلب در راهنماهای مربوطه بیان گردیده و آنرا خارج از محدودهٔ SSADM دانسته‌اند یکی از متدهای ایجاد آن خصوصاً برای ایجاد و توسعهٔ سریع یک سیستم متد تجزیه و تحلیل عملکردی می‌باشد که در این کتاب به آن اشاره می‌شود امّا SSADM هر روشی را که واجد خصوصیات زیر باشد می‌پذیرد این خصوصیات عبارتند از:

ـ تعریف اهداف سیستم

ـ تعریف نحوهٔ ارتباط فعالیتها با یکدیگر

ـ تعیین روشهایی برای اندازه‌گیری کارآیی

ـ تعیین مکانیزمهایی برای کنترل

ـ تعیین رویه‌هایی برای تصمیم‌گیری

ـ تعریف محدودهٔ سیستم

ـ تعریف منابع

ـ سلسله مراتب سیستم

ایجاد مدل فعالیت تجاری

برای تهیه یک مدل فعالیت تجاری مستندات زیر ایجاد می‌شود.

ـ شرح اهداف سیستم

ـ تجزیه و تحلیل عملکردها

ـ نمودار سطح صفر جریان داده‌ها

ـ شرح فعالیتهای سیستم

برای تشریح اهداف سیستم می‌توان یک تعریف کوتاه و مختصر از علت وجود سیستم و اهداف آن در حداکثر یک یا دو پاراگراف ارائه داد در بقیه کتاب بنگاه اجاره متل، را در نظر می‌گیریم این بنگاه کاروانهای خود را که در مناطق خوش آب و هوا واقع است به مردم اجاره می‌دهد در شکل ۴.۲ اهداف این مؤسسه بیان گردیده است.

هدف اصلی بنگاه اجاره متل اجاره کاروان به مردم در روزهای تعطیل می‌باشد. کاروانها توسط این شرکت در زمینهای متعلق به خود بنگاه یا دیگران تعبیه شده و اجاره داده می‌شود.

شکل ۴.۲ اهداف بنگاه اجاره متل

روش تجزیه و تحلیل عملکردها چیست

در این روش ابتدا فعالیتهای سیستم حداکثر به هفت یا هشت فعالیت اصلی تقسیم‌بندی می‌شود متأسفانه هیچ روش مشخصی برای تجزیه و تحلیل فعالیتها وجود ندارد ولی این فعالیتها باید به نحوی تقسیم‌بندی شود که تمام فعالیتهای مرتبط با یکدیگر با هم باشند و تحلیلگر بایستی از پرداختن به جزئیات خودداری و اهداف اصلی را در نظر داشته باشد.

یک سیستم مدیریت انبار را در نظر بگیرید می‌توان فعالیتهای اصلی سیستم را به شرح زیر دسته‌بندی کرد:

- ثبت و پذیرش درخواست خرید
- صدور گزارش اقلام مورد نیاز و زیرنقطه سفارش و گزارشات مرتبط با کاردکس کالا
- ثبت و پذیرش اطلاعات قسمت‌ها و قطعات و کالاها
- ثبت و نگهداری اطلاعات انبارگردانی

در اینجا تصویری کلی از سیستم مدیریت انبار ارائه شده ولی این تقسیم‌بندی نشان می‌دهد که چگونه بایستی به مسئله نگاه کرد. حال می‌توان فعالیتهای اصلی سیستم رزرواسیون بنگاه اجاره متل را به شرح زیر بیان کرد:

- ثبت اطلاعات رزرو و مشتریان
- ثبت اطلاعات همراهان مشتری

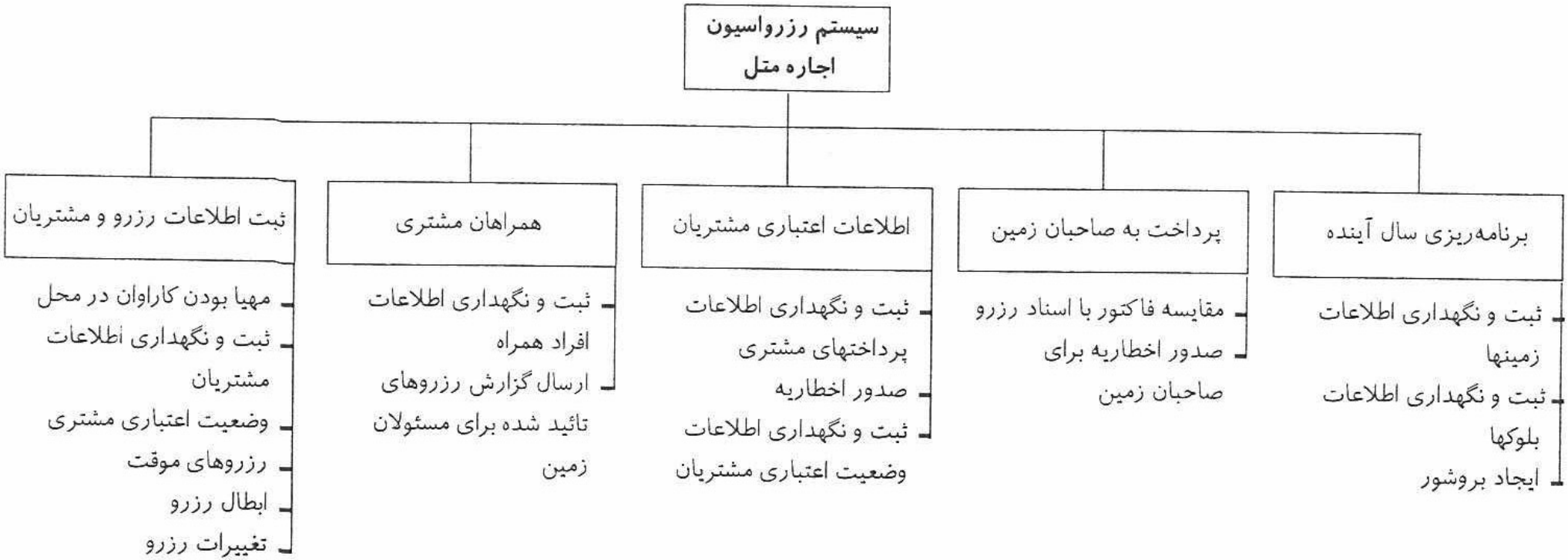
- ثبت اطلاعات پرداختهای هر مشتری
- ثبت اطلاعات پرداخت سالانه به صاحبان زمین
- برنامه‌ریزی سال آینده

قدم بعدی تجزیه هر یک از این فعالیتها به فعالیتهای ریزتر می‌باشد برای هر فعالیت از خود می‌پرسم که چه فعالیتهایی برای رسیدن به این هدف لازم است بنابراین فعالیت اول «ثبت اطلاعات رزرو و مشتریان» را می‌توان به زیرفعالیتهای زیر تقسیم کرد:

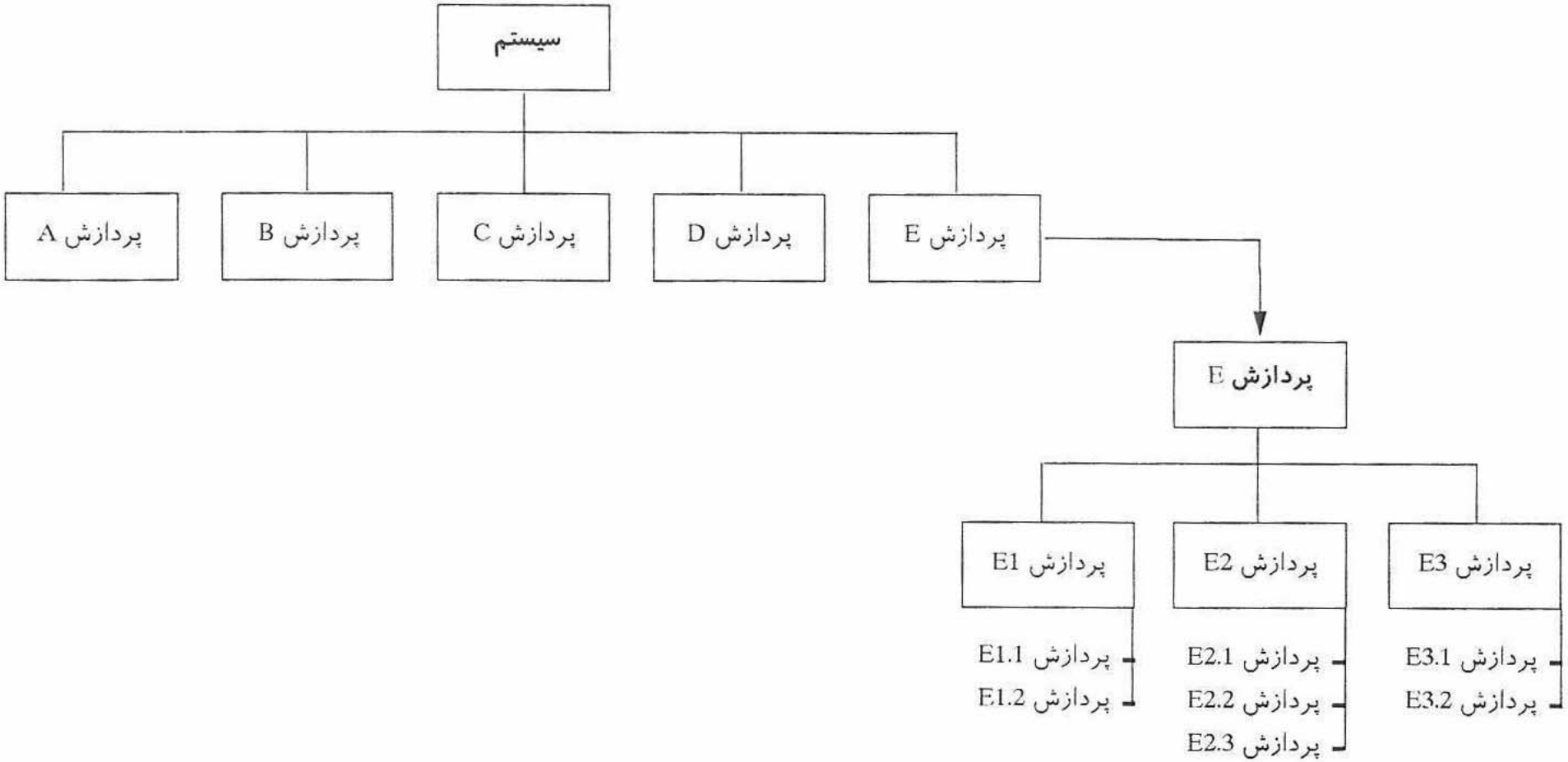
- مهیا بودن کاروان در محل
- اطلاعات مشتریان
- اطلاعات اعتبار مشتریان
- رزروهای موقتی
- ابطال رزرو
- تغییرات رزرو

شکل ۴.۳ فعالیتهای اصلی وریز فعالیتهای سیستم رزورواسیون اجاره متل را نشان می‌دهد مسلماً یک سیستم بزرگتر تجزیه و تحلیل بیشتری می‌طلبد ولی مهم اینست که هر فعالیت حداکثر به هفت یا هشت زیرفعالیت و هر کدام زیر فعالیتها نیز حداکثر به هفت یا هشت بخش تقسیم شود و همانطور که نمودار جریان داده‌ها معمولاً از سه سطح یا چهار سطح پایین‌تر نمی‌رود تجزیه و تحلیل نیز این این تعداد نبایستی فراتر رود.

تمامی فعالیتها در آخرین سطح براثر یک رویداد تجاری (Business Event) ایجاد شده و منطبق با توالی این فعالیتها باشد برای مثال فعالیت «مهیا بودن کاروان» درجهت این نیاز و سئوال بنگاه است که آیا کاروان خالی در محل موردنظر موجوداست یا خیر. ممکن است یک زیر فعالیت در چند فعالیت استفاده شود در این صورت در چند سطح تکرار گردد این بدان معنا نیست که بایستی آنها را جدا در نظر گرفت و هر یک از آنها بصورت جداگانه طراحی و ایجاد شود بلکه بایستی آن را شناخت تا از دوباره کاریها پرهیز گردد. برای نمایش بهتر، این قبیل فعالیتها را می‌توان جداگانه نمایش داد. هر تجزیه و تحلیل عملکرد از فعالیتهای سیستم اگر از یک یا دو سطح تشکیل شده باشد می‌توان به صورت شکل ۴.۳ در یک صفحه A4 یا حداکثر A3 نمایش داد. اما اگر تعداد این سطوح بیشتر باشد بایستی آنرا بصورت شکل ۴.۴ در چند صفحه نمایش داد که در صفحه اول اولین سطح و هر یک از فعالیتها در سایر صفحات به زیر فعالیت تقسیم می‌شود.



شکل ۴.۳ نمودار تجزیه و تحلیل عملکردها



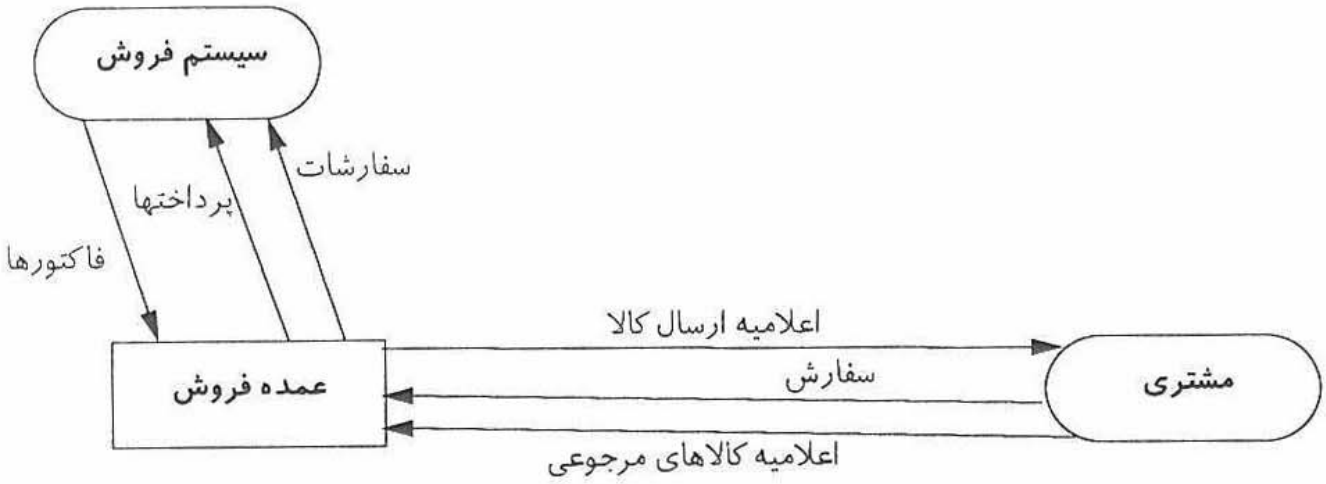
شکل ۴.۴ نمایش تجزیه و تحلیل عملکردها (سه سطح و یا بیشتر)

قبل از نمایش نمودار تجزیه و تحلیل عملکرد به کاربران بایستی از کامل بودن نمودار تجزیه و تحلیل عملکرد اطمینان حاصل کرد و مطمئن شد که تمامی فعالیتهای اصلی و جانبی و فعالیتهای برنامه‌ریزی و کنترل در آن نمایش داده شده است.

نمودار سطح صفر

نمودار تجزیه و تحلیل عملکردها ارتباط داخلی بین اجزای سیستم را نشان می‌دهد و ارتباط سیستم با سایر سیستم‌ها و موجودیتها را نشان نمی‌دهد برای این منظور بایستی از نمودار سطح صفر جریان داده‌ها که ارتباطات سیستم را نشان می‌دهد و در گام ۱۱۰ به آن اشاره شده، استفاده کرد. برای رسم این نمودار، کل سیستم را به شکل یک پردازش در نظر می‌گیریم و تمام مستنداتی که بایستی رد و بدل می‌شود را فهرست کرده سپس تمامی منابع و دریافت کنندگان این مستندات را شناسایی می‌کنیم.

یک مستطیل بزرگ نشان دهنده کل سیستم و تمامی موجودیتهای خارجی با بیضی نشان داده شده و مستنداتی که بین آنها رد و بدل می‌شود با نام آنها نشان داده می‌شود.



شکل ۴.۵ نمودار سطح صفر جریان داده‌ها (Content Diagram)

تشریح هر فعالیت

نمودار سطح صفر و نمودار تجزیه و تحلیل عملکردها بیانگر ارتباطات داخلی و خارجی سیستم می‌باشند ولی نشان نمی‌دهند که هر فعالیت چطور، چگونه و در چه زمانی انجام می‌شود؟ برای این منظور می‌توان از فرمی استفاده کرده و در آن به تشریح هر فعالیت پرداخت و مشخص کرد که چطور، چگونه و با توجه به چه پارامترهای هر فعالیت اجرا می‌شود.

فصل پنجم

تعیین مشخصات

نیازمندیها (گام ۱۲۰)

اهداف فصل

هدف از این گام شناسایی کاربران سیستم و نیازهای آنان است. در این مرحله لیست نیازمندی‌ها و لیست کاربران تهیه می‌شود. در ابتدا براساس مستندات آغاز پروژه و گزارش امکان‌سنجی (در صورت وجود) نیازها به صورت کلی شناسایی می‌شوند.

چگونگی تشخیص نیازها

در SSADM نیازها به دو گروه کارکردی (Functional) و غیرکارکردی (Non-Functional) تقسیم می‌شوند.

گروه اول شامل نیازهای اصلی است که بایستی با طراحی مناسب و نوشتن برنامه‌های لازم به آن پاسخ گفت. این نیازها عمدتاً شامل گزارشات و پرس‌وجوهای مختلف سیستم است. لازم است به این مطلب توجه کنیم که در این گام فقط نیازمندی‌های سیستم جدید شناسایی می‌شود و از پرداختن به جزئیات و در نظر گرفتن این مطلب که چه اقلام اطلاعاتی باید نگهداری شود و کدامیک بروز گردد اجتناب می‌کنیم و گزارشات و پرس‌وجوها به شکل کلی در لیست نیازمندیها وارد گردد. مطلب دیگری که در این گام بایستی بدان توجه کرد آن است که کاربران گاهی به درستی نمی‌دانند که چه می‌خواهند؟ علت این مسأله گاهی نداشتن درک صحیحی از وضعیت موجود و گاهی عدم اطلاع از قابلیت‌های سیستم کاربردی می‌باشد، در چنین مواقعی پس از پیاده‌سازی سیستم و کار با آن کاربران با قابلیت‌ها و توانایی‌های آن آشنا می‌گردند.

SSADM برای حل این مسأله روش نمونه‌سازی (Prototype) را پیشنهاد می‌کند. حتی موقعی که نیازمندیها کاملاً مشخص می‌باشد، ممکن است در بین کاربران اختلاف‌نظر وجود داشته باشد. این اختلاف نظر گاهی افقی و گاهی عمودی است اختلاف‌نظر عمودی اختلاف بین مدیریت و کارکنان می‌باشد که اگر این اختلاف رفع نشود تحلیل‌گران معمولاً نظر مدیریت را مدنظر قرار می‌دهند. اختلاف‌نظر افقی اختلاف‌نظر بین کاربران هم‌سطح است برای مثال سازمانی را در نظر بگیرید که شعب زیادی دارد (مثل بانک) کارکنان شعب ممکن است نظرات مختلفی در مورد سیستم جدید داشته باشند در این صورت می‌توان یک یا چند نماینده از بین آنان انتخاب و براساس نظریات آنان کار را ادامه داد.

گروه دوم شامل نیازمندی‌های غیرکارکردی (Non-Functional) می‌باشد. بررسی این نیازمندی‌ها از نیازمندی‌های کارکردی پیچیده‌تر است و آنها را می‌توان به صورت زیر دسته‌بندی کرد:

۱. میزان سرویس‌دهی (Service level Requirement)

نیازهایی که در رابطه با سرویس‌دهی سیستم می‌باشد در این گروه قرار می‌گیرند به طور مثال زمان پاسخ‌دهی سیستم و یا تعداد تراکنش‌هایی که سیستم در مدت معین انجام می‌دهد.

۲. نحوه دسترسی

نیازهایی که در رابطه با اطلاعات و نحوهٔ دسترسی به این اطلاعات می‌باشد، در این گروه قرار دارند.

نیازهایی از این قبیل که چه کسانی بایستی اطلاعات سیستم را به روز درآورند یا نحوهٔ رمزگذاری (Password) چگونه باشد و یا نحوه رمز کردن (Encryption).

۳. امنیت داده‌ها

نیازهایی که در رابطه با حفظ و نگهداری اطلاعات مطرح می‌شود در این گروه قرار دارد که شامل:

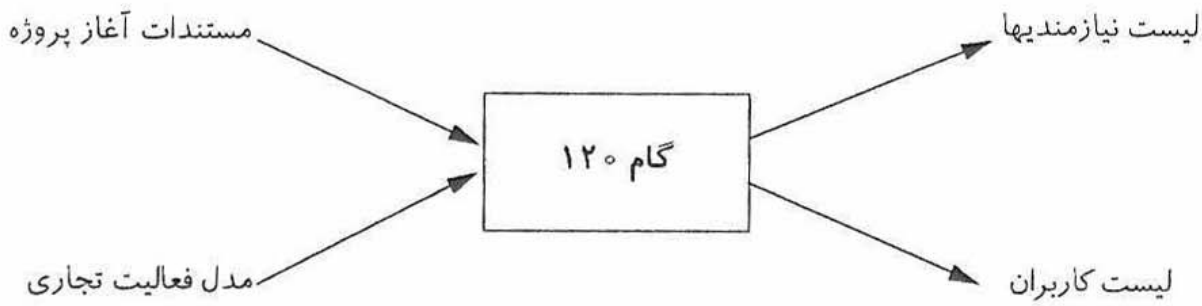
روش Back upگیری از اطلاعات سیستم و یا نحوهٔ برگرداندن اطلاعات در صورت بروز خرابی می‌باشد.

۴. گزارشات کنترلی

در این گروه نیازهایی قرار می‌گیرد که در رابطه با کاربران، نحوه عملکرد آنان و نحوه عملکرد سیستم می‌باشد. (به طور مثال چه کسانی یک تراکنش خاص را تأیید کرده‌اند)

۵. محدودیت‌ها

محدودیت‌هایی که برای سیستم وجود دارد در این قسمت بررسی می‌گردد به طور مثال اگر شرط استفاده از اطلاعات سیستم قدیمی وجود دارد نحوه تبدیل این اطلاعات و استفاده از آن در سیستم جدید چگونه باشد.



۵.۱ نمودار ورودی‌ها و خروجی‌های این گام

بطور کلی می‌توان گفت: کلیه نیازمندی‌هایی که بتوان برای آنها عبارت «انجام شده است یا خیر» را به‌کار برد نیازمندی‌های کارکردی و در مورد آنهایی که از عبارت «تا چند درصد و تا چه میزان انجام شده است» را استفاده نمود نیازمندی‌های غیرکارکردی سیستم می‌باشند.

به طور مثال این مطلب که آیا مشخصات مشتری نمایش داده می‌شود یا خیر یک نیاز کارکردی است. اگر این معیار اختیار گردد که حداکثر زمان پاسخ‌دهی سیستم ۲ ثانیه باشد و نیز در ۹۰ درصد موارد زمان پاسخ‌دهی زیر ۲ ثانیه باشد می‌توانیم بگوییم که در ۹۰ درصد مواقع زمان پاسخ‌دهی ۲ ثانیه است. این مثال معرف یک نیاز غیرکارکردی می‌باشد.

میزان سرویس‌دهی سیستم بهترین مثال برای نشان دادن نیازهای غیرکارکردی سیستم می‌باشد. به طور کلی غیر از موارد بالا، SSADM پیشنهاد می‌کند که نیازهای غیرکارکردی زیر را در نظر بگیرید:

- کارآیی
- میزان یادگیری
- میزان رضایت
- تعداد اشتباهات

قابلیت نگهداری و انتقال سیستم بر روی محیط‌های نرم‌افزاری مختلف نیز از سایر نیازهای غیر کارکردی است.

در گذشته چنین نیازهایی نادیده گرفته می‌شد و تنها مطلب مورد نظر عملکرد درست سیستم بود ولی اگر سیستم نیازهای کاربر را به شکل نامطلوبی انجام دهد بطوریکه کاربران راحت نباشند در اینصورت سیستم کنار گذاشته خواهد شد.

بهر حال در مورد هر کارکرد (Function)، بایستی این مسئله که از چه نوعی است را تشخیص دهیم و اگر غیرکارکردی است ابزارهایی را برای تست و اندازه‌گیری آن ارائه نمائیم. و این ابزارها می‌تواند پرسشنامه و یا بررسی آماری باشد شکل ۵.۲ یک فرم برای سنجش یک نیاز غیرکارکردی را نشان می‌دهد.

ایجاد لیست کاربران

آخرین کار در این گام تهیه فهرست کاربران و نقش هر یک از آنان می‌باشد. شکل ۵.۴ نمونه‌ای از این لیست را نشان می‌دهد.

لیست نیازمندیهای غیر کارکردی (Nonfunctional)		
نام سیستم تاریخ نسخه نام تحلیلگر		
شرح نیازمندی	منبع	درجه‌اولویت
کاهش اشتباهات اپراتور در هنگام ورود اطلاعات رزرو	گزارش امکان سنجی	زیاد
نتایج: تعداد اشتباهات در ۱۰۰ رزرو هدف: حداکثر ۴ اشتباه مقدار کنونی: هشت اشتباه		
روش تست: هر اپراتور بصورت سه ماه یکبار آزمایش می‌شود.		

شکل ۵.۲ نمونه‌ای از لیست نیازهای غیر کارکردی

لیست نیازهای کارکردی (Functional)					
نام سیستم	تاریخ	نسخه	نام تحلیلگر	شماره‌صفحه	
۱. اتوماسیون سیستم انبار				منبع	اولویت

شکل ۵.۳ نمونه‌ای از لیست نیازهای غیر کارکردی

فهرست کاربران				
نام سیستم	نام تحلیلگر	تاریخ	نسخه	شماره‌صفحه
عنوان شغل	شرح وظایف			
۱. مدیر امور مشتریان	نظارت بر کار کارکنان و همیاری آنان در وقت نیاز و تائید اعتبار مشتریان			
۲. کارمند امور مشتریان	پاسخگویی به مشتریان ـ ثبت رزروهای موقت			

شکل ۵.۴ فهرست کاربران و وظایف آنان (User Catalogue)

فصل ششم

بررسی وضع موجود

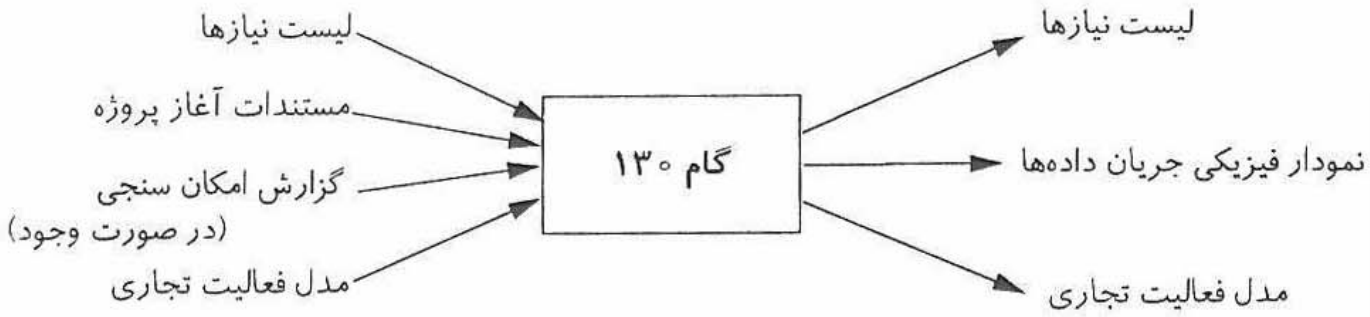
اهداف فصل

یکی از اهداف اصلی در مرحله تجزیه و تحلیل نیازها، شناخت و بررسی وضع موجود است. در SSADM پس از شناخت وضعیت موجود، از دو ابزار نمودار فیزیکی جریان داده‌ها و نمودار ساختار منطقی داده‌ها برای نمایش اطلاعات وضعیت موجود استفاده می‌گردد.

چگونگی رسم این دو نمودار در این فصل تشریح می‌گردد. در SSADM از نمودار جریان داده‌ها که ابزار مناسبی برای ارتباط با کاربران می‌باشد بسیار استفاده می‌گردد. از این ابزار در سایر متدولوژیهای ساخت یافته نیز استفاده می‌شود.

گام ۱۳۰ بررسی وضعیت موجود

در گام ۱۳۰ نمودار فیزیکی جریان داده‌ها، که شامل یک سری نمودار برای نمایش وضعیت موجود است، ایجاد می‌شود. لیست نیازها نیز در این گام بروز آوری می‌شود زیرا نتیجه بررسی وضعیت موجود ممکن است نیازهای جدیدی را مشخص سازد. ورودیهای این گام مدل فعالیت تجاری و مستندات آغاز پروژه و گزارش امکان سنجی (در صورت وجود) می‌باشد.



شکل ۶.۱ ورودی و خروجی گام ۱۳۰

هدف از این گام مستندسازی پردازش‌های وضع موجود و تعیین حدود سیستم مورد مطالعه می‌باشد. برای این کار از نمودار فیزیکی جریان داده‌ها استفاده می‌شود. این نمودار ابتدا در سال ۱۹۷۹ توسط دو مکروگین مورد استفاده قرار گرفت.

نمادهایی که در ترسیم نمودار استفاده می‌شوند عبارتند از:

۱. **جریان داده‌ها:** حرکت اطلاعات از یک منبع به یک مقصد را نشان می‌دهد که می‌تواند بصورت یک درخواست کتبی و یا یک نامه و یا یک تلفن باشد.

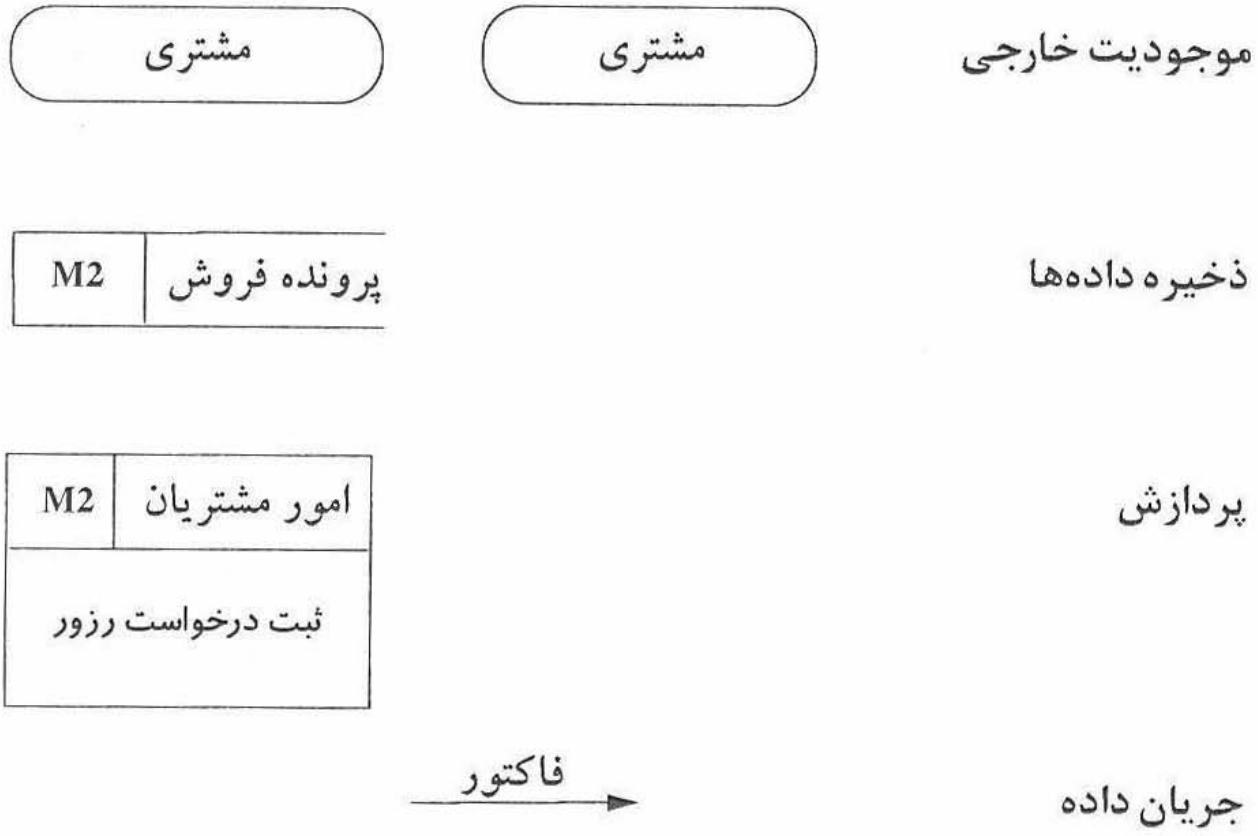
۲. **موجودیت:** عبارت است از اشخاص و یا چیزهایی که خارج از محدودهٔ سیستم واقع می‌شوند و ممکن است کلاً خارج از سازمان مورد بررسی و یا خارج از سیستم مورد مطالعه باشند.

بطور مثال اگر سیستم فروش را تجزیه و تحلیل می‌کنیم. سیستم حسابداری و یا سیستم انبار موجودیتهای خارجی سیستم فروش می‌باشند. موجودیتهای خارجی را با بیضی نشان می‌دهند اگر موجودیت در نمودار تکرار شود، در بالای بیضی خط راست ترسیم می‌گردد.

۳. **محل ذخیره داده‌ها (Data Store):** محلی که اطلاعات پردازش شده در آن ذخیره می‌گردد را نشان می‌دهد که ممکن است. یک پرونده، یک پوشه و یا یک فایل کامپیوتری باشد. محل ذخیره داده‌ها با مستطیل نمایش داده می‌شود.

بر حسب اینکه نوع ذخیره داده‌ها دستی یا مکانیزه باشد، در گوشه آن از حرف M یا D استفاده می‌شود.

۴. **پردازش:** برای نمایش فعالیتی که اطلاعات را دریافت نموده و عملیاتی روی آن انجام می‌دهد و یک خروجی تولید می‌کند از یک مربع استفاده می‌شود که سه قسمت دارد، قسمت بالای سمت چپ شماره گذاری می‌شود، قسمت بالای سمت راست محل انجام پردازش را نشان می‌دهد و در قسمت پائین نام پردازش درج می‌گردد.



شکل ۶.۲ علائم مورد استفاده در ترسیم نمودار فیزیکی جریان داده‌ها

نمودار فیزیکی جریان داده‌ها امکان بسط و گسترش هر پردازش و تشریح آنرا تا پایین‌ترین سطح فراهم می‌آورد. به این ترتیب می‌توان هر پردازش را با جزئیات کامل آن با کاربران در میان گذاشت. معمولاً در ترسیم نمودار فیزیکی جریان داده‌ها از سه سطح پائین‌تر نمی‌روند. پردازش‌های آخرین سطح با استفاده از فرمهایی بنام شرح اولیه پردازش کاملاً تشریح می‌شوند. محصولات نهایی این گام عبارتند از:

- ترسیم نمودار سطح صفر جریان داده‌ها
- ترسیم نمودار جریان داده‌ها برای کلیه سطوح
- شرح اولیه پردازش‌ها
- لیست موجودیتهای خارجی
- شرح فرمهای ورودی و خروجی واقلام اطلاعاتی آنها
- شرح ذخایر داده و اقلام اطلاعاتی آنها

گام ۱۴۰ بررسی و ایجاد نمودار ساختار منطقی داده‌ها

هدف از ایجاد نمودار ساختار منطقی داده‌ها جمع‌آوری اطلاعات و ایجاد مدلی

برای نمایش اطلاعات مورد نظر سیستم و نحوه ارتباط آنها با یکدیگر می‌باشد.

تهیه نمودار ساختار منطقی داده‌ها یکی از کارهای اصلی در SSADM می‌باشد زیرا در مراحل ۱، ۳ و ۶ از آن استفاده می‌شود. این نمودارها ابتدا در سال ۱۹۷۶ معرفی شدند و البته قبل از آن در سال ۱۹۶۹ نمودارهای مشابهی از طرف CODASYL پیشنهاد شده بودند. نظر به موفقیت‌آمیز بودن این تکنیک، SSADM نیز از آن استفاده کرد. برای ترسیم نمودار ساختار منطقی داده‌ها تحلیلگر بایستی با مفاهیم موجودیت و اقلام داده آشنا باشد. لذا به منظور آشنایی بیشتر با مفاهیم فوق در این قسمت به معرفی آن می‌پردازیم.

موجودیت (Entity)

عبارت است از شخصی یا چیزی یا مفهومی که سیستم می‌خواهد درباره آن اطلاعاتی داشته باشد فرض کنید سیستم فروش و یا سیستم حسابداری بخواهد اطلاعاتی را راجع به مشتری، محصول یا سفارش داشته باشد در این صورت موجودیتهای مشتری، محصول و سفارش در این سیستم مطرح خواهد بود.

اقلام اطلاعاتی (Attribute)

هر موجودیت شامل مجموعه‌ای از اقلام اطلاعاتی مرتبط بهم است. به عبارت دیگر، هر موجودیت توسط مجموعه‌ای از داده‌ها، شناسائی و توصیف می‌شود. بنابراین اگر موجودیت محصول را در نظر بگیریم اقلام اطلاعاتی (Attribute) آن می‌تواند شامل کد محصول، شرح (نام محصول)، رنگ و چیزهایی از این قبیل باشد. شناسایی اقلام اطلاعاتی مهم است اما در این مرحله نباید کار اصلی فراموش گردد و به دنبال اقلام اطلاعاتی موجودیتها باشیم. نکته دیگر آنکه اقلام اطلاعاتی بایستی ساده (Atomic) باشند یعنی صرفاً یک مقدار داشته باشند.

بعد از شناسایی اقلام اطلاعاتی به شناسایی رابطه (Relationship) بین موجودیتها می‌پردازیم که از اهمیت ویژه‌ای نیز برخوردار است.

رابطه بین موجودیتها

ارتباط بین دو موجودیت، به یکی از صورتهای ذیل می‌تواند باشد:

۱. ارتباط یک به یک

۲. ارتباط یک به چند

۳. ارتباط چند به چند

در SSADM معمولاً رابطه بین دو موجودیت از نوع یک به چند است، به این معنا که هر نمونه (occurance) از موجودیت اول می‌تواند با یک یا چند نمونه از موجودیت دوم ارتباط داشته باشد. به طور مثال رابطه بین دو موجودیت پزشک و بیمار را در نظر بگیرید هر پزشک می‌تواند چندین بیمار داشته باشد.



۶.۳ ارتباط بین دو موجودیت

رابطبین دو موجودیت می‌تواند با خط‌چین و خط پر نمایش داده شود خط پُر نشان می‌دهد که هر نمونه (occurance) از موجودیت اول حتماً با تعدادی (بزرگ‌تر از صفر) از نمونه‌های موجودیت دوم در ارتباط می‌باشد ولی خط‌چین نشان می‌دهد که این تعداد صفر هم می‌تواند باشد به مثال‌های زیر توجه کنید.

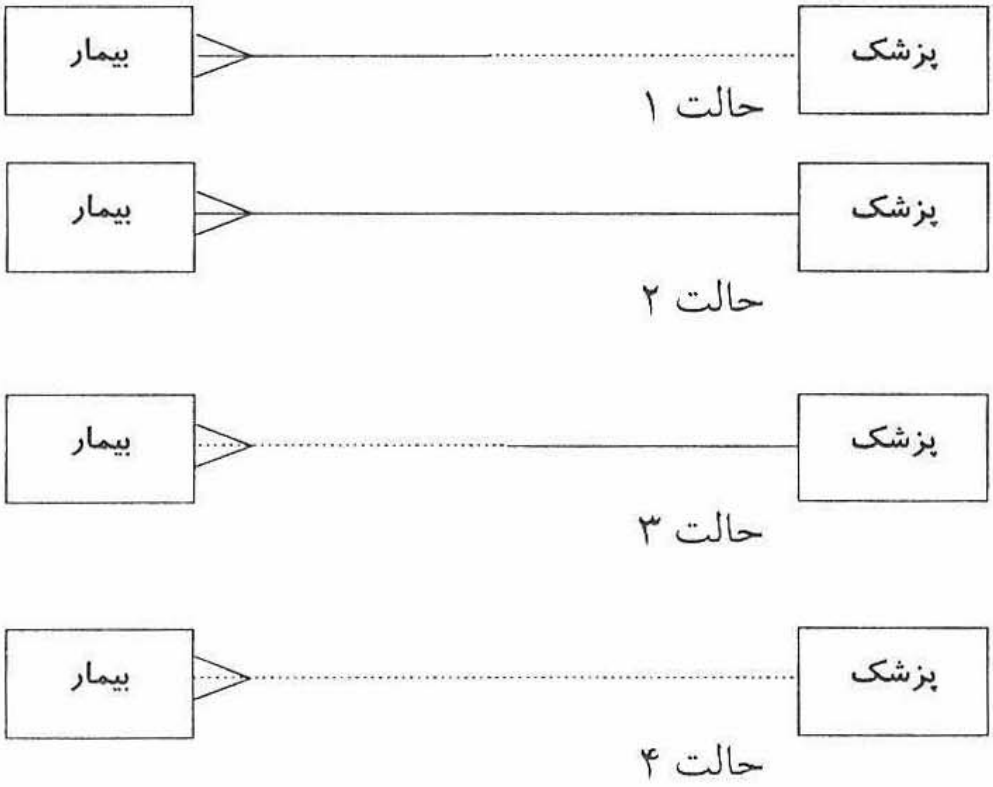
۱. یک پزشک می‌تواند صفر یا چند مریض داشته باشد و هر مریض نیز توسط یک پزشک ویزیت می‌شود. (در این صورت پزشکی وجود دارد که بیماری نداشته باشد ـ پزشک تازه‌کار)

۲. یک پزشک می‌تواند یک یا چند بیمار داشته باشد و هر بیمار توسط یک پزشک ویزیت شود.

۳. یک پزشک می‌تواند یک یا چند بیمار داشته باشد ولی هر بیمار می‌تواند توسط یک پزشک ویزیت شود یا هیچ پزشکی بیمار را ویزیت نکند. (حالتی که هیچ پزشکی راضی نشود تا بیمار را ویزیت کند و یا هنوز هیچ پزشکی به بیمار اختصاص داده نشده باشد)

۴. یک پزشک می‌تواند صفر یا چند بیمار داشته باشد و یک بیمار می‌تواند توسط صفر یا یک پزشک ویزیت گردد.

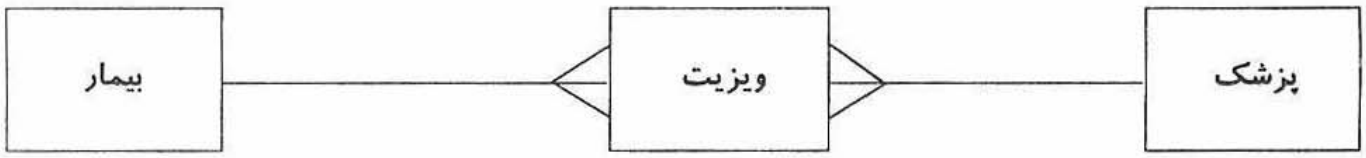
نوع رابطه دو موجودیت بستگی به سیستم مورد مطالعه دارد و می‌تواند شامل هر یک از موارد فوق باشد.



۶.۴ نحوه نشان دادن ارتباط بین دو موجودیت (در حالت‌های مختلف ارتباط دو موجودیت)

رابطه چند به چند

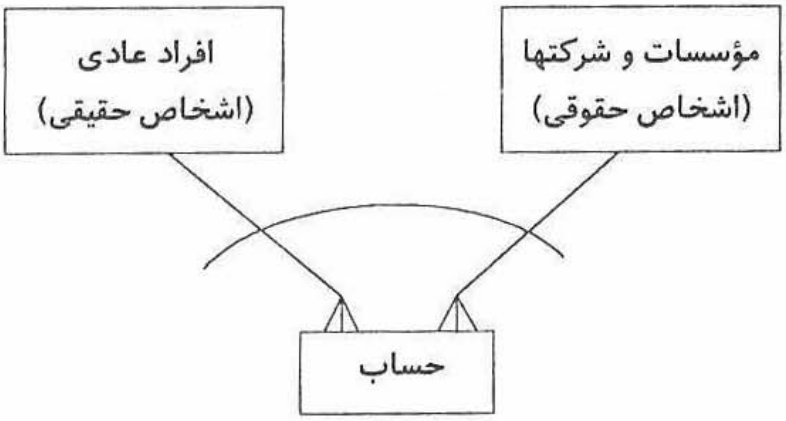
فرض کنید بیمارستانی را بررسی می‌کنیم که هر بیمار را چندین پزشک متخصص ویزیت نماید و هر پزشک چندین بیمار داشته باشد. معمولاً رابطه چند به چند را در نمودار منطقی ساختار داده‌ها نمایش نمی‌دهند و آن را به دو رابطه یک به چند مطابق شکل ۶ـ۵ تبدیل می‌کنند. یکی از مشکلات یک رابطه چند به چند تک مقداری نبودن اقلام اطلاعاتی می‌باشد چرا که در موجودیت بیمار بایستی اطلاعات چندین پزشک نگهداری گردد و همین طور در موجودیت پزشک بایستی اطلاعات چندین بیمار ثبت شود به همین دلیل رابطه چند به چند را تبدیل به دو رابطه یک به چند می‌نمایند و از یک موجودیت واسط استفاده می‌کنند. در شکل زیر موجودیت ویزیت چنین نقشی را دارد.



شکل ۶.۵ تبدیل یک رابطه چند به چند به دو رابطه یک به چند

فرض کنید که می‌خواهیم رابطه بین موجودیت‌های مشتری و حساب بانک را نمایش دهیم. حساب بانکی می‌تواند توسط افراد عادی (اشخاص حقیقی) و یا توسط مؤسسات و شرکت‌ها (اشخاص حقوقی) افتتاح شود ولی فقط یکی از این دو موجودیت با یک حساب رابطه دارند.

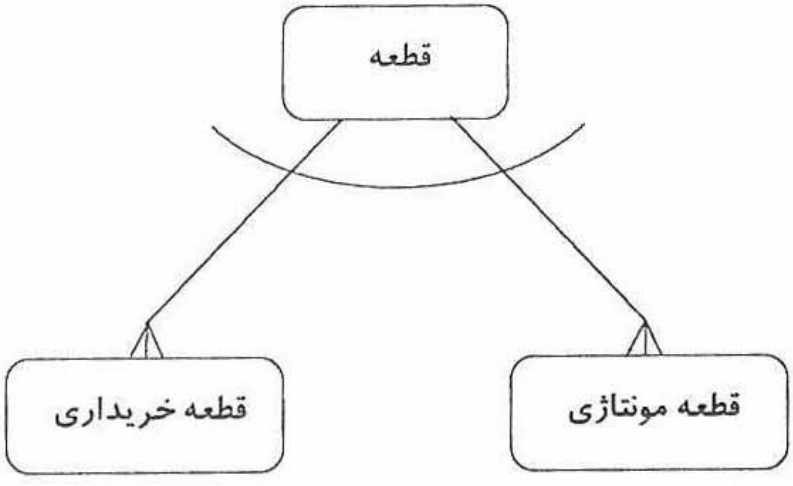
پس حالتی را که یک موجودیت بسته به مورد به یکی از دو موجودیت بالاسری (Master) ارتباط دارد به صورت شکل ۶ـ۶ نشان می‌دهیم.



۶.۶ ارتباط یک موجودیت به یکی از دو موجودیت بالاسری بسته به مورد

زیرموجودیت

مفهوم Subtype یا زیرموجودیت اولین بار توسط SSADM نسخه ۴ پیشنهاد شد. فرض کنید دو موجودیت در بسیاری از موارد اطلاعات مشترکی داشته باشند و تنها برخی از اقلام اطلاعاتی آنها با یکدیگر متفاوت باشد به طور مثال موجودیت قطعه را در نظر بگیرید این موجودیت نشان‌دهندهٔ اطلاعات قطعات است این قطعات می‌تواند قطعات خریداری شده توسط کارخانه و یا قطعات مونتاژ شده باشند که برخی از اقلام اطلاعاتی آنها با یکدیگر متفاوت است شکل زیر این وضعیت را نشان می‌دهد.



مدل منطقی داده‌ها

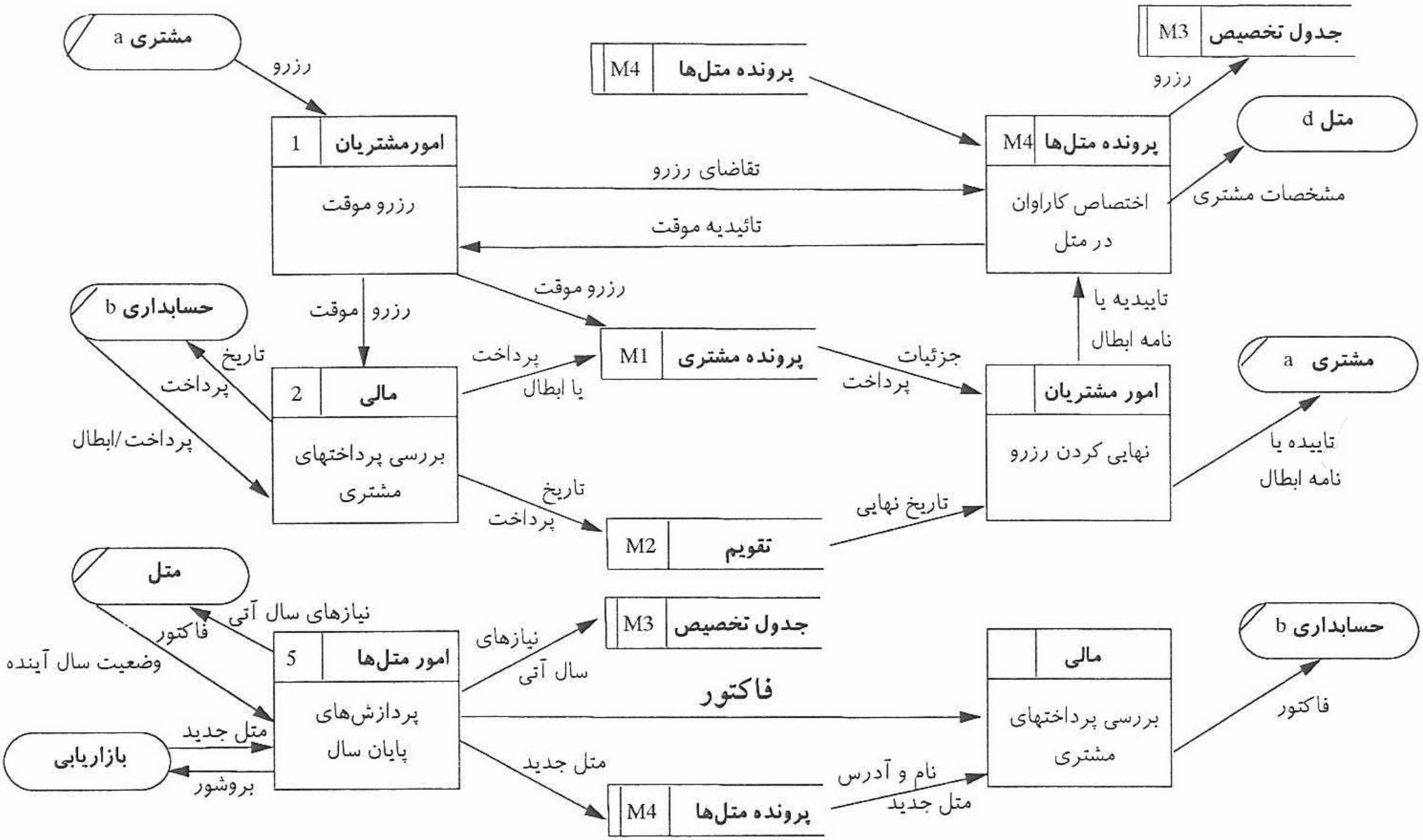
یکی از محصولات این گام تهیه مدل منطقی داده‌ها است که شامل مستندات زیر می‌باشد.

- نمودار ساختار منطقی داده‌ها
- فهرست موجودیت‌ها
- شرح اقلام اطلاعاتی هر موجودیت
- تعیین دامنه اقلام اطلاعاتی، حوزه مقادیری که یک قلم اطلاعاتی می‌تواند اختیار کند.

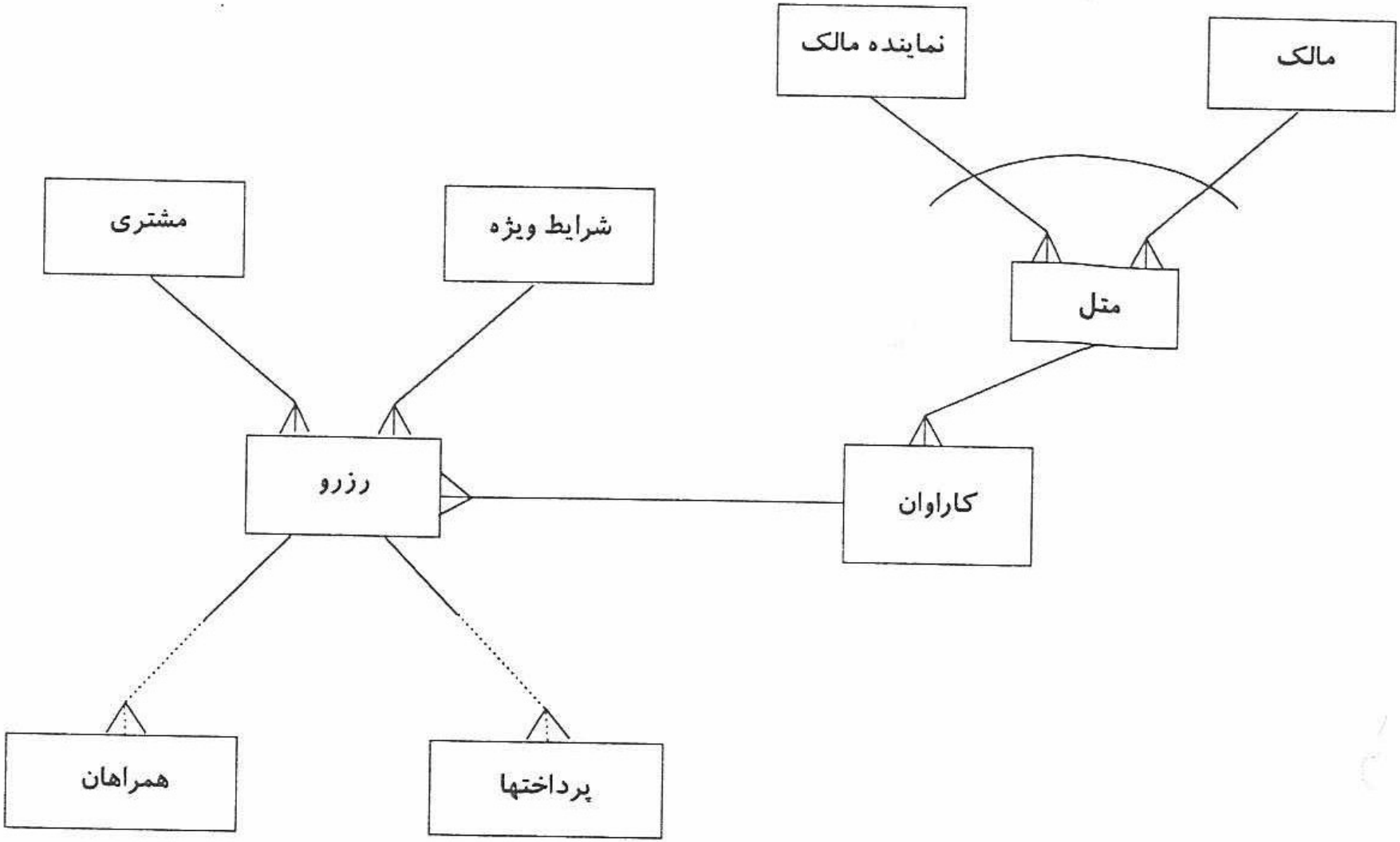
ترسیم نمودارهای سیستم رزرواسیون اجاره متل (یک مثال)

فرض کنید بنگاهی یک سری کاراوان را در متل‌هایی که متعلق به بنگاه و یا سایرین می‌باشد اجاره می‌دهد. و می‌خواهیم سیستم رزواسیون آن را مکانیزه نماییم. (همان طور که قبلاً بیان شد) حال می‌خواهیم اطلاعات قسمت رزرو را کامل‌تر کنیم. رزرو کاروان ابتدا توسط قسمت مشتریان صورت می‌گیرد و جزئیات رزرو به قسمت متل‌ها منتقل می‌شود تا از آزاد بودن کاراوان در متل مربوطه در زمان مورد درخواست مشتری اطمینان حاصل شود. اگر کاراوان آزاد نباشد تقاضای مشتری رد شده و به اطلاع وی می‌رسد. معمولاً کاراوان خالی است بنابراین به مشتری اطلاع داده می‌شود که برای او کاراوان مورد نظر موقتاً رزرو شده است، تا پرداخت‌ها توسط مشتری صورت گیرد و در پرونده وی ثبت گردد این تاریخ در یک تقویم در قسمت مالی ثبت می‌شود، قسمت مالی تاریخ پرداخت را برای قسمت حسابداری در سال می‌دارد وقتی پرداختی صورت نمی‌گیرد و یا مشتری رزرو را باطل (کنسل) می‌کند حسابداری قسمت مالی را مطلع می‌کند تا در پرونده مشتری ثبت گردد و رزرو موقت حذف شود.

امور مشتریان مسؤل رسیدگی به پرداخت‌های مشتریان است در صورتی که در تاریخ مقرر پرداخت صورت گیرد تأییدیه رزرو نهایی برای مشتری فرستاده می‌شود نام مشتری به همراه نام همراهان به متل مربوطه ارسال می‌گردد. در غیر این صورت نامه ابطال برای مشتری فرستاده می‌شود. در پایان هر سال هر متل یک فاکتور برای بنگاه می‌فرستد و سپس امور متل‌ها آن را به قسمت مالی ارسال می‌دارد تا با اطلاعات امور مالی برای پرداخت‌ها مقایسه صورت گیرد. متل بنگاه را از برنامه سال آینده خود آگاه می‌نماید و قسمت بازاریابی مشخصات محل‌های جدید را به امور مشتریان در سال می‌دارد تا بروشور سال آینده را به قسمت بازاریابی تحویل دهد.



۶۸ نمودار فیزیکی جریان داده‌ها (سیستم رزرواسیون متل)



۶.۹ نمودار ساختار منطقی داده‌ها برای سیستم رزرو متل

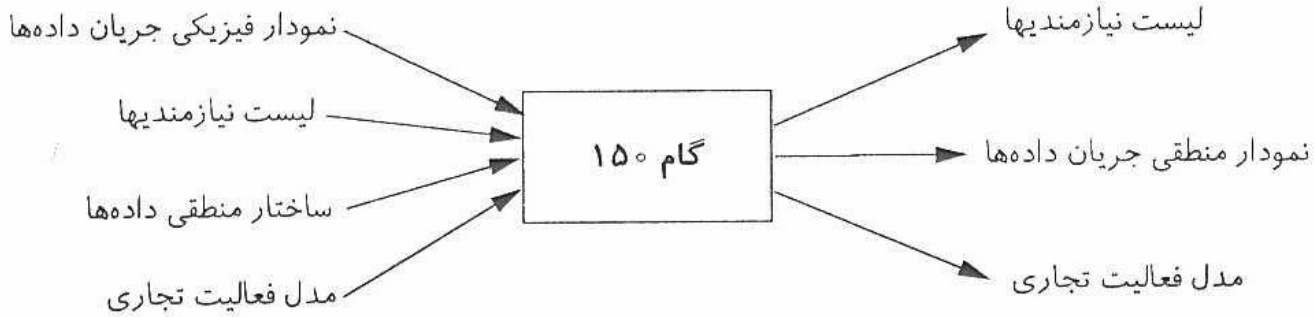
فصل هفتم

ارائه دیدگاه منطقی

از وضعیت موجود

اهداف فصل

هدف از این فصل شناسایی مشکلات و نارسایی‌های وضعیت موجود می‌باشد. این مشکلات ممکن است بر اثر محدودیت‌های جغرافیایی و یا قانونی یا بر اثر وجود رویه‌های دوگانه و انجام کارهای موازی ایجاد شده باشد. در این بخش این محدودیت‌ها شناسایی و رویه‌های چندگانه مشخص می‌شود، بعلاوه با منطقی‌سازی نمودار فیزیکی جریان داده‌ها، ذخایر داده به نحوی تغییر می‌کند که هر موجودیت تنها در یک محل ذخیره داده وجود داشته باشد و نمودار فیزیکی جریان داده‌ها و ساختار منطقی داده‌ها با یکدیگر مطابقت پیدا می‌کنند. نمودار فیزیکی جریان داده‌ها و ساختار منطقی داده‌ها دو ورودی اصلی به این گام می‌باشند.



۷.۱ ورودی‌ها و خروجی‌های گام ۱۵۰

مراحل مختلف منطقی‌سازی

برای منطقی‌سازی جریان داده‌ها می‌توان از دو روش بالا به پایین (Top-Down) و پایین به بالا (Bottom-Up) استفاده کرد. روش اول سریع‌تر به نتیجه می‌رسد ولی ممکن است بعضی از نیازها نادیده گرفته شود ولی روش دوم زمان‌بر است. با توجه به این که در ابتدا با استفاده از تکنیک تجزیه و تحلیل عملکردها به شناخت وظایف سیستم پرداخته‌ایم و از آن استفاده کرده پس روش بالا به پایین را انتخاب می‌کنیم. مراحل مختلف این گام به شرح زیر است.

۱. بررسی مدل فعالیت تجاری موجود
۲. تعریف محل ذخیره داده‌ها با توجه به نمودار منطقی داده‌ها
۳. ایجاد نمودار منطقی جریان داده‌ها
۴. مطابقت نمودار منطقی با نمودار فیزیکی جریان داده‌ها

۵. تأیید کاربران

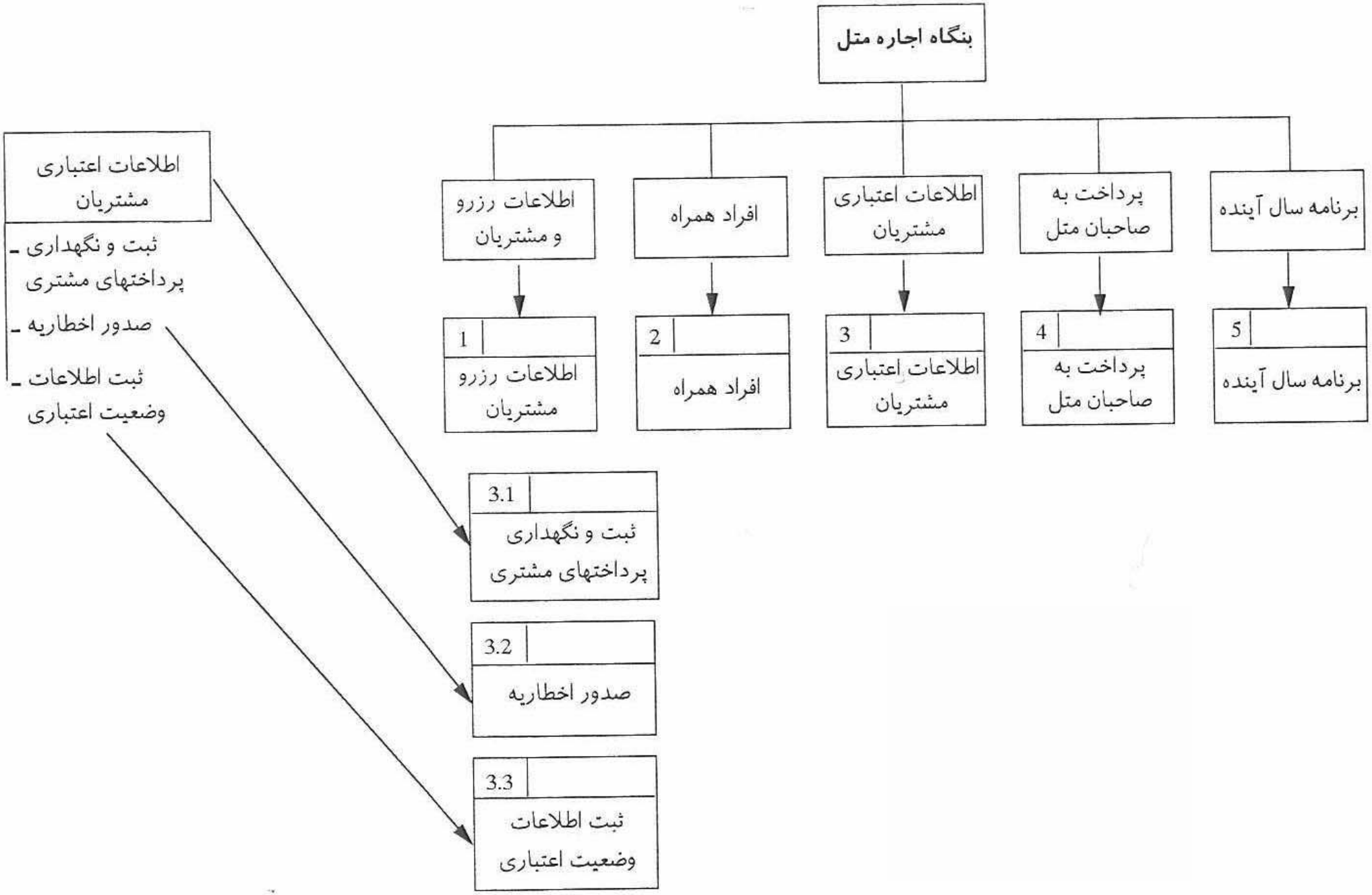
پس از انجام مراحل فوق آخرین گام مرحله ۱ پایان می‌یابد و مرحله شناخت و بررسی وضعیت موجود به اتمام می‌رسد.

بررسی مجدد مدل فعالیت تجاری

مدل فعالیت تجاری در گام ۱۱۵ به منظور بررسی عملیات و فعالیت‌های سیستم ایجاد شده است. بدین ترتیب که فعالیت‌ها حداکثر به هفت یا هشت بخش تقسیم می‌شود و هر بخش خود شامل تعدادی زیربخش است که نتیجه آن در نمودار تجزیه و تحلیل عملکردها نشان داده می‌شود. ممکن است پس از بررسی مجدد وضعیت موجود، لازم باشد سطوح پایین آن تغییر کند. بنابراین بایستی به این مطلب توجه کنیم که چه پردازش‌هایی در این سطوح قرار می‌گیرد.

معمولاً در این سطح پردازش‌هایی که باعث تغییر اطلاعات شوند، قرار می‌گیرند و پردازش‌هایی که منحصر به دستیابی به اطلاعات می‌شوند در نظر گرفته نمی‌شوند پردازش‌هایی که عمدتاً شامل گزارشات هستند از این دسته محسوب می‌شوند ولی این امر کلیت نداشته و بستگی به سیستم مورد مطالعه دارد. به طور مثال در یک سیستم حقوق و دستمزد صدور فیش حقوقی اگر چه جنبه گزارش‌گری دارد ولی یکی از پردازش‌های اصلی سیستم می‌باشد و حذف آن صحیح نیست. در این قسمت بیشتر گزارشات موردی مدنظر است.

برای تبدیل نمودن نمودار تجزیه و تحلیل عملکردها به نمودار منطقی جریان داده‌ها باید در نظر داشته باشیم که فعالیت‌های سطح اول، پردازش‌های سطح اول نمودار منطقی را نمایش داده و فعالیت‌های سطح دوم پردازش‌های سطح دوم نمودار را نشان می‌دهد و همین طور الی آخر. در نمودار منطقی جریان داده‌ها محل فیزیکی انجام پردازش‌ها نشان داده نمی‌شود. شکل ۲ – ۷ چگونگی این تبدیل را نشان می‌دهد.



۷.۲ تبدیل مدل فعالیت تجاری به پردازش‌ها

محل ذخیره داده

در نمودار فیزیکی جریان داده‌ها محل ذخایر داده نمایانگر محل موقت نگهداری اطلاعات از جمله فایل‌ها، پرونده‌ها، پوشه‌ها و غیره می‌باشد که اطلاعات در آنها می‌تواند به صورت تکراری نگهداری شود، و معمولاً با یکدیگر ناسازگاری دارند و یا می‌تواند به تعداد زیادی موجودیت اشاره کند و اطلاعات زیادی راجع به آنها را در بر داشته باشد. به طور مثال پرونده مشتری می‌تواند اطلاعات سفارشات و محصولات و اطلاعات دیگری راجع به مشتری را در بر گیرد اما محل ذخیره داده‌ها در نمودار منطقی جریان داده‌ها به علت ارتباط آن با موجودیت‌ها این مسأله را حل کرده است. در نمودار منطقی جریان داده‌ها نمی‌توان هر موجودیت را با یک محل ذخیره داده نشان داد و یک رابطه یک به یک برقرار کرد زیرا که بسیاری از آنها برای کاربران قابل‌توجه نیستند مانند موجودیت رابط بین دو موجودیت دیگر. پس بایستی یک گروه از موجودیت‌ها را با اسامی قابل‌فهم برای کاربر به صورت محل ذخیره داده نمایش داد و اگر بتوانیم موجودیت‌های کنار هم را توسط یک محل ذخیره داده واحد نمایش دهیم در ترسیم نمودار منطقی جریان داده‌ها کمک زیادی خواهد شد زیرا خطوط جریان داده را کاهش می‌دهد و گروه‌بندی موضوعی موجودیتها نیز در کاهش خطوط جریان داده‌ها مؤثر است. گروه‌بندی موضوعی موجودیت‌ها را می‌توان در فرمی به صورت شکل ۴-۷ نمایش داد.

ترسیم نمودار منطقی جریان داده‌ها

پس از تعیین محل ذخایر داده‌ها بایستی سری کامل نمودار منطقی جریان داده‌ها تهیه شود. قبل از تهیه نمودار باید نکات زیر را روشن نمود تا در ترسیم نمودار با مشکلی مواجه نشویم.

۱. ورودی‌ها و خروجی‌های هر پردازش مشخص شود.

۲. مشخص گردد اطلاعات مورد نیاز پردازش از محل ذخایر داده‌ها و موجودیت‌های داخلی تأمین می‌شود یا از موجودیت‌های خارجی

۳. اگر پردازشی باعث ذخیره اطلاعات می‌شود، در کدام موجودیت و کدام محل ذخیره داده آن را ثبت می‌کند؟

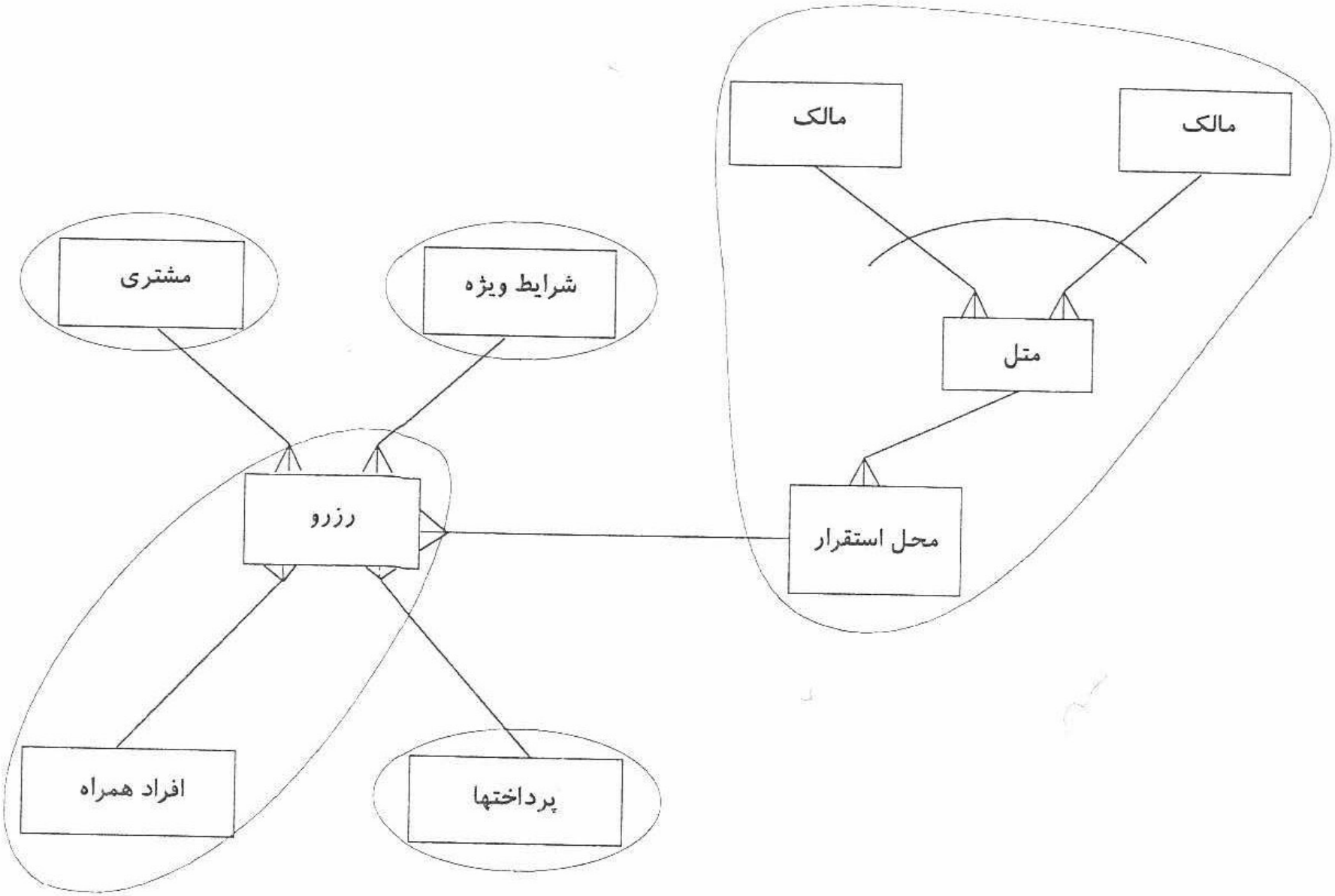
۴. آیا پردازش، بهنگام رسانی هم انجام می‌دهد، اگر بلی کدام محل ذخیره داده و

موجودیت را؟

۵. خروجی پردازش در کدام موجودیت نگهداری می‌شود و یا به کدام موجودیت خارجی ارسال می‌گردد؟

پاسخ به سؤالات بالا مشخص می‌کند که هر پردازش با چه موجودیت‌هایی ارتباط دارد.

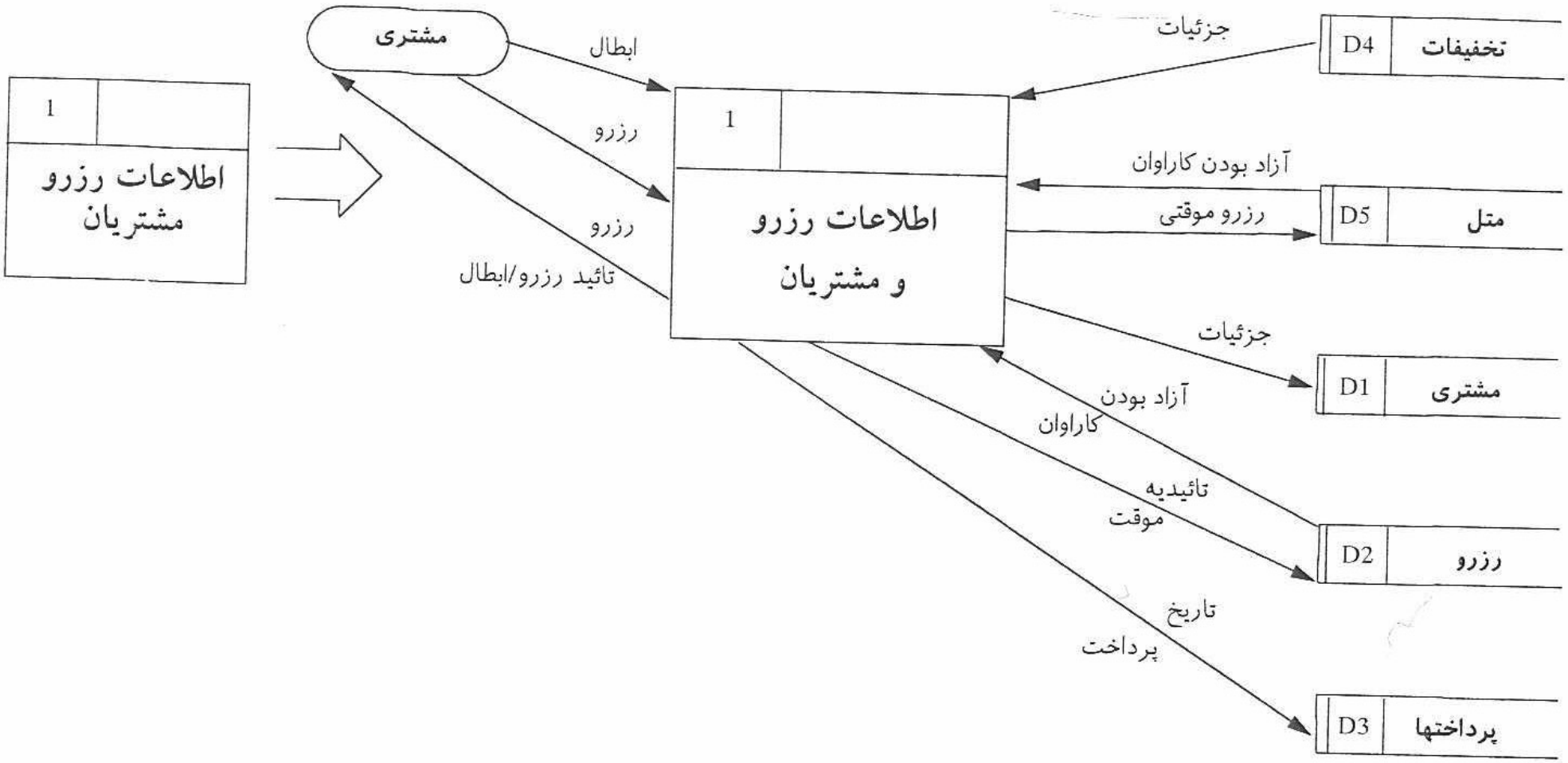
هر سطح نمودار تجزیه و تحلیل عملکردها مطابق سطح متناظر نمودار منطقی جریان داده‌ها است. شکل ۷.۵ نمودار منطقی سطح ۱ جریان داده‌ها را برای بنگاه اجاره متل نشان می‌دهد که مطابق سطح ۱ نمودار تجزیه و تحلیل عملکرد بنگاه مزبور می‌باشد.



۷.۳ گروه‌بندی ساختار منطقی داده‌ها (بنگاه اجاره متل)

فرم ارتباط ذخیره داده/موجودیت		
کد ذخیره داده	نام ذخیره داده	نام موجودیت
D1	مشتری	مشتری
D2	رزرو	رزرو افراد همراه
D3	پرداختها	پرداختها
D4	تخفیف	شرایط ویژه
D5	متل	متل کاروان مالک نماینده

۷.۴ فرم ارتباط ذخیره داده / موجودیت



۷.۵ چگونگی تبدیل به مدل منطقی

قواعد ترسیم نمودار منطقی جریان داده‌ها

برای ترسیم نمودار منطقی جریان داده‌ها نکات زیر باید رعایت شود.

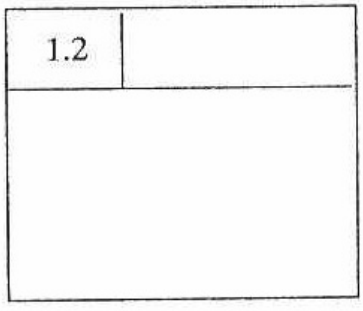
۱. به لحاظ این که به محل فیزیکی و نام اشخاص انجام دهنده پردازش نیازی نیست، کلیه نام اشخاص و محل فیزیکی انجام پردازش‌ها از نمودار حذف گردد.

۲. در مواقعی که شخص خاصی بایستی پردازش را انجام دهد به طور مثال «تعیین افرادی که بایستی ترفیع بگیرند توسط مدیر پرسنلی» بایستی به ترتیب شکل ۷.۷ آن را نمایش داد تا مشخص گردد که این فعالیت نمی‌تواند مکانیزه شود، زیرا مدیر پرسنلی احتیاج به ثبت و بازبینی اطلاعات دارد، دو پردازش مزبور اضافه گردیده است.

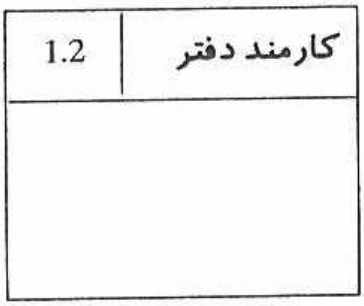
۳. اگر پردازشی بین چندین نفر شکسته شده است لزومی به نمایش کلیه این تقسیمات نیست، و آن را بایستی در قالب یک پردازش نمایش داد.

۴. بعد از هر رویداد بایستی یک پردازش انجام شود و نه چندین پردازش پشت سر هم.

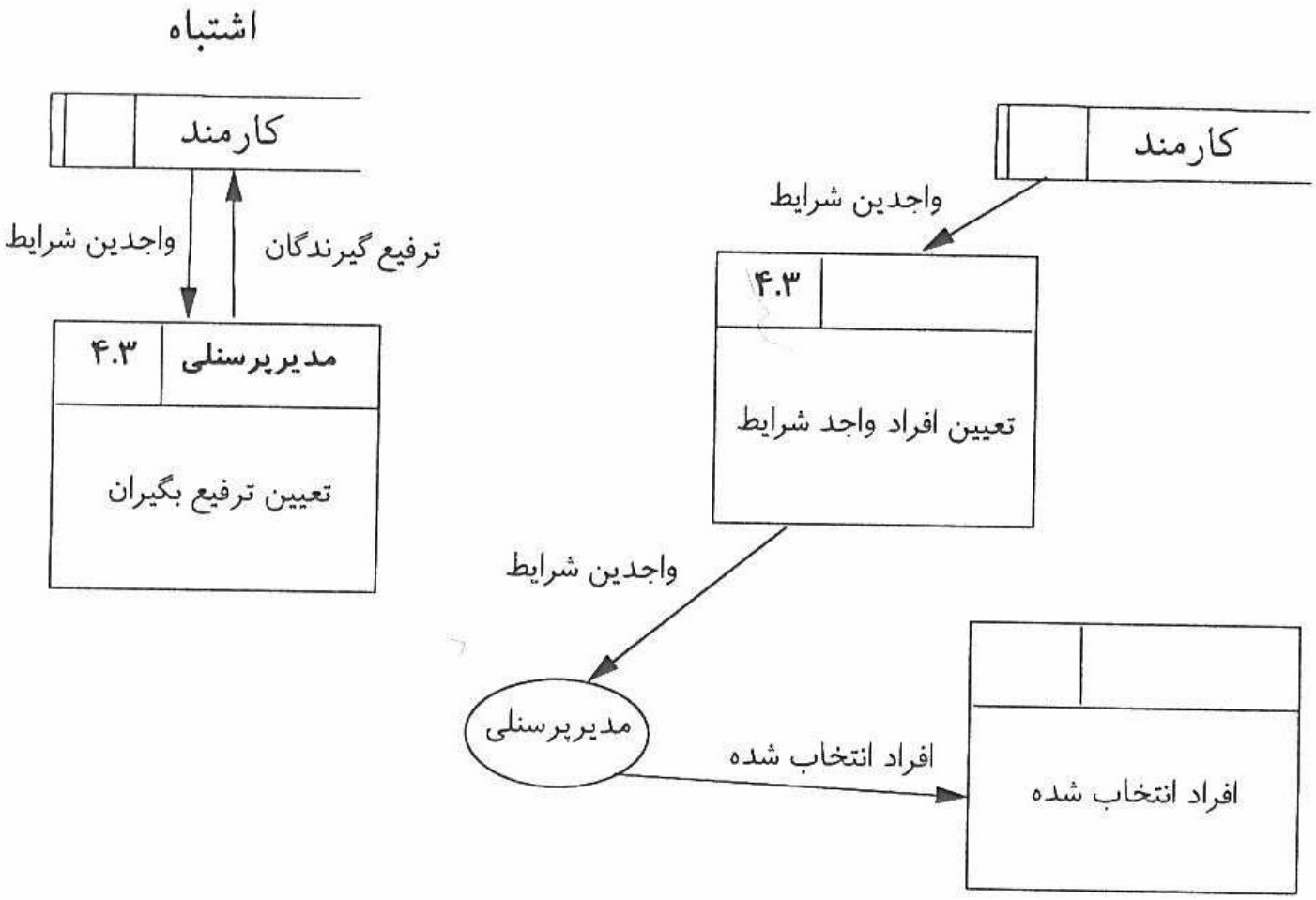
درست



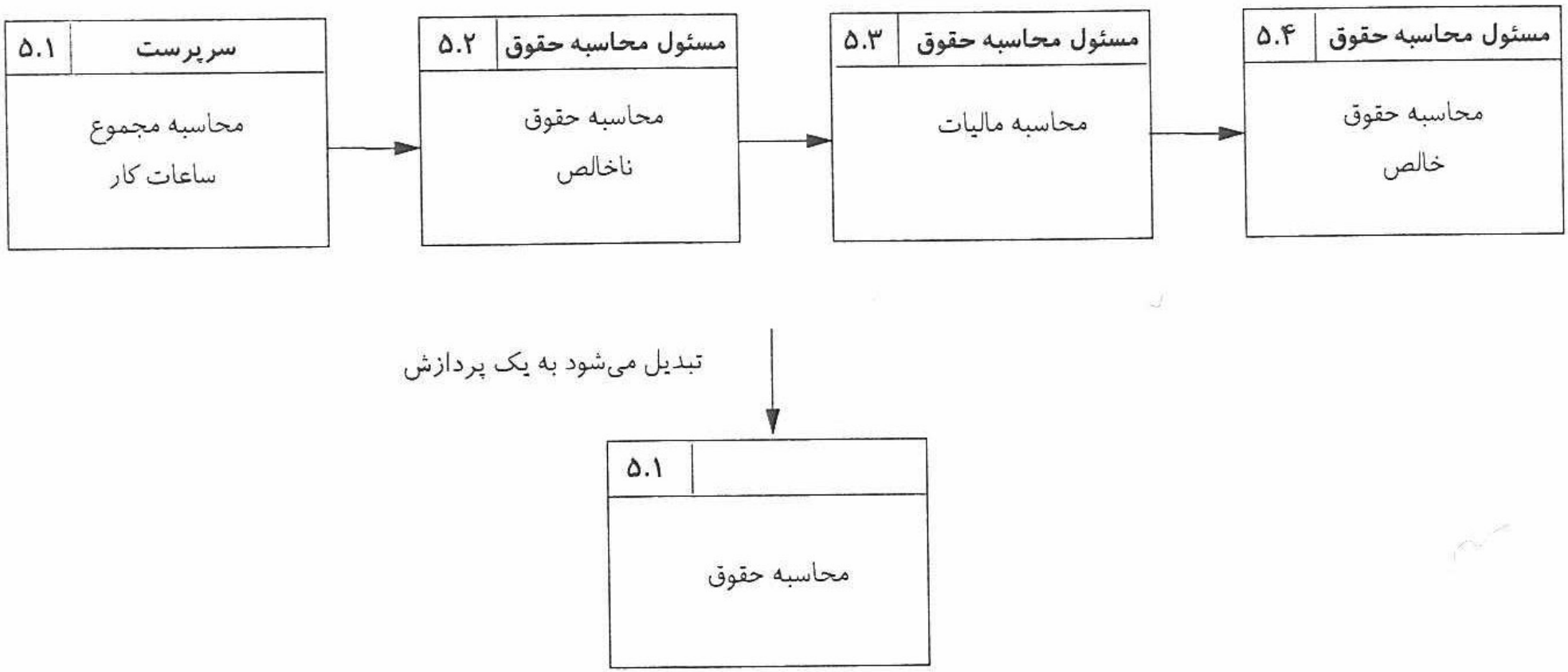
اشتباه



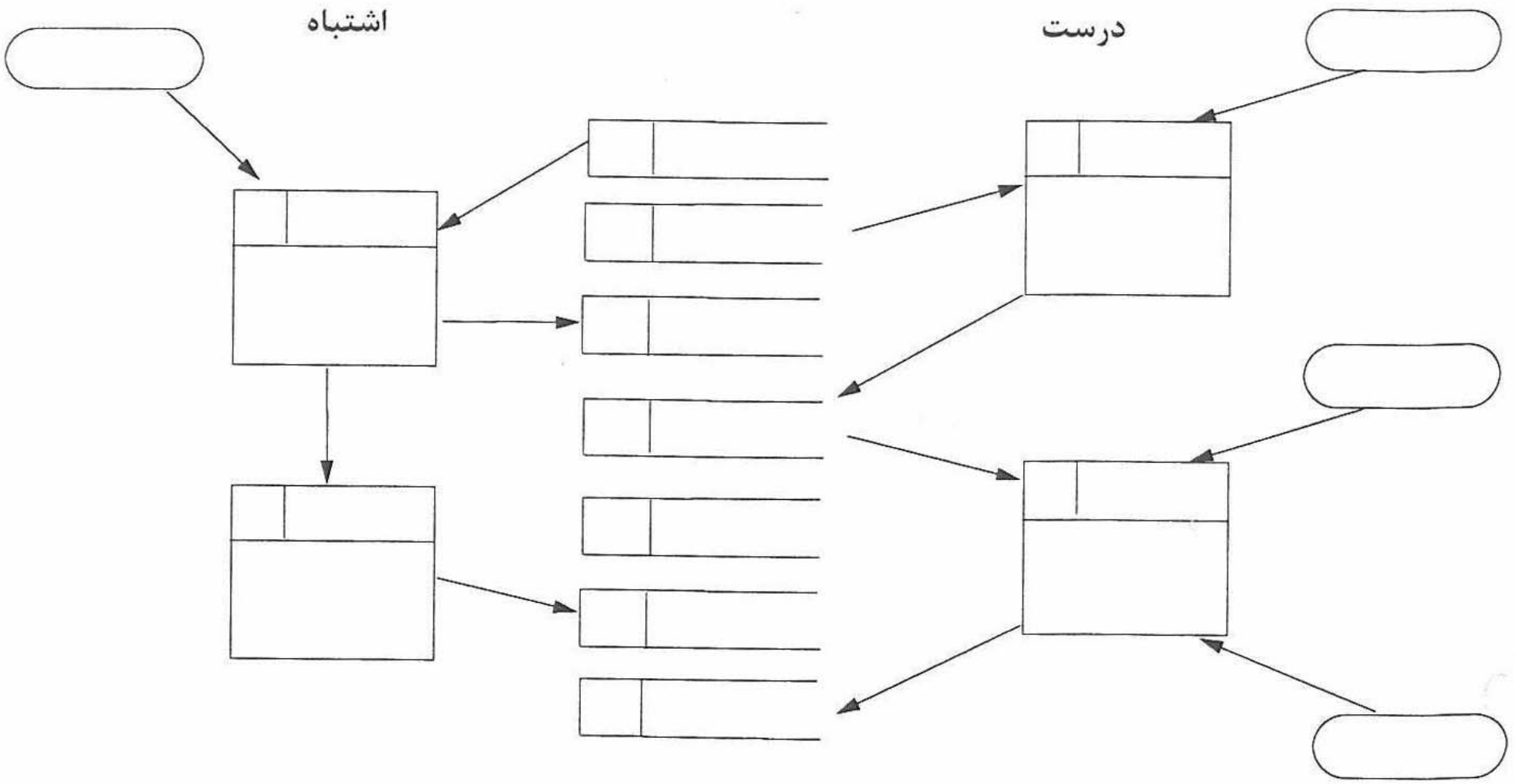
شکل ۷.۶ حذف مکان فیزیکی و اشخاص انجام‌دهنده پردازش



شکل ۷.۷ چگونگی نشان دادن انتخاب ترفیع بگیران توسط مدیر پرسنلی



شکل ۷.۸ نمایش ترتیب پردازش‌ها



شکل ۷.۹ براثر هر رویداد یک پردازش باید فعال گردد

محل ذخایر داده انتقالی

این نوع محل ذخایر داده از نوع موقتی هستند یعنی در آنها اطلاعات به طور موقت ذخیره می‌شود سپس به پرونده‌های اصلی منتقل می‌گردد. در ترسیم نمودار منطقی جریان داده‌ها این نوع ذخایر داده‌ها نبایستی در نظر گرفته شود. زیرا این ذخایر برای رفع محدودیت‌های موقت ایجاد شده‌اند.

ارزیابی مدل منطقی داده‌ها

آخرین کار در این گام ارزیابی مدل ایجاد شده، بررسی و رفع نقایص احتمالی است. برای این کار ابتدا نمودار منطقی جریان داده‌ها را با نمودار فیزیکی جریان داده‌ها مطابقت داده و پردازش‌هایی که از قلم افتاده‌اند مشخص می‌شود. ورودی‌ها و خروجی‌های هر پردازش از نمودار فیزیکی را با ورودی‌ها و خروجی‌های پردازش مشابه از نمودار منطقی مطابقت می‌دهیم. سپس به بررسی محل ذخایر داده فیزیکی می‌پردازیم و از مطابقت آن با محل دخایر داده منطقی و در نظر گرفتن اقلام اطلاعاتی آن اطمینان حاصل می‌کنیم و پس از انجام مراحل فوق تأییدیه کاربران اخذ می‌شود، تا از نقطه نظرات آنان آگاه شویم. نمودار فیزیکی ابزار مناسبی جهت ارتباط با کاربران است ولی نمودار منطقی این حالت را ندارد زیرا در اولی کاربران خود و نقش خود را می‌بینند و در دومی کاربران با پردازش‌های مجرد روبرو هستند.

ممکن است گفته شود که این نمودار بیشتر نحوهٔ کار را از نگاه مدیریت نشان می‌دهد ولی در عمل مدیران وقت زیادی برای بررسی نمودار منطقی ندارند و بهتر است از نمودار تجزیه و تحلیل عملکردها برای ارتباط با مدیران استفاده شود.

مدل منطقی داده‌ها

مهمترین خروجی این گام مدل منطقی داده‌ها است که اجزاء زیر تشکیل شده است:

۱. نمودار منطقی جریان داده‌ها در تمامی سطوح
۲. شرح پردازش‌ها که چیزی جزء شرح پردازش‌های آخرین سطح نیست.
۳. شرح ورودی‌ها و خروجی‌ها
۴. فهرست موجودیت‌ها
۵. فرم ارتباط ذخیره داده / موجودیت

فصل هشتم

گزینه‌های موجود و راه‌حل انتخابی

(گام ۲۱۰ و ۲۲۰)

اهداف فصل

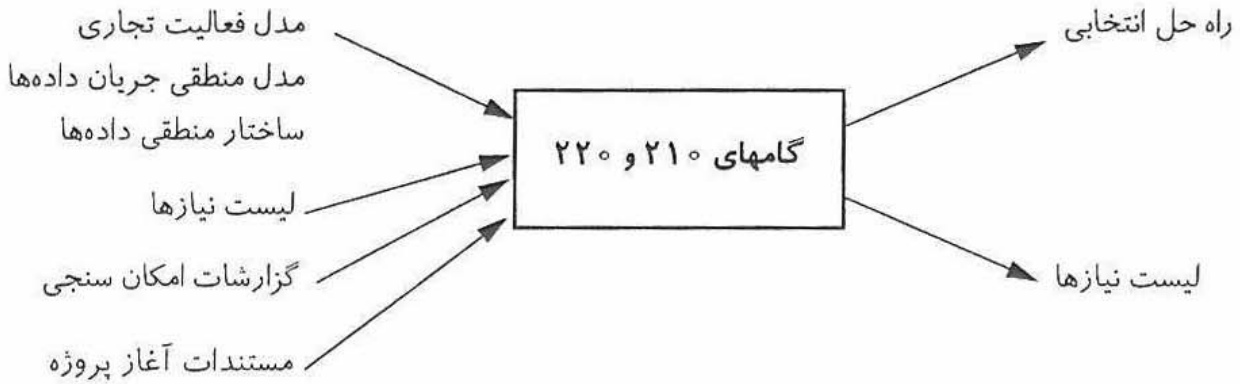
در این فصل گامهای مرحله دوم از مراحل SSADM تشریح می‌گردد.

در این مرحله حالت‌های مختلف و ممکن سیستم جدید به کاربران ارائه می‌شود تا کاربران از بین آنها یکی را جهت پیاده‌سازی انتخاب کنند.

مستنداتی که براساس آن راه‌حل پیشنهادی ارائه می‌گردد عبارت است از:

- نمودار تجزیه و تحلیل عملکردها، نمودار ساختار منطقی داده‌ها و مدل منطقی جریان داده‌ها
- لیست نیازها
- گزارشات امکان‌سنجی
- مستندات آغاز پروژه

همراه با هر یک از راه‌حل‌های پیشنهادی لیست مزایا و برآورد هزینه‌های پیش‌بینی شده نیز به کاربران ارائه می‌شود تا بتوانند راه‌حل مناسب را اختیار کنند.



شکل ۸.۱ ورودی‌ها و خروجی‌های گام‌های ۲۱۰ و ۲۲۰

گام ۲۱۰ گزینه‌های موجود

برای هر راه‌حل پیشنهادی علاوه بر تعیین محدودیت‌ها ورودی‌ها و خروجی‌های مورد نظر مطالب زیر نیز باید مدّنظر قرار گیرد:

- **عملکرد:** کلیه نیازهایی را که سیستم پوشش می‌دهد بایستی مشخص گردد.
- **برآورد هزینه‌ها:** با توجه به راه‌حل پیشنهادی میزان هزینه مربوطه براساس استانداردهای سازمان محاسبه و ارائه گردد. در صورتی که مسائل فنی کاملاً روشن نباشد تهیه این برآورد مشکل خواهد بود.

ـ **زمان‌بندی:** زمان‌بندی اجرای پروژه با توجه به کلیه مراحل پروژه از مرحله طراحی تا پیاده‌سازی، تست و آموزش ارائه شود.

ـ **تأثیر مستقیم:** تأثیر سیستم بر سازمان در نظر گرفته شود از جمله تأثیر سیستم بر نحوهٔ کار کارکنان و نقش آنان در انجام عملیات سیستم و نحوه آموزش‌های لازم برای آنها مورد نظر قرار گیرد.

ـ **مزایای سیستم:** مزایای سیستم پیشنهادی کاملاً تشریح گردد.

این گام فرصتی را فراهم می‌آورد تا پیشنهادات اولیه در گزارش امکان‌سنجی و یا مستندات آغاز پروژه دوباره ارزیابی گردد و اگر عملکرد سیستم با آنچه ارائه شده متفاوت است. مغایرت‌های موجود شناسایی و به کاربران ارائه شود.

در گام ۲۱۰ حداکثر شش راه‌حل ارائه می‌گردد که پس از مشورت با کاربران ۲ یا ۳ وضعیت در نظر گرفته شده و مستندات بالا برای آنها تهیه می‌شود.

گام ۲۲۰ تعیین گزینه

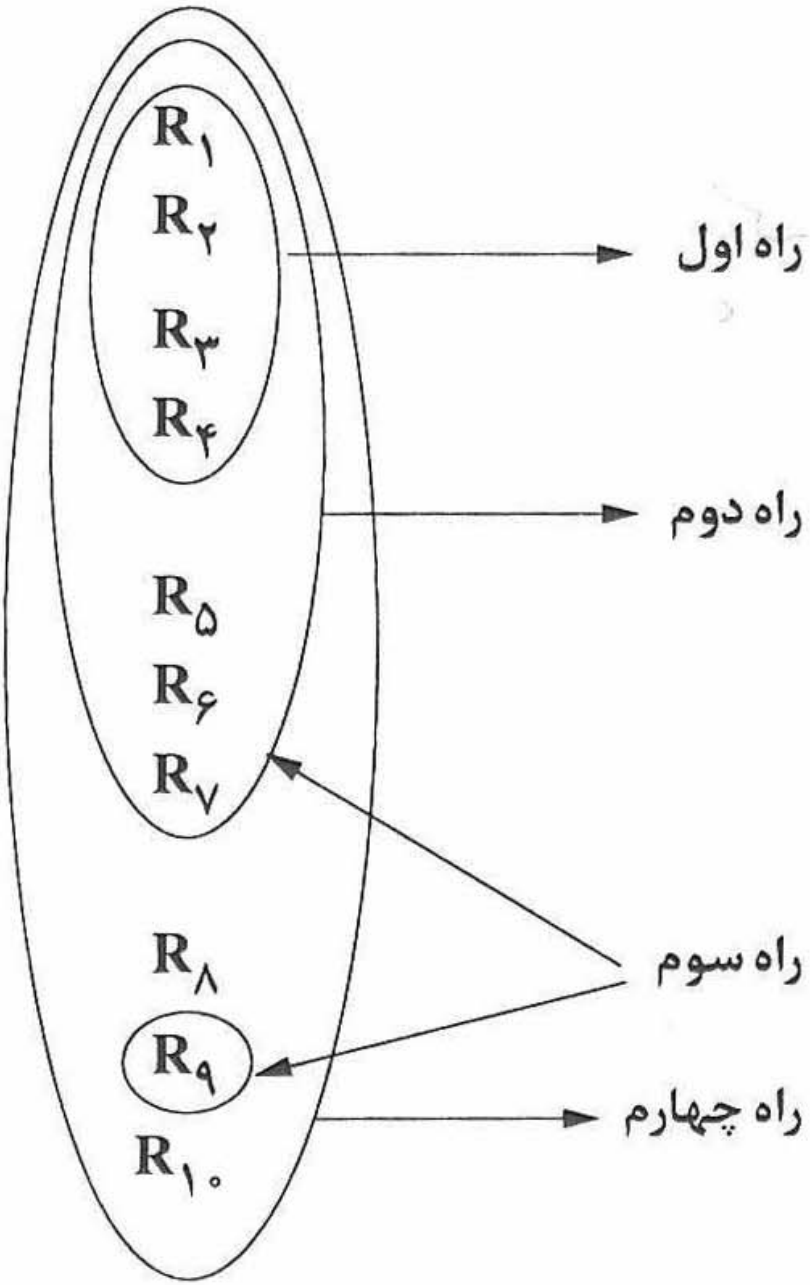
این گام آخرین گام ماجول تجزیه و تحلیل نیازمندی‌ها می‌باشد که شامل نمایش راه‌حل‌های ممکن به کاربران و انتخاب یکی از آنها توسط آنان می‌باشد.

در این گام سعی بر آن است که به کاربران نشان داده شود که سیستم متعلق به آنها می‌باشد و براساس نیازهای آنها توسعه می‌یابد تا بتوانند راه‌حل مناسب را انتخاب کنند.

در صورتی که سیستم کوچک باشد بهتر است که راه‌حل پیشنهادی نوشته شده و در اختیار کاربران قرار گیرد و اگر سیستم بزرگ باشد در صورتی که بتوانیم به تعداد بیشتری از کاربران راه‌حل‌های پیشنهادی را ارائه کنیم و از نظرات آنها بهره بگیریم مفید خواهد بود.

در پایان بایستی این مطلب ذکر شود که حذف این مرحله و نادیده گرفتن آن حتی در موقعی که می‌خواهیم سریعاً یک سیستم را ایجاد کنیم صحیح نیست زیرا این مرحله بسیار تأثیرگذار است.

می‌توان گفت که این گام آخرین گامی است که کاربران به صورت رسمی بر ایجاد سیستم تأثیر می‌گذارند. بنابراین حذف آن درست به نظر نمی‌آید.



شکل ۸.۲ نمایش راه‌حل‌های پیشنهادی و نیازمندی‌های پوشش داده شده

فصل نهم

تعیین پردازش‌ها و ایجاد ساختار منطقی داده‌ها برای سیستم جدید

اهداف مرحله سوم

از این فصل تا فصل هفدهم گامهای مختلف مرحله سوم یا مرحله تعریف و مشخص‌سازی نیازها تشریح می‌گردد. همانطور که قبلاً عنوان شد در این مرحله گامهای ۳۱۰ تا ۳۷۰ موجود می‌باشد و در این مرحله براساس راه‌حل انتخابی و نمودار منطقی و فیزیکی جریان داده‌ها تهیه شده برای وضعیت موجود و ساختار منطقی داده‌ها برای وضعیت موجود، پردازش‌های سیستم جدید و ساختار منطقی سیستم جدید ایجاد شده و سپس به شناسایی هر یک از پردازشها و ورودیها و خروجیهای پرداش با جزئیات کامل می‌پردازیم و با استفاده از ابزارها و روشهایی نظیر نمودار تاریخچه زندگی موجودیت و نمودار مسیر دسترسی و نمودار تطابق اثر اشتباهات و اشکالات مدل منطقی ایجاد شده شناسایی و رفع می‌گردد و مدل مربوطه بهینه می‌گردد.

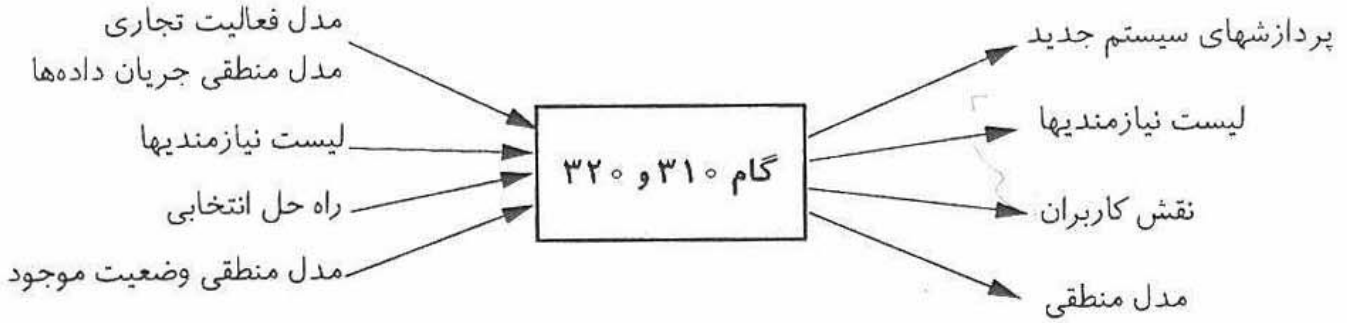
برنامه‌ریزی برای این مرحله از حساسیت زیادی برخوردار است زیرا که باید با توجه به اثر آن در سایر مراحل ساختار و توسعه سیستم صورت گیرد. هر گونه برنامه‌ریزی نامناسب باعث به هدر رفتن زمان و هزینه می‌شود و حتی می‌تواند موجودیت کل پروژه را به خطر اندازد و بعلاوه برنامه‌ریزی این مرحله باید به گونه‌ای باشد تا در پایان آن کلیه مستندات لازم در مراحل بعدی تولید گردد.

یکی از مباحث مطرح در این مرحله تهیه مستندات واسطهای گرافیکی یا بعبارتی نحوه طراحی فرمها و پنجره‌های ورود اطلاعات و گزارش‌گیری می‌باشد که تهیه این مستندات می‌تواند باعث طولانی‌تر شدن این مرحله گردد و اگر چه طراحی فرمها در این مرحله می‌تواند باعث شناخت بهتر نیازهای کاربران شود بنابراین اگر مزیت خاصی برای تهیه این قبیل مستندات موجود نباشد این کار می‌تواند به مراحل پنج یا شش موکول گردد و بهرحال مدیریت پروژه در مورد اینکه کدامیک از گامها باید انجام شود و از چه روشها و ابزارهایی برای تصحیح و بهینه‌سازی استفاده گردد، می‌تواند تصمیم بگیرد.

اهداف این فصل

در این بخش می‌خواهیم پردازش‌های سیستم جدید را تعیین و نمودار ساختار منطقی داده‌ها را برای آن ترسیم کنیم مسلماً برای این کار می‌باید از مدل منطقی جریان داده‌ها و ساختار منطقی داده‌ها برای وضعیت موجود استفاده نماییم و براساس راه‌حل انتخابی و نیازمندی‌هایی که پوشش داده می‌شود پردازش‌های سیستم جدید و ساختار منطقی

داده‌ها را ایجاد کنیم.



شکل ۹.۱ ورودی‌ها و خروجی‌ها گام‌های ۳۱۰ و ۳۲۰

گام‌های ۳۱۰ و ۳۲۰

این دو گام می‌تواند به صورت موازی انجام شوند. ابتدا در مورد گام ۳۲۰ صحبت می‌کنیم در این گام می‌خواهیم ساختار منطقی داده‌ها برای سیستم جدید را بسازیم، این ساختار براساس ساختار وضعیت موجود با در نظر گرفتن نیازهای جدید و سیستم پیشنهادی ایجاد می‌گردد. در این گام سعی می‌شود که موجودیت‌ها به صورت دقیق‌تر شناسایی شوند کلیدها مشخص گردیده و روابط آنها به صورت دقیق شناسایی می‌گردد و مشخص می‌شود که کدام یک از اقلام اطلاعاتی به صورت کلید خارجی می‌باشد.

در گام ۳۱۰ پردازش‌های سیستم جدید را مشخص می‌کنیم این پردازش‌ها براساس مدل منطقی داده‌ها و نمودار تجزیه و تحلیل عملکردها برای وضعیت موجود و با در نظر گرفتن نیازهای جدید ساخته می‌شوند.

برای مشخص کردن پردازش‌ها ابتدا نمودار تجزیه و تحلیل عملکردها را برای سیستم جدید تهیه می‌کنم. این نمودار نشان‌دهنده فعالیت‌های اصلی سیستم است و آخرین سطح نمودار پردازش‌های سیستم جدید را مشخص می‌کند و در نهایت تبدیل به منوی سیستم می‌شود.

اگر بخواهیم نمودار جریان داده‌ها را برای سیستم جدید به طور کامل تهیه کنیم، می‌توانیم آن را براساس نمودار تجزیه و تحلیل عملکردها بسازیم، در غیر این صورت تهیه مستندات زیر کافی است:

۱. نمودار سطح صفر
۲. نمودار تجزیه و تحلیل عملکردها
۳. شرح پردازش‌ها

نمودار سطح صفر حدود سیستم و ارتباط آن را با سایر موجودیت‌ها نشان می‌دهد و از شرح پردازش در بخش‌های آینده برای تعیین شرح کارکردها (Functions) استفاده می‌شود.

از آنجایی که جریان اطلاعات بین موجودیت‌ها و سیستم در گام‌های بعد توصیف می‌گردد و جریان اطلاعات بین اجزای داخلی سیستم نیز در بخش‌های بعدی توضیح داده می‌شود، بنابراین به نظر می‌رسد که نیازی به ایجاد نمودار جریان داده‌ها به صورت کامل نیست.

کار بعدی در این قسمت تعیین نقش کاربران است. در این قسمت وظیفه هر یک از کاربران در رابطه با سیستم مشخص می‌گردد و معلوم می‌شود که سیستم از هر یک چه چیزهایی می‌خواهد شکل ۷.۲ نقش کاربران را در رابطه با سیستم رزرواسیون متل نشان می‌دهد.

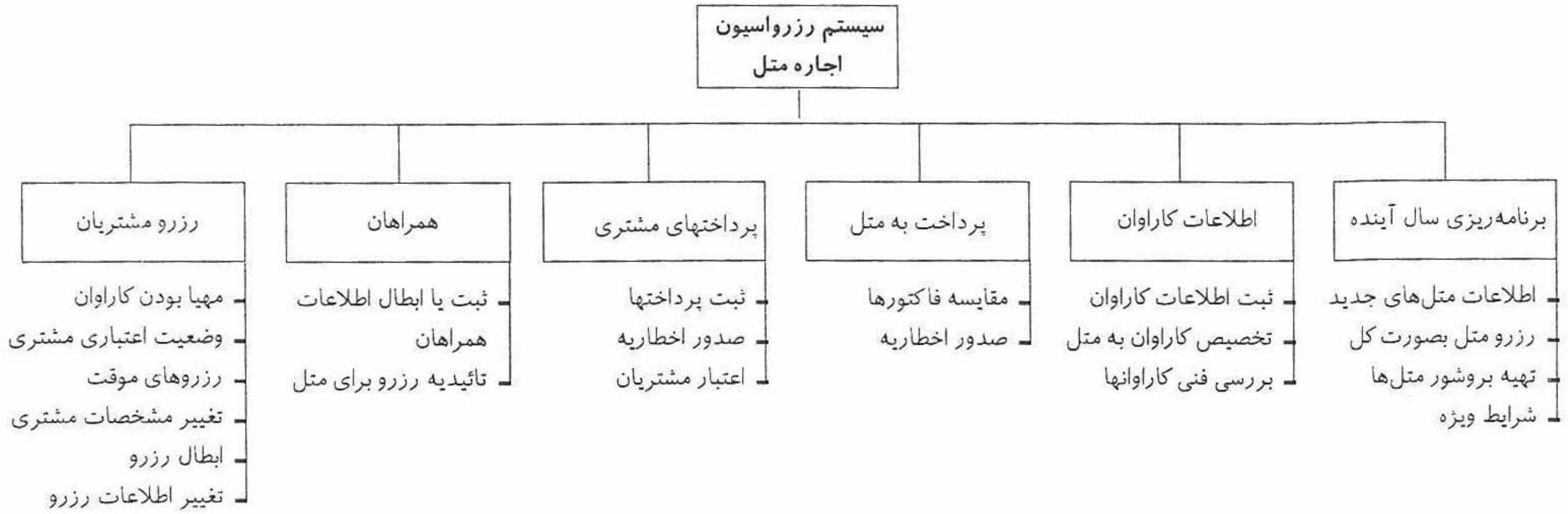
نقش کاربران نام سیستم: تاریخ: صفحه:			
نام نقش کاربر	عنوان شغل	کدکارکرد	نام کارکرد
کارمند امور مشتریان	مدیر امور مشتریان کارمند امور مشتریان		
مدیریت امور مشتریان	مدیریت امور مشتریان مدیریت امور کاراوانها		

شکل ۹.۲ نقش کاربران در سیستم مکانیزه رزرواسیون متل

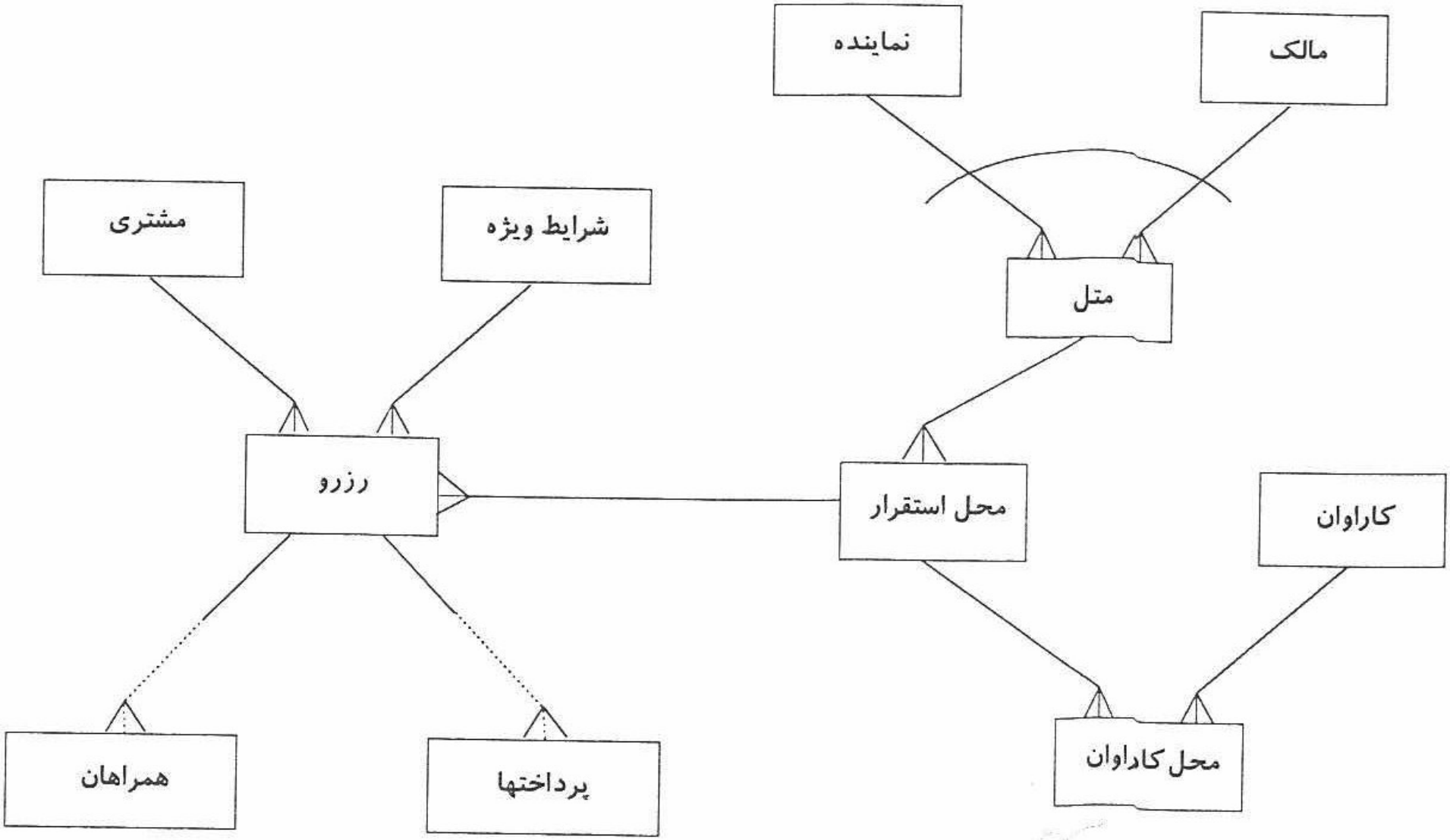
در ستون دوم فرم ۲ـ۹ بایستی عنوان مشاغل با مشاغل ذکر شده در فهرست کاربران مطابقت داشته باشد. به طور مثال این فرم نشان می‌دهد که مدیریت هم می‌تواند نقش کارمند را داشته باشد و هم از گزارشات مدیریتی استفاده کند.

سیستم رزرواسیون متل

فرض کنیم سیستم رزرواسیون اجاره متل را که قبلاً درباره آن بحث کردیم می‌خواهیم مکانیزه نماییم. چون کاراوان‌ها متعلق به بنگاه بوده و می‌توانند بین متل‌ها جابجا شوند. می‌خواهیم اطلاعات هر کاراوان نیز در سیستم مکانیزه نگهداری گردد و اطلاعات جابجایی آنها را داشته باشیم نمودار ساختار منطقی داده‌ها و نمودار تجزیه و تحلیل عملکرد سیستم مکانیزه در شکل‌های ۳ـ۹ و ۴ـ۹ نمایش داده می‌شود و تنها تفاوت میان نمودار ساختار منطقی سیستم مکانیزه و وضعیت موجود در اضافه شدن یک موجودیت بنام کاراوان که نشان‌دهنده ارتباط محل استقرار کاراوان است با خود کاراوان.



شکل ۹.۳ نمودار تجزیه و تحلیل عملکرد برای سیستم رزرواسیون متل



شکل ۹.۴ ساختار منطقی داده‌ها (سیستم مکانیزه رزرواسیون متل)

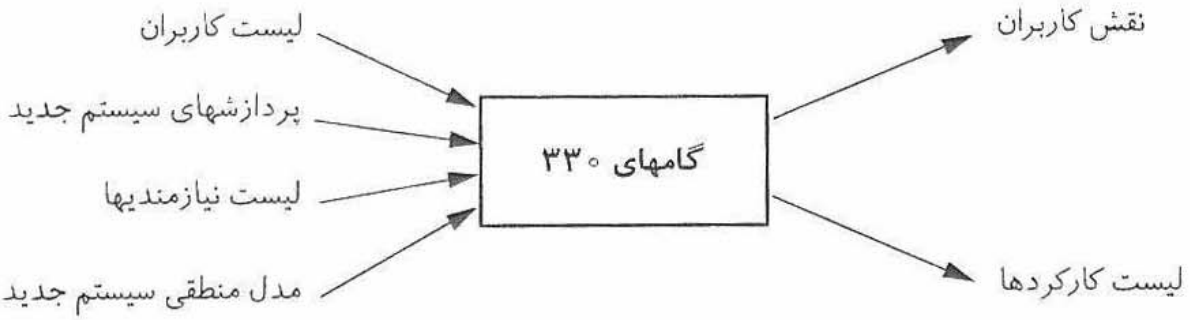
فصل دهم

کارکردها (Functions)

اهداف فصل

در این گام به بررسی و شناخت کارکردهای (Function) سیستم می‌پردازیم و می‌خواهیم پردازش‌ها را به گونه‌ای تعریف کنیم که در مرحله ۶ قابل استفاده باشند.

در گذشته مشخصات برنامه‌ها با توجه به نمودار جریان داده‌ها تهیه می‌شد. نمودار جریان داده‌ها اگر چه وسیله خوبی برای ارتباط با کاربر می‌باشد ولی برای تهیه مشخصات برنامه‌ها ابزار خوبی نیست. در این مرحله نقش و وظیفه کاربران را نیز مشخص می‌کنیم.



۱۰.۱ ورودی‌ها و خروجی‌های گام ۳۳۰

کارکرد چیست؟

کارکرد عبارت است از قسمتی از سیستم که مستقلاً وظیفه مشخص و معینی را از نظر کاربر انجام می‌دهد و در این تعریف به دو مسأله اشاره می‌شود، اول آن که کارکرد وظیفه مشخص و معینی را انجام می‌دهد و دیگر آن که این وظیفه مستقلاً و یکباره انجام می‌شود، یعنی اگر کاربر آن را انتخاب کند کارکرد مزبور مستقل از سایر قسمت‌های سیستم انجام می‌شود. برای آن که مسأله روشن‌تر شود به ذکر چند مثال می‌پردازیم.

یک دستگاه عابر بانک را در نظر بگیرید برای آن که از آن استفاده کنیم ابتدا مشتری کارت خود را درون دستگاه می‌گذارد و سپس رمز آن را وارد می‌کند دستگاه رمز مشتری را با شماره وارد شده تطبیق می‌کند و اگر مطابقت نداشت پس از چند بار که پیغام مناسب می‌دهد آن را می‌بلعد و اگر مطابقت داشت منوی زیر روی صفحه مانیتور دستگاه ظاهر می‌شود.

۱. دریافت وجه
۲. دریافت وجه با صدور رسید
۳. نمایش موجودی
۴. درخواست صورت‌حساب
۵. انتقال وجه بین حساب‌ها
۶. خروج

پس از انجام هر یک از کارهای ۱ تا ۵ دستگاه به این منو باز می‌گردد و این کار تا جایی ادامه پیدا می‌کند که کاربر گزینه ششم را انتخاب و از سیستم خارج می‌شود. حال با توجه به تعریف کارکرد می‌توان گفت که ۵ کارکرد در این جا وجود دارد که عبارتند از:

۱. پرداخت وجه توسط عابر بانک
۲. پرداخت وجه همراه با صدور رسید مربوطه
۳. نمایش موجودی حساب مشتری
۴. پذیرش درخواست صدرو صورت‌حساب و تهیه و ارسال آن
۵. نقل و انتقال وجه بین حساب‌های مختلف

در این جا ذکر چند نکته ضروری است:

۱. پردازش‌های لحظه‌ای اغلب در آخرین سطح منو می‌باشند و با استفاده از یک کلید تعریف شده اجرا می‌شوند پس از این که کارکرد مورد نظر انتخاب شد کنترل از کاربر به سیستم منتقل شده و مثل آن است که کاربر از اتومبیل خود که امکان تعویض مسیر را دارد پیاده شده و سوار قطار گردد که در این صورت تا رسیدن به مقصد نمی‌تواند از آن خارج شود.

بنابراین هر زمان که کاربر موجودی خود را بخواهد، کنترل از کاربر به سیستم منتقل می‌شود و اطلاعات برای او فراهم می‌گردد و در حین اجرای کارکرد کاربر نمی‌تواند مسیر خود را تعویض کند تا بخواهد که کارکرد دیگری را اجرا کند. البته در همهٔ موارد اختیارات کاربر به این اندازه محدود نیست. فی‌المثل اگر کاربر بخواهد از عابر بانک پول بگیرد و میزان وجه مورد نظر را وارد می‌کند ولی همین میزان تحت نظارت سیستم می‌باشد و اگر از موجودی دستگاه یا موجودی مشتری و یا تعداد اسکناس‌هایی که هر بار عابر بانک می‌تواند پرداخت کند بیشتر باشد اجرای کارکرد متوقف می‌گردد.

۲. معمولاً کاربران بین کارکردها انتخاب می‌کنند و نمی‌توانند در هنگام اجرای کارکرد چیزی را انتخاب کنند، بنابراین اگر یک کارکرد در هنگام اجرا از کاربر سؤال کند که می‌خواهد از بین دو کار الف و ب کدام یک را انجام دهد آن کارکرد بایستی خود به دو کارکرد مستقل که یکی کار الف و دیگری کار ب را انجام می‌دهد شکسته گردد.

در پایان باید ذکر گردد که کارکرد با پردازش‌های موجود در نمودار جریان داده‌ها متفاوت است در عین حال اگر چه مشابهت‌هایی موجود هست. کارکرد یک برنامه نیست زیرا مواردی موجود است که برای یک کارکرد به چندین برنامه نوشته می‌شود.

این مسأله با بررسی کارکرد «ارسال صورت‌حساب» به خوبی نشان داده می‌شود. معمولاً برای این کار از دو برنامه استفاده می‌شود که یکی به صورت لحظه‌ای اطلاعات درخواست مشتری را ثبت می‌کند و دیگری در هنگام اجرای برنامه‌های پایان روز از اطلاعات ثبت شده استفاده کرده و صورت‌حساب را چاپ می‌کند.

اگرچه برای اجرای این کارکرد لازم است که دو برنامه متفاوت در دو زمان متفاوت اجرا گردد ولی از نظر کاربر، سیستم یک وظیفه مشخص رابرای او انجام می‌دهد و آنچه برای او اهمیت دارد دریافت صورت‌حساب پس از چند روز می‌باشد.

به عنوان مثال دوم یک سیستم پرداخت حقوق و دستمزد را در نظر بگیرید در این سیستم کارکردهای متفاوتی وجود دارد مثل تغییر حقوق پرسنل یا ثبت اطلاعات پرسنل و غیره ولی مسلماً مهم‌ترین فعالیت آن پرداخت حقوق می‌باشد. برای انجام این کارکرد تعداد زیادی برنامه بایستی اجرا شوند ولی از نگاه کاربر سیستم یک کار مشخص را برای او انجام می‌دهد بنابراین یک کارکرد می‌باشد از طرف دیگر حتی اگر کاربر فعالیت پرداخت حقوق را شامل کارکردهای جدا و پشت سرهم بداند مانند محاسبه حقوق، محاسبه مالیات و صدور فیش حقوقی چون این وظایف به دنبال یکدیگر اجرا می‌شود باز تشکیل یک کارکرد را می‌دهد.

در یک سیستم حقوق و دستمزد فعالیت‌های دیگری از جمله «انتقال اطلاعات پرسنل مستعفی به آرشیو» و یا «انتقال اطلاعات پرداخت حقوق سنوات گذشته به آرشیو» می‌تواند وجود داشته باشد. اگر چه ممکن است این گمان پیش آید که این فعالیت‌ها ربطی به کاربران ندارد و فقط بایستی توسط پرسنل قسمت کامپیوتر انجام شود ولی این کاربران هستند که می‌توانند تعیین کنند تا در چه زمان این اطلاعات به آرشیو

منتقل شود بنابراین باید توسط خود آنان انجام شود و کارکردهای لازم برای آنان در سیستم تعبیه گردد.

به عنوان آخرین مثال یک برنامه کاربردی مثل لغت‌نامه را در نظر بگیرید در نگاه اول این طور به نظر می‌آید که این برنامه کار خاصی را انجام می‌دهد بنابراین یک کارکرد می‌باشد ولی پس از اجرای برنامه و ورود به محیط آن متوجه می‌شویم که وظایف مختلفی را بعهده دارد به جستجوی معنی یک لغت انگلیسی می‌پردازد و کلمات مترادف با یک کلمه را پیدا می‌کند. یک لغت را تلفظ می‌کند که تمامی آنها برای کاربر معنی دارد و براساس تعریف هر کدام از این وظایف یک کارکرد می‌باشد. در تمام مرحله سوم به هر کارکرد به صورت منطقی نگاه می‌شود و کاری به نحوه پیاده‌سازی و جزئیات آن نداریم.

دسته‌بندی کارکردها

می‌توان کارکردها را از جهات مختلف به صورت‌های زیر دسته‌بندی کرد:

۱. به روزرسانی یا پرس‌وجو

هر کارکردی که اطلاعات را تغییر دهد کارکرد به روزرسانی و آنهایی که فقط به اطلاعات دستیابی پیدا می‌کنند کارکرد پرس‌وجو نامیده می‌شود.

۲. کارکرد لحظه‌ای (on line) یا (off line)

تمامی کارکردهایی را که از طریق صفحه کلید با آن ارتباط برقرار می‌کنیم، لحظه‌ای (on line) نامیده می‌شوند. کارکردهایی که در پشت صحنه و با فاصله زمانی اجرا می‌شوند، (off line) نامیده می‌شوند. البته بعضی از کارکردها مانند ارسال صورت‌حساب‌ها استثناء می‌باشند که بخشی از آن حالت لحظه‌ای (on line) و بخشی حالت off line دارد.

۳. توسط سیستم یا کاربر اجرا شوند

اکثر کارکردها توسط کاربران اجرا می‌شوند ولی بعضی از کارکردها به صورت اتوماتیک توسط سیستم اجرا می‌شوند مثل ساخت جداول پشتیبان پس از خاتمه فعالیت‌های روزانه در ساعت مقرر.

اجزاء کارکرد

هر کارکردی از دو بخش تشکیل می‌شود یک بخش آن در ارتباط با کاربران وصفحات ورودی و خروجی اطلاعات بوده و بخش دیگر در ارتباط با وظیفه و عملکرد آن می‌باشد که به چه اطلاعات دستیابی پیدا کند و کدام یک را به روز درآورد؟ به این قسمت در گام ۳۶۵ «تعیین مسیر دستیابی» می‌پردازیم

اگر قرار باشد سیستم توسط زبان‌های نسل چهارم ایجاد شود در آن صورت احتیاجی به ذکر کلیه جزئیات وجود ندارد ولی اگر از زبان‌هایی مثل COBOL قرار است استفاده گردد بایستی در مرحله پنجم به ذکر کلیه جزئیات بپردازیم تا بتوانیم مشخصات و شرح هر برنامه را براساس آن استخراج کنیم.

شناسایی کارکردها

برای شناسایی کارکردها می‌توانیم از پنج منبع استفاده کنیم.

۱. نمودار جریان داده‌ها برای سیستم جدید

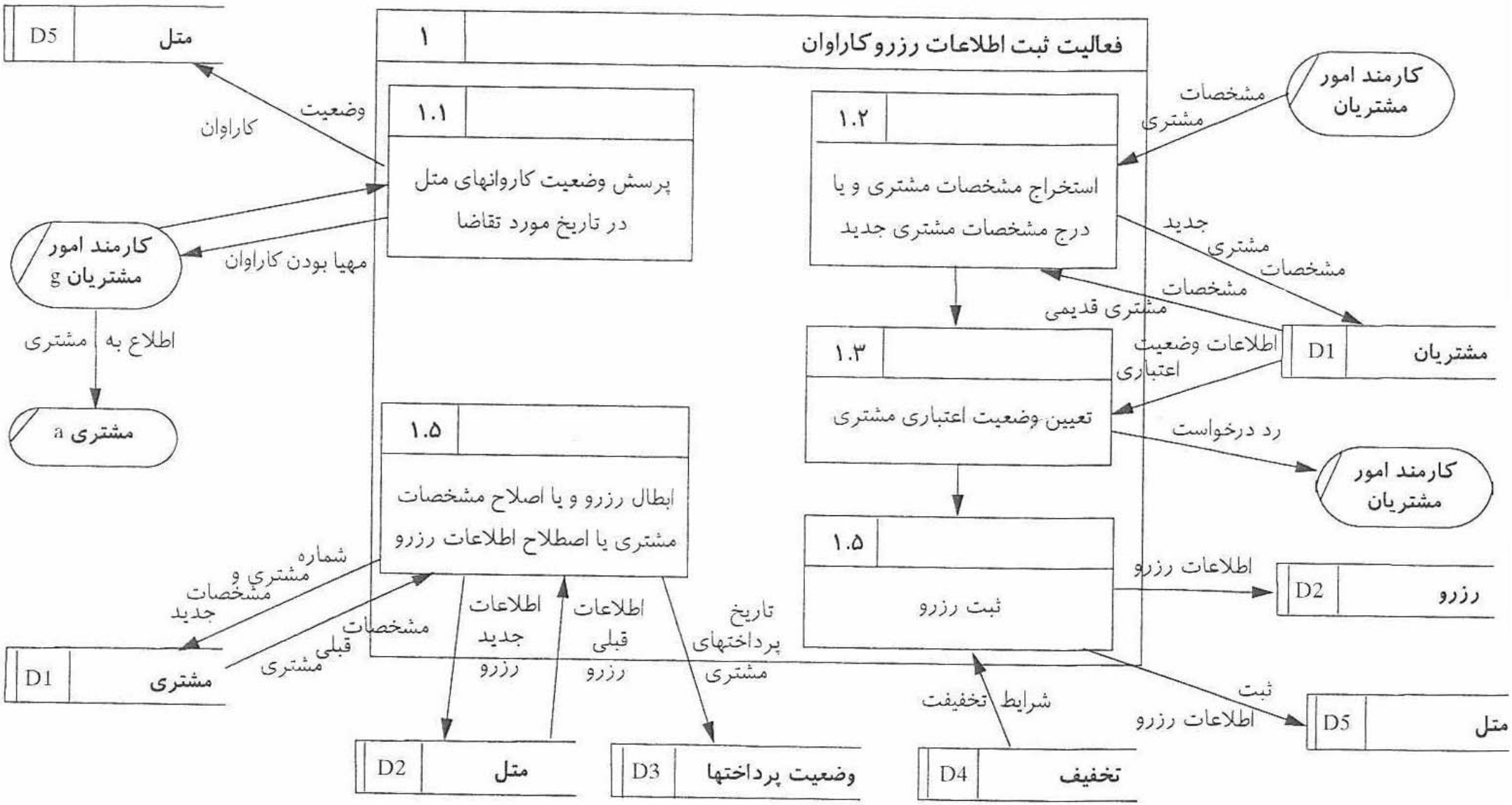
۲. تجزیه و تحلیل عملکرد سیستم جدید

۳. لیست نیازمندی‌ها

۴. ساختار منطقی داده‌ها و گروه‌بندی مربوطه

۵. شناخت سیستم

براساس رهنمودهای SSADM می‌توان کارکردها را از نمودار جریان داده‌ها استخراج کرد بهتر است براساس یک مثال پیش برویم پس به مثال سیستم رزرواسیون متل بر می‌گردیم و براساس آن مطلب را ادامه می‌دهیم. در شکل ۱۰.۳ نمودار جریان داده‌ها برای سیستم رزرواسیون متل آورده شده است نمودار نشان دهنده پردازش‌هایی می‌باشد که در هنگام رزور کاراوان و ثبت اطلاعات مشتری صورت می‌گیرد و براساس این نمودار می‌توان گفت که پردازش ۱۰۱ مسلماً منجر به یک کارکرد می‌شود به نام «وضعیت کاراوان» که نشان‌دهنده وضعیت کاراوان در تاریخ مورد نظر مشتری از جهت رزرو شدن یا خالی بودن آن است.



شکل ۱۰.۲ نمودار جریان داده‌ها برای فعالیت رزرو کاراوان

پردازش‌های ۱۰۲ و ۱۰۳ و ۱۰۴ به دنبال یکدیگر اجرا می‌شوند و طبق تعریف یک کارکرد را شامل می‌شوند. پردازش ۱۰۳ به تنهایی نیز قابل استفاده است بنابراین یک کارکرد دیگر به نام تعیین وضعیت اعتبار مشتری نیز موجود است و در این صورت این دو کارکرد با هم قسمت‌های مشترکی دارند که طبیعتاً می‌توانند از برنامه مشترکی استفاده کنند ولی در این قسمت به لحاظ منطقی به صورت دو کارکرد جدا به آن نگاه می‌کنیم.

پردازش ۱۰۵ از دو قسمت تشکیل شده و برای اجرای آن از کاربر پرسیده می‌شود که می‌خواهد رزرو را باطل کند و یا اطلاعات رزرو را اصلاح کند بنابراین چون این سؤال مطرح می‌شود که دلالت بر وجود دو کارکرد می‌باشد و به علاوه این کارکرد می‌بایست بر روی نمونه‌های (occurance) دو موجودیت کار کند. بنابراین بهتر است تبدیل به دو کارکرد شود.

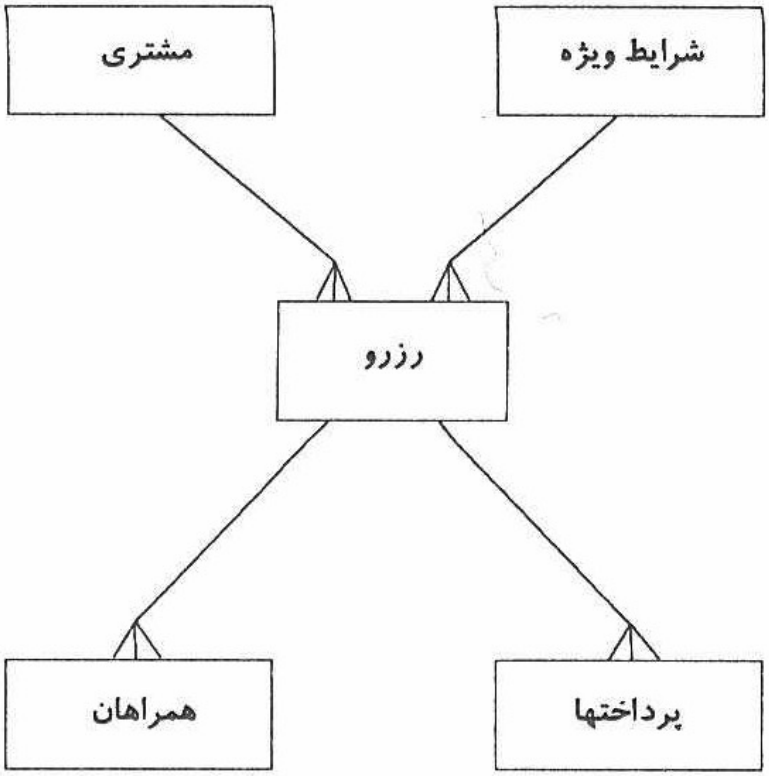
پس می‌توان نتیجه گرفت که نمی‌توان یک پردازش را با یک کارکرد تطبیق کرد بلکه در پاره‌ای موارد یک پردازش منجر به ایجاد کارکردهای زیادی می‌شود و در مواردی بالعکس چندین پردازش منجر به ایجاد یک کارکرد می‌شوند.

در فصل قبل گفته شد که می‌توان بدون ترسیم نمودار جریان داده‌ها و فقط براساس نمودار تجزیه و تحلیل عملکرد بتوانیم کارکردها را تعیین کنیم. در این صورت سطوح آخر نمودار کارکردها را تعیین می‌کند و برای ایجاد نمودار صحیح تجزیه و تحلیل عملکردها بایستی فعالیت‌های سیستم و محدودهٔ آن را به خوبی شناخت و بر این اساس عمل کرد ولی باز به مراتب مطمئن‌تر است که براساس نمودار جریان داده‌های سیستم جدید پیش برویم اگرچه زمان بیشتری گرفته می‌شود.

ممکن است نتوانیم بسیاری از کارکردها را براساس نمودار جریان داده‌ها یا نمودار تجزیه و تحلیل عملکرد شناسایی کنیم می‌توان گفت که گزارشات موردی از این دسته هستند پس برای شناسایی آنها باید به لیست نیازمندی‌ها مراجعه کنیم و بر این اساس کارکردها را تعیین نماییم.

با توجه به ساختار منطقی داده‌ها نیز می‌توان کارکردهای زیادی را شناسایی کرد. در این جا می‌پرسیم اطلاعات کدام یک از موجودیت‌ها با هم و کدام یک جدا به روزآوری می‌شوند و اگر به شکل ۱۰.۳ توجه کنیم مشاهده می‌کنیم که در زمان ثبت رزرو کاراوان اطلاعات دو موجودیت رزرو و افراد همراه با یکدیگر درج می‌شوند اما آیا می‌توان

تعداد افراد همراه را با استفاده از کارکرد رزرو اضافه و یا کم کرد و یا اطلاعات آنان را تغییر داد اگر جواب منفی است بایستی کارکرد جداگانه‌ای برای این منظور در نظر گرفت.



شکل ۱۰.۳ قسمتی از ساختار منطقی داده‌ها برای سیستم رزرواسیون

آخرین راه شناسایی کارکردها مصاحبه با کاربران است حتی اگر نمودار جریان داده‌ها به بهترین صورت تهیه شده باشد باز احتمال بسیار وجود دارد که پس از مصاحبه با کاربران کارکردهای جدیدی شناسایی شود.

مصاحبه با کاربران و ساختار منطقی داده‌ها ممکن است نیازهای جدیدی را مشخص کند که حتی در لیست نیازمندی‌ها نیامده است و چون قبلاً زمان و هزینه سیستم پیشنهادی برآورد شده است، باعث ایجاد تغییراتی گردد و برآورد زمان و هزینه پروژه تغییر کند. این تغییرات بایستی با کاربران در میان گذاشته شود.

پس از شناسایی کارکردها می‌توان فهرست آنها را ایجاد کرد و در فرمی مطابق شکل ۱۰.۴ نمایش داد.

فهرست کارکردها

نام سیستم: سیستم مکانیزه رزرواسیون متل‌ها **تاریخ:** **شماره صفحه:**

کدکارکرد	مرجع	نام کارکرد	نوع کارکرد
۵	۱۰۱	بررسی وضعیت کاراوان	لحظه‌ای ـ پرس و جو

شکل ۱۰.۴ فهرست کارکردها (سیستم رزرواسیون متل‌ها)

مستندسازی کارکردها

پس از شناسایی کارکردها و تهیه فهرست آنها به مستند کردن وظایف هر کارکرد و فرم‌ها و صفحات ورودی و خروجی آن می‌پردازیم برای مستند کردن وظایف هر کارکرد فرم زیر که شامل دو قسمت است، پیشنهاد می‌شود.

شرح کارکردها (قسمت اول)

نام سیستم: سیستم مکانیزه رزرواسیون متل‌ها **کدکارکردها** **تاریخ**

نام کارکرد: رزرو موقت	**نوع کارکرد**

این کارکرد از سه قسمت استخراج اطلاعات مشتری و بررسی وضعیت اعتباری او و صدور تأییدیه رزرو موقت تشکیل می‌شود و اگر مشتری جدید باشد قسمت ثبت اطلاعات مشتری نیز اضافه می‌شود پس از بررسی اعتبار مشتری (مشتری جدید همیشه اعتبارش پذیرفته می‌شود) و برآورد تعیین میزان هزینه اطلاعات زمان‌های پرداخت مشتری ایجاد می‌شود.

مغایرت‌ها	اقلام اطلاعاتی ورودی	خروجی‌ها
۱. اگر کد اعتبار مشتری B یا Cباشد رزرو انجام نمی‌شود. ۲. اگر روز آغاز رزرو شنبه باشد رزرو انجام نمی‌شود. ۳. اگر کاراوان قبلاً رزروشده باشد رزرو انجام نمی‌شود.	کد مشتری مشخصات مشتری (اگر جدید باشد) کد متل کد محل استقرار تاریخ نام همراهان کد شرایط ویژه	صفحه نمایش رزرو نامه تأییدیه نامه ابطال

شکل ۱۰.۵

اقلام اطلاعاتی قسمت اول این فرم‌ها به شرح زیر می‌باشد:

۱. کد کارکرد: به هر یک از کارکردها یک کد اختصاص می‌دهیم تا بتوانیم آن را شناسایی کنیم.

۲. نام کارکرد: نام کارکرد مربوطه ذکر می‌گردد.

۳. نوع کارکرد: از چه نوع است وظیفه به روزرسانی را انجام می‌دهد یا پرس‌وجو؟ از نوع on line است یا off line

۴. شرح کارکرد: در این جا وظیفه کارکرد به صورت کلی و به زبان ساده شرح داده می‌شود و از به کار بردن لغات تکنیکی اجتناب می‌کنیم تا کاربران بتوانند آن را درک کنند.

۵. مغایرت‌ها (Errors):

در این بخش اشتباهات یا مغایرت‌هایی که اطلاعات با اصول کار کاربران دارد ذکر می‌گردد و به اشتباهاتی که در فرمت ورودی اطلاعات وجود دارد مورد نظر قرار نمی‌گیرد. (مثلاً اگر در کد قطعه می‌بایستی کلید کاراکترها عددی باشند) این نوع اشتباهات در فصول بعدی مستند می‌شوند آنچه در این بخش مهم است اصول کار کاربران می‌باشد بعضی از این مغایرت‌ها در شکل ۱۰.۵ توضیح داده شده است مثلاً اگر رده اعتباری مشتری B یا C باشد رزرو برای او صورت نمی‌گیرد یا آغاز روز رزرو نمی‌تواند شنبه باشد شناسایی این گونه مغایرت‌ها بسیار مهم است و حتماً در فرم مربوطه باید ثبت گردد. اطلاعات ذیل در قسمت دوم فرم ثبت می‌گردد.

اقلام اطلاعاتی: اقلام اطلاعاتی ورودی توضیح داده می‌شود. و به فرم الصاق می‌گردد.

خروجی‌ها: خروجی‌های کارکرد در این قسمت مشخص می‌شود و طرح (layout) آنها به فرم الصاق می‌گردد.

کاربران (نقش کاربران): چه کاربرانی می‌توانند از این کارکرد استفاده کنند.

مرجع: در این قسمت مرجع کارکرد مشخص می‌گردد که لیست نیازمندی‌ها است و یا غیره.

رویداد: بر اثر چه رویدادهایی این کارکرد بایستی عمل کند.

کارکردهای مرتبط: چه کارکردهایی با این کارکرد به خصوص در ارتباط می‌باشد. به طور مثال برای رزرو کاراوان کارمند مربوطه باید به کد مشتری و کد متل مربوطه دسترسی داشته باشد و کاراوان‌های خالی را بشناسد. پس کارکردهایی در رابطه باید

برای او ایجاد گردد.

پردازش‌های مشترک: برنامه‌هایی که به نظر می‌رسد در این جا کارکرد استفاده می‌شود و با سایر کارکردها مشترک می‌باشد، در این جا ذکر می‌شود. به طور مثال برنامه تست درستی تاریخ.

ملاحظات: در این قسمت نکات مهمی که بایستی در طراحی برنامه‌های کارکرد در نظر گرفته شود ذکر می‌گردد.

پریود اجرا: از این کارکرد چند بار در روز یا هفته استفاده می‌گردد.

میزان سرویس‌دهی: حداکثر زمان پاسخ‌دهی سیستم برای این کارکرد چه می‌باشد. این مطلب بیشتر برای کارکردهای لحظه‌ای مورد استفاده است.

شرح کارکردها (قسمت دوم)
نام سیستم: سیستم رزرواسیون متل‌ها **کدکارکردها** ۴۱ **تاریخ** ۸۰/۶/۶
نام کارکرد: ثبت رزرو موقت
کاربری بر: کارمند امور مشتریان
رویداد: رزرو موقت
منبع: مستندات پردازش‌های ۱.۲ و ۱.۳ و ۱.۴ در مدل منطقی داده‌ها
کارکردهای مرتبط **کارکرد ۴۰** - پرسش وضعیت کاراوان **کارکرد ۴۵** - استخراج مشخصات مشتری
ملاحظات
پردازشهای مشترک
پریود اجرا: ۳۵ بار در روز
میزان سرویس‌دهی: حداکثر زمان استخراج اطلاعات ۳ ثانیه

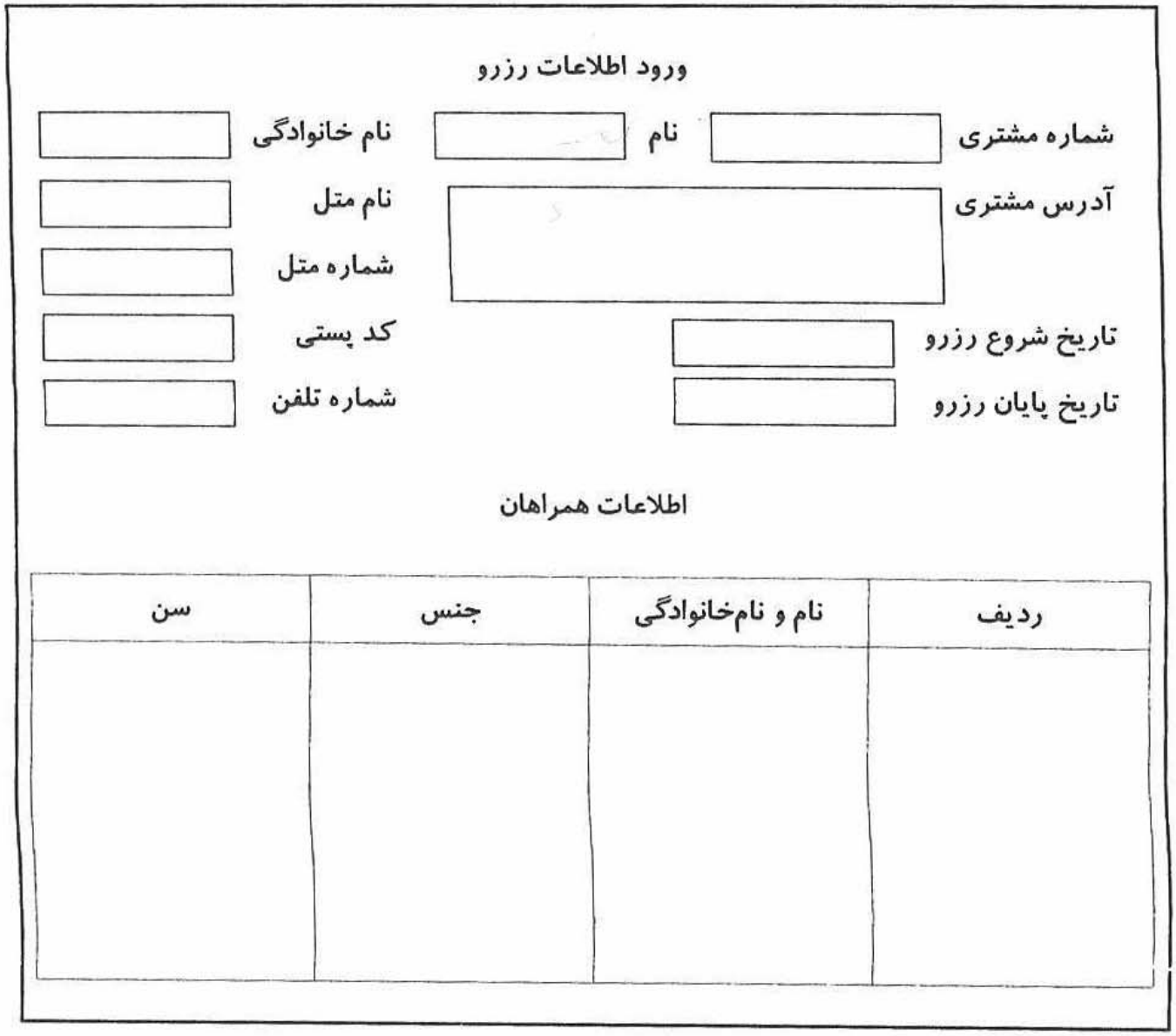
شکل ۱۰.۵ قسمت دوم

طراحی صفحات ورودی و خروجی

در SSADM از نمودارهای جاکسون برای نمایش اقلام اطلاعاتی ورودی و خروجی استفاده می‌شود ولی این نمودارها ابزار خوبی برای ارتباط با کاربران نیستند به همین دلیل از این نمودارها استفاده نکردیم.

طراحی فرم‌ها و گزارشات، علاوه بر استانداردهای سازمان به سلیقه‌های شخصی نیز بستگی دارد و محیط نرم‌افزاری و ابزارهایی که در اختیار می‌گذارد نیز یکی از پارامترهای اصلی طراحی فرم‌ها و گزارشات می‌باشد. اما آنچه در پی می‌آید توصیه‌هایی به خواننده است تا بتواند طراحی صفحات و گزارشات را انجام دهد.

به طور مثال کارکرد رزرو موقت را در نظر بگیرید این کارکرد از سه فرمت برای صفحات ورودی و خروجی استفاده می‌کند یک فرمت برای صفحات ورودی و یکی برای نامه تأییدیه و یکی برای نامه عدم تأیید شکل ۱۰.۶ نشان‌دهنده فرم ورودی در صفحهٔ مانیتور و شکل ۱۰.۷ نشان‌دهنده نامه تأییدیه و یا عدم تأییدیه که مشابه آن است، می‌باشد.



شکل ۱۰.۶ صفحه ورود اطلاعات رزرو

نام و نام خانوادگی مشتری

مشتری گرامی خانم / آقای نام و نام خانوادگی مشتری

خوشوقتم که به اطلاع شما برسانم که یکی از کاراوان‌های متل _____

از تاریخ

تا تاریخ

برای شما رزرو شده است و هزینه کاراوان به مبلغ _____ تا تاریخ _____ برای ما ارسال نمایید.

با احترام رئیس قسمت رزرو

نام تأییدیه رزرو موقت ،

(سیستم رزرواسیون مکانیزه)

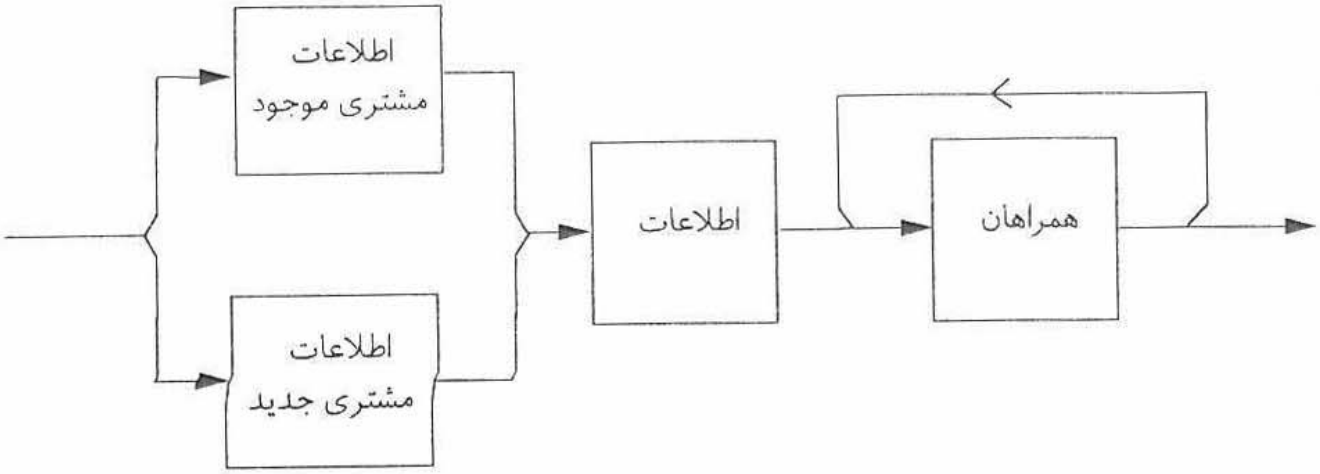
شکل ۱۰.۷ تأئیدیه رزرو موقت

بسیاری از کارکردها می‌توانند از یک صفحه برای ورود اطلاعات استفاده کنند ولی همهٔ کارکردها این گونه نیستند و ممکن است از چندین صفحه برای ورود اطلاعات استفاده می‌کنند در این صورت روشی برای نمایش حرکت بین صفحات ورودی آن کارکرد باید در نظر گرفته شود. در شکل ۷-۱۰ فرض شده که کارکرد رزرو موقت از چهار صفحهٔ ورودی اطلاعات استفاده می‌کند و نحوهٔ گردش اطلاعات بین این صفحات نشان داده شده است. ابتدا کاربر صفحهٔ اطلاعات مشتری موجود و یا صفحهٔ اطلاعات مشتری جدید را انتخاب می‌کند پس از آن صفحهٔ رزرو موقت برای کاربر نشان داده می‌شود و سپس صفحهٔ ورود اطلاعات همراهان نمایش داده می‌شود.

در این جا فقط اطلاعات هر صفحه مورد نظر است و نحوه حرکت بین صفحات بعداً بررسی می‌گردد و کلام آخر آن که در محیط ویندوز هر کارکرد می‌تواند در یک پنجره نشان داده شود و این مسأله کمک می‌کند تا چندین پنجره در عین حال بر روی صفحه مانیتور باز باشد و چندین کارکرد بتوانند به صورت موازی اجرا شوند و از اطلاعات آنان

استفاده گردد.

به طور کلی در این گام طراحی صفحات و فرم‌ها به صورت منطقی مورد نظر است و طراحی فیزیکی آنان به بعد موکول می‌شود.



شکل ۱۰.۸ نحوه حرکت بین صفحات

حقوق کاربران (User Role)

آخرین کار در این گام تهیه فهرستی به نام حقوق کاربران است. در ابتدا در مراحل اولیه SSADM لیست کاربران احتمالی سیستم جدید در فهرستی به نام فهرست کاربران تنظیم شد.

در این قسمت می‌خواهیم تصمیم بگیریم که هر کاربر چه کارکردهایی را می‌تواند اجرا کند. به عبارتی می‌خواهیم منویی را که برای هر کاربر روی صفحه ظاهر می‌شود را تعیین کنیم و مشخص نماییم که هر کاربر چه کارکردهایی را می‌تواند اجرا کند؟

سپس فرمی مطابق شکل ۹-۱۰ تنظیم می‌کنیم و کارکردهایی را که توسط هر کاربر اجرا می‌شود در آن مشخص می‌نماییم. اگر یک کارکرد توسط چندین کاربر اجرا شود چندین بار در لیست ظاهر می‌شود.

در SSADM به جای فرم شکل ۹-۱۰ از یک ماتریس استفاده می‌شود که یک بعد آن کاربران و بعد دیگرش کارکردها می‌باشد. در شکل ۹-۱۰ ماتریس سیستم رزرواسیون مکانیزه نمایش داده شده است.

ولی ترسیم ماتریس برای یک سیستم کوچک با چند کاربر می‌تواند خوب باشد ولی برای یک سیستم بزرگ با تعداد زیادی کاربر نمی‌تواند تصویر درستی ارائه کند و SSADM تهیه لیست حقوق کاربران را نیز براساس این ماتریس پیشنهاد می‌کند.

حقوق کاربران

نام سیستم: سیستم رزرواسیون **تاریخ:** ۱۳۸۰/۰۳/۰۱ **صفحه:** ۱

نام کاربر	عنوان	کدکارکرد	شرح
کارمند	کارمند امور متل‌ها	۴۰	مهیا بودن کاراوان
	مدیریت مشتریان	۴۱	رزرو موقت
		۴۴	تغییر تاریخ رزرو
		۴۵	نمایش مشخصات مشتری
		۴۷	نمایش مشخصات همراهان
		۴۸	اضافه کردن همراهان
		۴۹	حذف همراهان
مدیریت	مدیریت مشتریان	۴۰	مهیا بودن کاراوان
	مدیریت کاروانها	۴۲	تغییر مشخصات مشتری
		۴۳	ابطال رزرو
		۴۵	نمایش مشخصات مشتری
		۴۶	نمایش رده اعتباری مشتری

شکل ۱۰.۹ حقوق کاربران (سیستم رزرواسیون)

کدکارکرد	شرح کارکرد	کارمند امور مشتریان	کارمند حسابداری	مدیریت
۴۰	پرسش وضعیت کاراوان	×	×	×
۴۱	رزرو موقت	×		
۴۲	اصلاح مشخصات مشتری			×
۴۳	ابطال رزرو			×
۴۴	تغییر تاریخ رزرو	×		
۴۵	پرسش مشخصات مشتری	×	×	×
۴۶	نمایش وضعیت اعتباری مشتری		×	×
۴۷	اصلاح اطلاعات همراهان	×		
۴۸	اضافه کردن همراه	×		
۴۹	حذف همراه	×		
۵۰	ثبت پرداختهای مشتری		×	

شکل ۱۰.۱۰ ماتریس کاربران / کارکردها

فصل یازدهم

تجزیه و تحلیل منطقی اطلاعات
(منطقی‌سازی گام ۳۴۰)

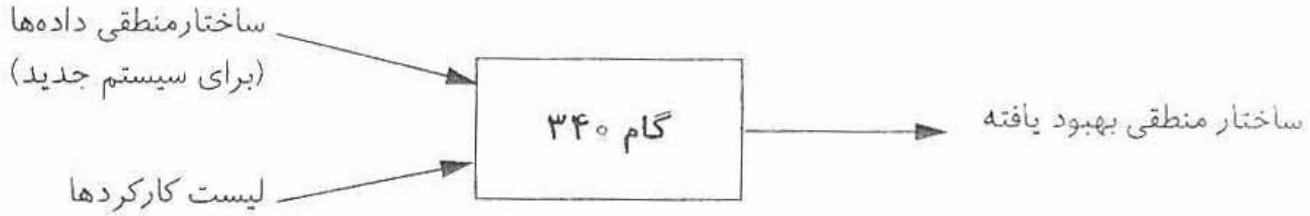
اهداف فصل

ساختار منطقی داده‌ها براساس نظرات تحلیل‌گر راجع به اطلاعاتی که در سیستم نگهداری می‌شود، در گام‌های قبلی ایجاد شده است. تحلیل‌گران باتجربه می‌توانند سریعاً به رسم این نمودار بپردازند و آن را ایجاد کنند. ولی ممکن است اشتباهاتی رخ دهد، بنابراین در این گام به بررسی این نمودار می‌پردازیم.

در این گام براساس صفحات ورودی اطلاعات (فرم‌ها) و گزارشات هر کارکرد که در گام پیش تعیین شده‌اند ساختار منطقی داده‌ها ایجاد و با ساختار قبلی مقایسه می‌شود، بدین لحاظ که ساختار جدید براساس فرم‌ها و گزارشات ایجاد گردیده می‌توان گفت که ساختار جدید براساس روش پایین به بالا ساخته می‌شود برخلاف ساختار قبلی که با استفاده از نظرات تحلیل‌گر و براساس روش بالا به پایین ساخته شده است.

در طراحی ساختار جدید تمامی فرم‌ها و گزارشات مدنظر قرار گرفته شده‌اند. پس می‌توان گفت کلیه اقلام اطلاعاتی در نظر گرفته شده و چیزی از قلم نیفتاده است.

گام ۳۴۰ محل مناسبی برای این کار است چون فرم‌ها و گزارشات در گام قبلی ایجاد شده و براساس آن قبل از این که وارد گام بعدی شویم می‌توانیم ساختار منطقی موجود را اصلاح کنیم.



شکل ۱۱.۱ ورودی‌ها و خروجی‌های گام ۳۴۰

مقدمه‌ای بر منطقی‌سازی

منطقی‌سازی داده‌ها اولین بار براساس مطالعات دکتر کد[۱] انجام شد. مطالعات دکتر کد منجر به ایجاد مدل پایگاه داده‌ها[۲] از نوع رابطه‌ای شد. دکتر کد متوجه شد برای این که بتوان اطلاعات را نگهداری کرد ابتدا بایستی اطلاعات خام را پالایش کرده و به صورت نرمالیزه درآورد و سپس آن را در یک پایگاه داده نگهداری نمود. برای این کار

سه مرحله کار باید بر روی اطلاعات خام صورت گیرد، ابتدا اطلاعات به صورت فرم اول نرمال و سپس فرم دوم و بعد از آن به صورت فرم سوم نرمال درمی‌آید که قابل نگهداری در یک پایگاه داده از نوع رابطه‌ای می‌باشد. البته فرم‌های چهارم و پنجم نرمال نیز شناسایی شده‌اند ولی معمولاً در طراحی پایگاه داده‌ها از آن استفاده نمی‌شود.

جداول

مطالعات دکتر کد که براساس تئوری مجموعه‌ها صورت گرفته است نشان می‌دهد که اطلاعات را باید در جداولی نگهداری کنیم. در این جداول می‌توان یک یا چند ستون را در نظر گرفت ستونی یا ستون‌هایی که سطرهای (Tuple) جدول را به صورت یکتا شناسایی می‌کند. را کلید اصلی (Primary key) می‌نامیم. جدول شکل ۲-۱۱ را در نظر بگیرید در این جدول می‌توان براساس کد پرداخت، سایر اطلاعات از قبیل این که به چه کسی در چه قسمتی پرداخت صورت گرفته را به صورت یکتا شناسایی کرد. برای نشان دادن هر جدول می‌توان به صورت خلاصه اقلام اطلاعاتی آن را مشخص کرده و کلید آن را با یک خط نشان داد به طور کلی خصوصیات جدول را می‌توان به شرح زیر بیان کرد:

۱. هیچ سطر تکراری در جدول وجود نداشته باشد و هیچ دو سطری مشابه یکدیگر نباشند.
۲. ترتیب سطرها اهمیتی ندارد.
۳. ترتیب ستون‌ها اهمیتی ندارد. اگر چه رسم بر این است که کلیدها در ابتدای جدول آورده شوند.
۴. نام ستون‌ها در جداول به صورت یکتا باشد اگر چه از یک نام می‌توان در جداول مختلف استفاده کرد.
۵. مقادیر هر ستون تک مقداری باشد یعنی فقط یک مقدار خاص را نشان دهد. به طور مثال اگر اقلام اطلاعاتی زیر مشخص‌کننده جدول سفارشات باشند.

شماره سفارش

نام مشتری

آدرس مشتری

کد کالاهای سفارش شده

در این صورت در ستون کد کالای سفارش شده به تعداد کالاهای سفارش شده کد

وجود دارد بنابراین برای نمایش این قلم اطلاعاتی باید از جدول دیگری به نام اقلام سفارش استفاده کنیم که آن جدول شامل اقلام اطلاعاتی کد کالا، کد سفارش و شماره ردیف می‌باشد.

کلیدها

همان طور که اشاره شد به مجموعه یک یا چند ستون که سطرهای جدول را به صورت یکتا شناسایی می‌کنند کلید می‌گوییم کلیدها را می‌توان به صورت زیر دسته‌بندی کرد:

۱. **کلید ساده:** به یک قلم اطلاعاتی که سطرها را مشخص می‌کند، کلید ساده می‌گوییم به طور مثال کد کارمند در جدول کارمندان و کد قطعه در جدول قطعات.

۲. **کلید مرکب (Compound key):** از دو یا چند قلم اطلاعاتی که هر کدام در جدول دیگری کلید می‌باشند تشکیل شده است. جدول ۳-۱۱ را در نظر بگیرید، بهترین کلید برای این جدول «کد کارمند / کد پروژه» می‌باشد که هر کدام از این دو یکی در جدول کارمندان و دیگری در جدول پروژه‌ها کلید می‌باشند.

۳. **کلید ترکیبی (Composite key):** از چند قلم اطلاعاتی که قسمتی از این اقلام در جدول دیگری کلید می‌باشد تشکیل شده است. به طور مثال جدول ۴-۱۱ را در نظر بگیرید که نشان‌دهنده اطاق‌های ساختمان‌ها می‌باشد. در این جا کلید از دو قسمت کد ساختمان و شماره اطاق تشکیل شده است که کد ساختمان در جدول ساختمان‌ها کلید می‌باشد ولی شماره اطاق در هر ساختمان از ۱ شروع می‌شود و بنابراین تکراری است پس قسمتی از کلید این جدول در جدول دیگری کلید می‌باشد ولی قسمت دیگر کلید جدولی نمی‌باشد.

۴. **کلید جایگزین:** قلم اطلاعاتی را گویند که می‌تواند کلید باشد ولی کلید در نظر گرفته نشده است مثل نام کارمند در جدول کارمندان در صورتی که نام تکراری نداشته باشیم.

۵. **کلید خارجی:** اگر یک قلم اطلاعاتی در جدول دیگری کلید باشد به آن کلید خارجی می‌گویند. به طور مثال جداول شکل ۵-۱۱ را در نظر بگیرید قلم اطلاعاتی «کد قسمت» در جدول الف متناظر با مقادیر همین قلم اطلاعاتی در جدول ب می‌باشد که این قلم اطلاعاتی در جدول ب کلید می‌باشد.

کدپرداخت	نام	کدشعبه	کدقسمت
۱۱۰۰۱	حسین طباطبایی	۱۱	۱۵
۱۱۰۰۲	حسین دیلمقانی	۱۰	۱۲

شکل ۱۱.۲ جدول پرداخت

کدپروژه	کدکارمند	نام	ساعات کار
P11	۴۴۴	حسن طباطبایی	۵
P12	۵۴۴	حسین دیلمقانی	۶

کلید جدول
(کد پروژه‌ای کدکارمند)

شکل ۱۱.۳ جدول کارکرد کارمندان در پروژه

کدساختمان	شماره اطاق	ظرفیت
P11	۱	۵
P12	۳	۲۵

کلید جدول
(کد ساختمان/شماره اطاق)

شکل ۱۱.۴ جدول اطاق‌های ساختمان‌ها

کدپروژه	نام	کدشعبه	کدقسمت
P۱۵	حسین طباطبایی	۱۱	۵۱
P۱۶	اکبر دیلمقانی	۱۷	۵۲

جدول الف (پرداخت‌ها)

کدقسمت	نام قسمت
۵۱	امور مالی
۵۲	امور اداری

جدول ب (قسمت‌ها)

شکل ۱۱.۵

فرم نرمال از نوع سوم

اگر جدولی دارای خواص زیر باشد می‌توانیم بگوییم دارای فرم نرمال از نوع سوم است.

۱. برای هر کلید سایر اقلام اطلاعاتی فقط یک مقدار داشته باشند.

۲. تمامی اقلام اطلاعاتی فقط به کلید وابسته باشند.

اکثراً برای نرمال‌سازی و تبدیل اطلاعات به فرم سوم نرمال ابتدا اطلاعات را به فرم نوع اول و سپس نوع دوم و پس از آن نوع سوم تبدیل می‌کنند ولی در روشی که در ذیل بیان می‌گردد اطلاعات مستقیماً به فرم سوم نرمال تبدیل می‌شوند.

در این روش:

۱. اقلام اطلاعاتی را از فرم‌های صفحات ورودی و گزارشات خروجی که در لیست کارکردها موجود است استخراج می‌کنیم.

۲. ترسیم نمودار وابستگی برای کلیه گزارشات و فرم‌های ورودی که در نظر گرفته شده است (طرز رسم آن در صفحات بعد ارائه می‌گردد)

۳. ایجاد جداول به صورت فرم سوم نرمال و تعیین اقلام اطلاعاتی آنها

۴. منطقی‌سازی جداول: براساس نمودار وابستگی جداول زیادی به دست می‌آید که تا حد زیادی اطلاعات مشترکی دارند با این کار قسمت‌های مشترک حذف می‌گردند.

۵. ایجاد ساختار منطقی داده‌ها: براساس جداول و موجودیت‌های به دست آمده نمودار ساختار منطقی داده‌ها را ترسیم می‌کنیم.

۶. تهیه ساختار منطقی بهبود یافته: ساختار منطقی به دست آمده را با ساختار قبلی که در گام ۳۲۰ ایجاد شده بود مقایسه و نواقص احتمالی را برطرف می‌کنیم.

نمودار وابستگی

برای ترسیم نمودار وابستگی از نمادهای موجود در شکل ۶-۱۱ استفاده می‌شود این شکل نشان می‌دهد که B وابسته به A است یعنی برای هر مقدار A یک و فقط یک مقدار برای B وجود دارد مثال کاربردی آن در شکل ۷-۱۱ نشان داده شده است که نام کارمند وابسته به شماره کارمندی می‌باشد به عبارت دیگر به ازای هر شماره کارمندی نام یک کارمند موجود است.



نمودار وابستگی ۱۱.۶



نمودار وابستگی نام کارمند به شماره کارمندی ۱۱.۷

با توجه به شکل ۸-۱۱ متوجه می‌شویم که این شکل درست نیست زیرا در هر فاکتور چندین شماره قطعه وجود دارد. در صورتی که شکل نشان می‌دهد برای هر فاکتور فقط یک شماره قطعه موجود است.



شکل ۱۱.۸

پس به طور کلی برای رسم نمودار وابستگی کارهای زیر را انجام می‌دهیم:

۱. نوشتن تمام اقلام اطلاعات در مستطیل مربوطه
۲. شناسایی کلیدها
۳. ترسیم خطوط وابستگی به طوری که تمام اقلام اطلاعاتی به کلیدها متصل شده باشند و ارتباط بین کلیدها را شناسایی می‌کنیم.
۴. کلیدهای جدید شناسایی شده را مشخص می‌کنیم.
۵. وابستگی غیرمستقیم را برطرف می‌کنیم.

برای نمایش نحوه رسم نمودار وابستگی به ذکر چند مثال می‌پردازیم.

ترسیم نمودار وابستگی

گزارش ۹-۱۱ را در نظر بگیرید برای رسم نمودار وابستگی این گزارش، ابتدا تمامی اقلام اطلاعاتی را درون مستطیل نمایش داده و تمامی کلیدهای اصلی را با یک علامت

" + " مشخص می‌کنیم و خطوط وابستگی از کلید به سایر اقلام وابسته را رسم می‌کنیم.

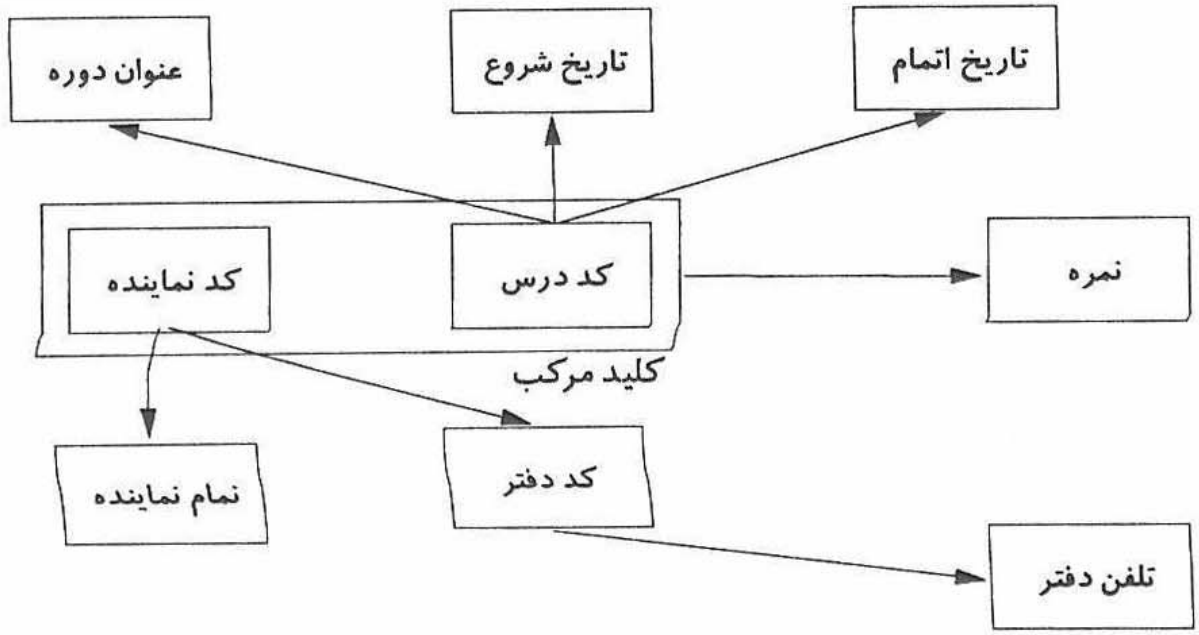
و در نهایت وابستگی کلیدها مورد نظر قرار می‌گیرد که نتیجه آن رسم شکل ۱۰-۱۱ می‌باشد و مستطیل بزرگ نشان‌دهنده کلید مرکب کد نماینده کد درس می‌باشد که قلم اطلاعاتی «نمره» وابسته به این کلید است پس براساس این نمودار می‌توان چهار جدول زیر را در نظر گرفت.

جدول‌دوره‌ها	جدول‌نمایندگی‌ها	جدول‌دفاترنمایندگی	جدول‌دوره/نماینده
کد دوره	**کد نماینده**	**کد دفتر**	**کد دوره**
نام دوره	نام نماینده	شماره تلفن	کد نماینده
تاریخ شروع	شماره تلفن	نمره	
تاریخ اتمام			

اگر بخواهیم در این نمودار شماره تلفن نماینده را نیز نشان دهیم. شماره تلفن هم به کد نماینده و هم به کد دفتری که در آن مستقر است وابسته می‌باشد پس نمودار را به صورت شکل ۱۱-۱۱ قسمت الف نشان می‌دهیم.

اما شکل ۱۱-۱۱ قسمت الف درست نیست زیرا هر نماینده در دفتری مستقر است و شماره تلفن او چیزی جز شماره دفتر نیست پس به طور کلی می‌توان گفت هر گاه در ترسیم نمودار وابستگی به یک مثلث و یا یک چند ضلعی بسته برخورد کنیم این نمودار نشان می‌دهد که وابستگی غیرمستقیم موجود است که باید به وابستگی مستقیم تبدیل گردد پس نمودار وابستگی به صورت شکل ۱۱-۱۱ قسمت ب تبدیل می‌شود. به عنوان مثال دوم فرم ۱۲-۱۱ را در نظر بگیرید در این جا فرم صفحهٔ ورود اطلاعات رزرو مشخص شده است برای رسم نمودار وابستگی دو مطلب را در نظر می‌گیریم.

۱. فرم ۱۲-۱۱ اطلاعات رزرو را نشان می‌دهد پس ارتباط مشتری با متل مربوطه را نیز بایستی نشان دهد ولی در فرم هیچ یک از اقلام اطلاعاتی این ارتباط را مشخص نمی‌کند دلیل آن نمایش داده نشدن شماره رزرو می‌باشد که به صورت اتوماتیک توسط سیستم تولید می‌گردد و در رسم نمودار باید در نظر گرفته شود.



شکل ۱۱.۱۰ نمودار وابستگی گزارش «لیست شرکت‌کنندگان در دوره آموزشی»

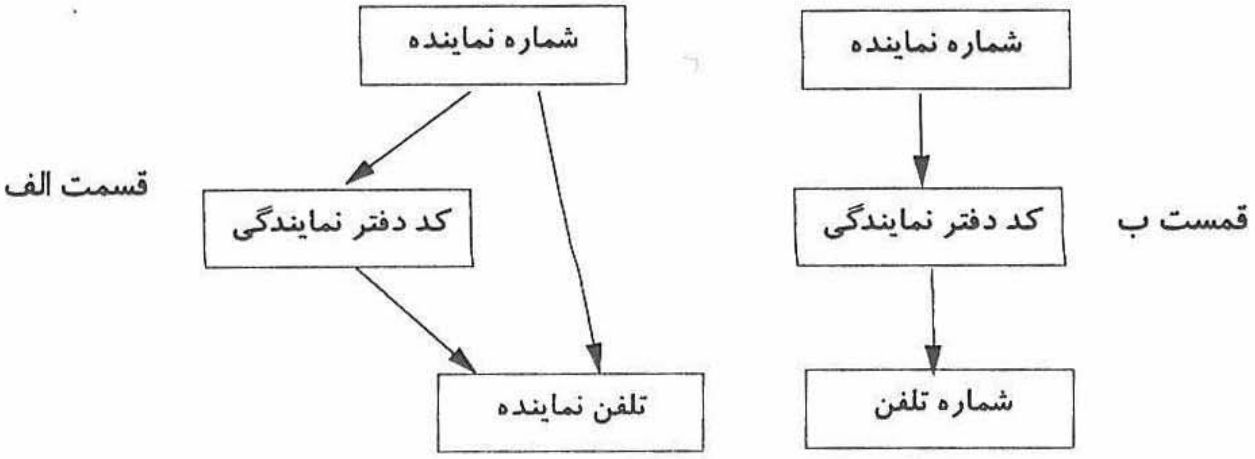
سیستم مکانیزه رزرواسیون

لیست شرکت کنندگان در دوره: مدیریت متل

تاریخ شروع دوره: **تاریخ اتمام دوره:**

کد نماینده	نام نماینده	نمره	کددفتر	تلفن دفتر
۱۱	حسن دیلمقانی	الف	۱۱	۱۱۱۲۱۳
۱۲	حسین رمضانی	ب	۱۵	۱۵۱۵۱۵

۱۱.۹ لیست شرکت‌کنندگان در دوره آموزشی



۱۱.۱۱ طریقه نمایش ارتباط قلم اطلاعاتی شماره نماینده با تلفن نماینده

۲. برای هر یک از افراد همراه یک کلید در نظر می‌گیریم که همان شماره ردیف می‌تواند باشد با در نظر گرفتن مطالب فوق جداول زیر را شناسایی می‌کنیم.

جدول‌مشتریان	جدول رزروشن	جدول همراهان	جدول متل‌ها
کد مشتری	**شماره رزرو**	**کد رزرو**	**کد متل**
نام خانوادگی	کد مشتری	شماره ردیف	نام متل
نام	تاریخ آغاز	نام	
آدرس	تارخ خاتمه	نام خانوادگی	
جنسیت	کد شرایط ویژه	جنس	
کدپستی	هزینه	سن	
شماره‌تلفن	کد متل		

شکل ۱۱.۱۴ نمودار وابستگی را نشان می‌دهد

رزرو مشتریان

کدمشتری ۱۱۱۲ **جنسیت** مرد **نام** حسن

نام خانوادگی رمضانی **آدرس**

کد متل ۱۱ **نام متل**

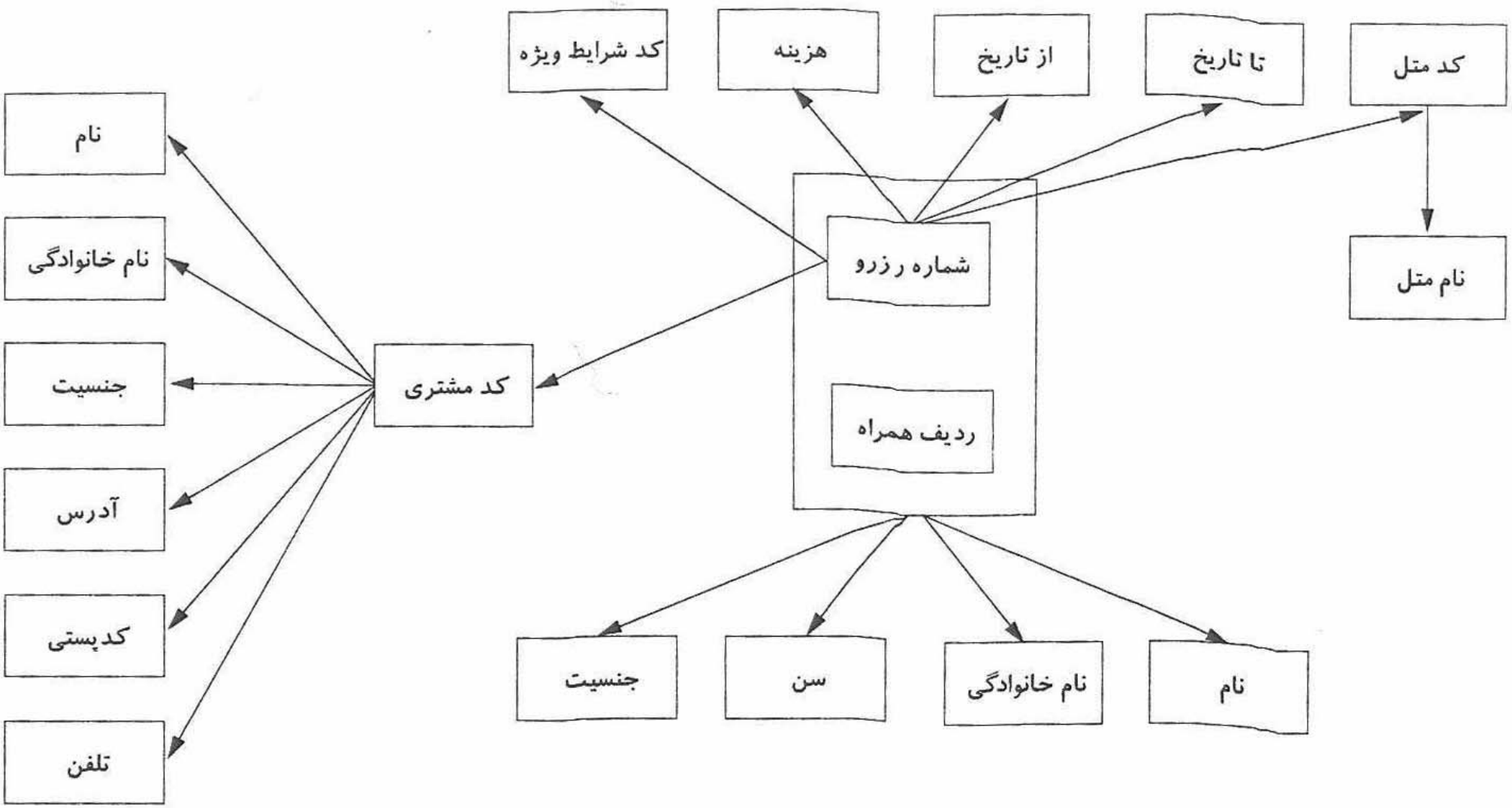
از تاریخ **کد پستی** **هزینه**

تا تاریخ **تلفن**

لیست همراهان

نام **نام خانوادگی** **جنسیت** **سن**

شکل ۱۱.۱۲ فرم رزرو مشتریان


کد شرایط ویژه
هزینه
از تاریخ
تا تاریخ
کد متل
نام
نام خانوادگی
جنسیت
آدرس
کدپستی
تلفن
کد مشتری
شماره رزرو
ردیف همراه
نام متل
جنسیت
سن
نام خانوادگی
نام

شکل ۱۱.۱۳

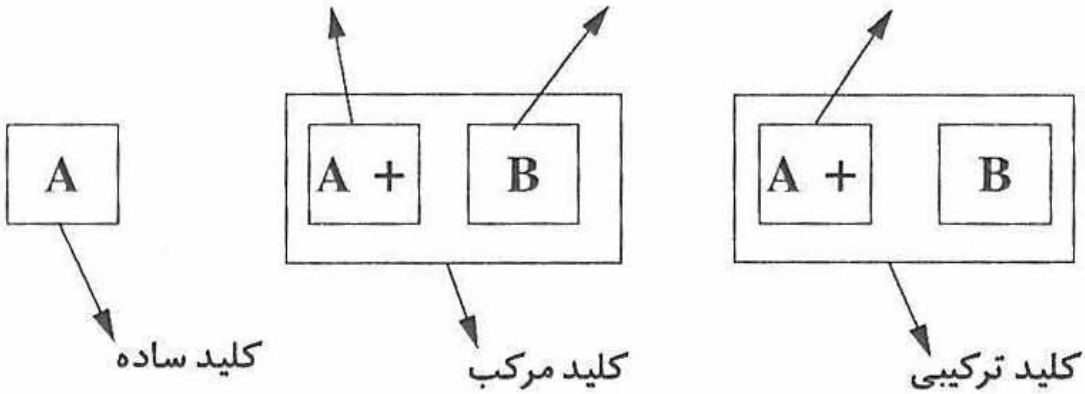
سایر ملاحظات

در نمودار وابستگی تمامی خطوط وابستگی از کلیدها به طرف سایر اقلام اطلاعاتی رسم می‌گردد بنابراین مبدأ خطوط وابستگی کلیدها می‌باشند.

در مورد کلیدهای ترکیبی خطوط وابستگی از اقلامی که در جدول و یا جداول دیگری کلید نیستند خطوط وابستگی رسم نمی‌گردد و اگر در مورد چنین جداولی بخواهیم خطوط وابستگی را ترسیم کنیم از مستطیل بزرگ‌تر که تمام اقلام اطلاعاتی کلید را دربرمی‌گیرد استفاده می‌کنیم.

در مورد کلیدهای مرکب هم همین‌طور برای ترسیم خطوط وابستگی بین کلید و سایر اقلام اطلاعاتی از مستطیل بزرگ‌تری که شامل کلید اقلام اطلاعاتی کلید می‌باشد استفاده می‌شود.

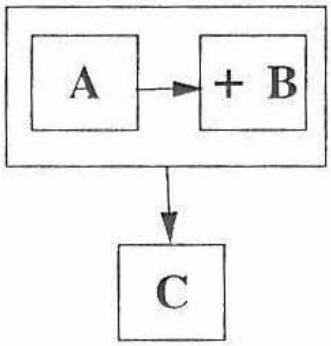
به شکل ۱۵-۱۱ توجه کنید:



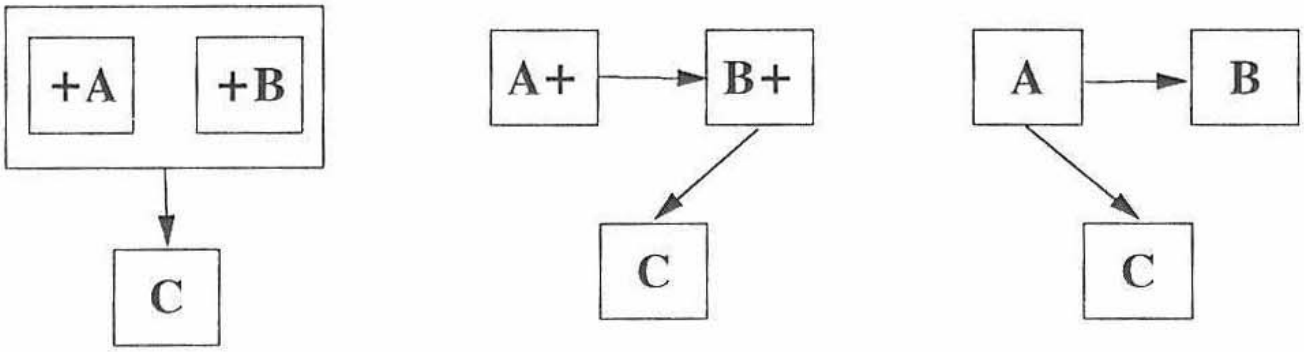
شکل ۱۱.۱۴

شکل ۱۱.۱۵ چگونگی رسم نمودار وابستگی در مورد کلیدهای مختلف

وضعیت شکل ۱۶-۱۱ را در نظر بگیرید این شکل نشان می‌دهد که B وابسته به A و C وابسته به هر دو می‌باشد و یا به عبارتی از A می‌توان به B و بعد به C رسید. این وضعیت آن طورکه در شکل ۱۶-۱۱ نشان داده شده است درست نیست. چنین وضعیتی را می‌توان به یکی از صورت‌های شکل ۱۷-۱۱ نمایش داد.



شکل ۱۱.۱۶ ترسیم نادرست خطوط وابستگی



شکل ۱۱.۱۷ نمایش صحیح خطوط وابستگی

نکته دیگری که به آن باید توجه شود طرز رسم خطوط وابستگی در مورد اقلام اطلاعاتی خاص از قبیل سرجمع‌ها می‌باشد. فرض کنید که می‌خواهیم در یک سیستم فروش سرجمع مبلغ سفارشات هر ناحیه را به تفکیک کالاها و بازاریاب مربوطه داشته باشیم، در این صورت مطابق سایر اقلام اطلاعاتی این اقلام نیز به کلید جدول وابسته می‌باشند.

منطقی‌سازی

پس از پایان بررسی کلیه فرم‌ها و گزارشات و ساخت جداول مربوطه و تبدیل آنها به فرم سوم نرمال به بررسی اقلام اطلاعاتی و حذف موجودیت‌ها و یا اقلام اطلاعاتی اضافی می‌پردازیم. به طور مثال اگر دو جدول زیر که یکی اطلاعات مشتری و دیگری آدرس او می‌باشد را در نظر بگیرید

جدول مشتری

کدمشتری
نام مشتری
آدرس
کدپستی
شماره تلفن

جدول آدرس

کد مشتری
نام مشتری
آدرس مشتری
وضعیت مشتری

می‌توانیم این دو جدول را با یکدیگر ادغام کرده و یک جدول به صورت زیر داشته باشیم:

اطلاعات مشتریان

کد مشتری

نام مشتری

آدرس

کدپستی

شماره تلفن

وضعیت آدرس

ایجاد ساختار منطقی داده‌ها

برای ترسیم ساختار منطقی داده‌ها براساس جداول ایجاد شده اقدامات زیر را انجام می‌دهیم:

۱. بررسی و شناخت کلیه اقلام اطلاعاتی که به صورت کلید خارجی هستند.

۲. تخصیص هر جدول به یک موجودیت.

۳. شناسایی روابط موجودیت‌ها به صورت زیر:

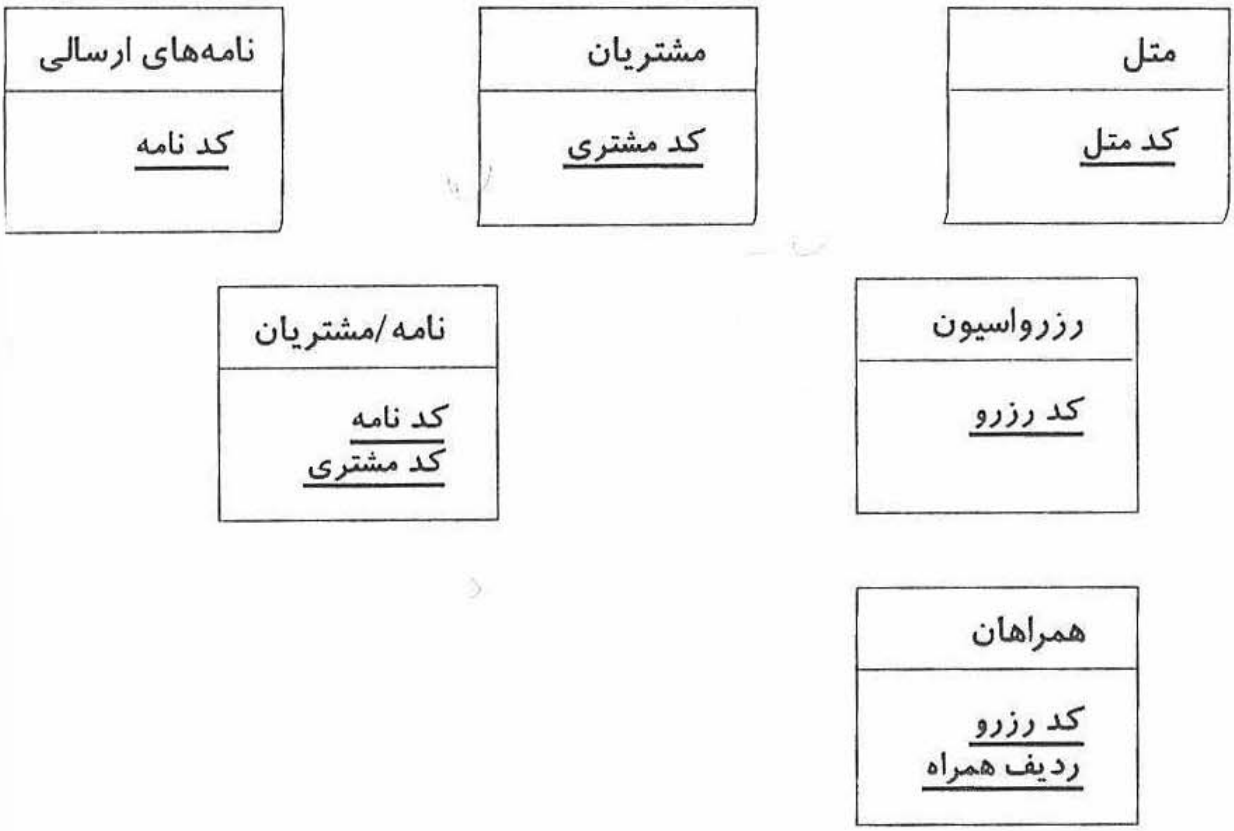
- تمام موجودیت‌هایی که کلید مرکب دارند وابسته (detail) موجودیت‌هایی هستند که کلیدشان یکی از اقلام اطلاعاتی کلید در این موجودیت می‌باشد.
- تمامی موجودیت‌هایی که اقلام اطلاعاتی آنها کلید خارجی هستند وابسته به (detail) موجودیت‌هایی هستند که کلید آنها کلید خارجی در این موجودیت است.
- تمام موجودیت‌هایی که کلید ترکیبی دارند وابسته (detail) موجودیت‌هایی هستند که کلید آنها قسمت اصلی کلید در این موجودیت است.

برای روشن شدن مطلب مثال جداول زیر را درنظر بگیرید:

جدول مشتریان	جدول متل‌ها	جدول رزرواسیون
کد مشتری	**کد متل**	**کد رزرو**
نام	نام متل	کد مشتری
نام خانوادگی		کد متل
آدرس		از تاریخ
کد پستی		تا تاریخ
تلفن		کد شرایط ویژه
		هزینه

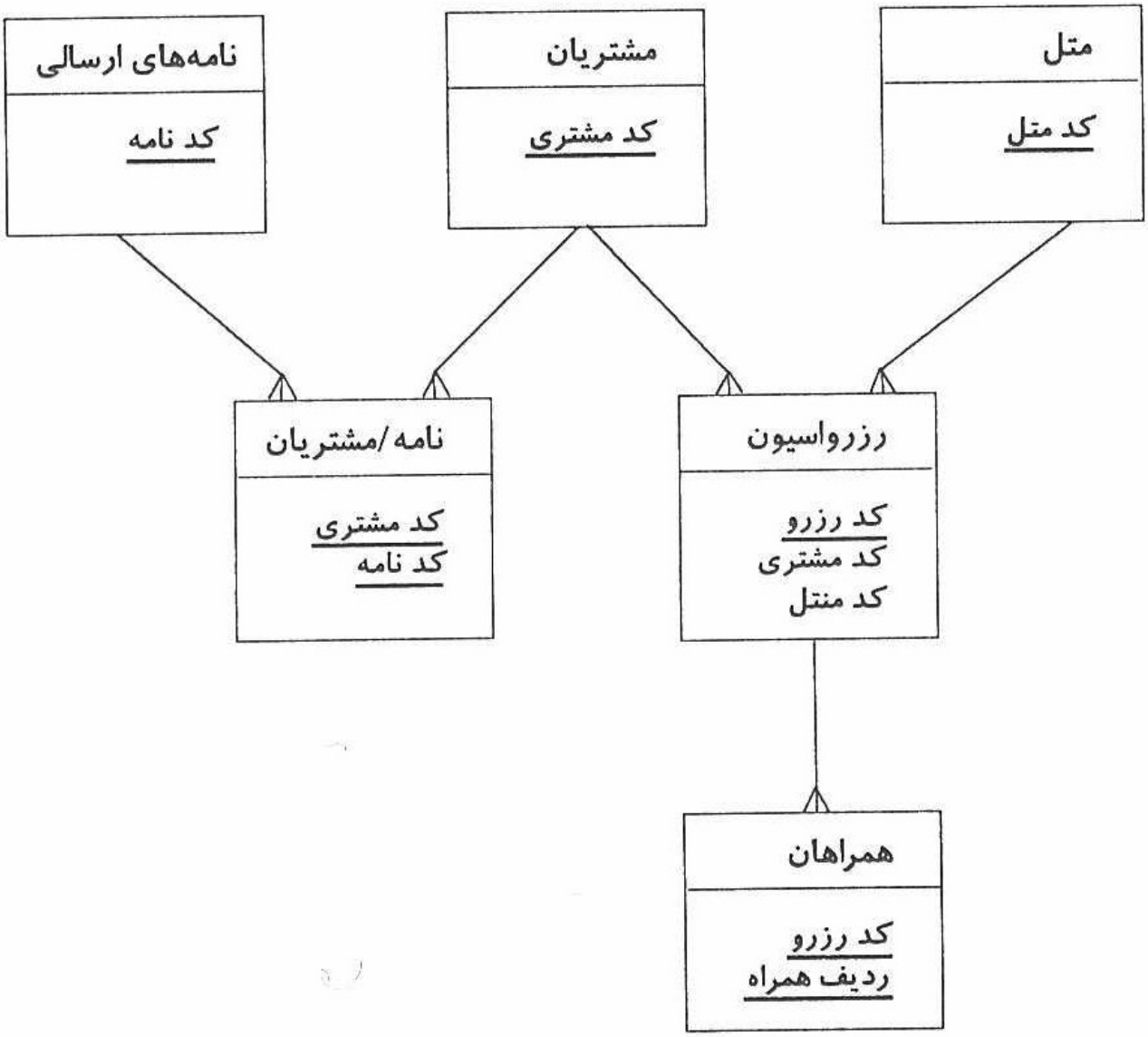
جدول نامه‌ای مشتری	جدول نامه‌های ارسالی	جدول همراهان
کد نامه	**کد نامه**	**کد رزرو**
کد مشتری	تاریخ	ردیف همراه
	عنوان نامه	نام
		جنس
		سن

با درنظر گرفتن جداول فوق موجودیت‌های زیر را شناسایی می‌کنیم.



شکل ۱۱.۱۸ موجودیت‌های به دست آمده در مرحله منطقی‌سازی

در این جا موجودیت نامه / مشتری دارای یک کلید مرکب از شماره نامه / شماره مشتری می‌باشد پس وابسته به هر دو می‌باشد و موجودیت رزرو دارای دو کلید خارجی شماره مشتری و کد متل می‌باشد پس وابسته به هر دو است و موجودیت همراهان فقط وابسته به موجودیت رزرو می‌باشد پس می‌توان ساختار زیر را ترسیم کرد:



شکل ۱۱.۱۹ ساختار منطقی داده‌ها (استخراج شده در این گام)

همان‌طور که گفته شد موجودیتی با کلید مرکب وابسته به موجودیت‌هایی است که کلیدهای آنها اقلام اطلاعاتی کلید را در آن موجودیت تشکیل می‌دهند حال اگر چنین موجودیتی وجود نداشت بایستی ایجاد شود به طور مثال اگر موجودیت نامه ارسالی وجود نداشت چون در موجودیت نامه / مشتری قلم اطلاعاتی شماره نامه وجود دارد و کلید است.

مقایسه دو ساختار

آخرین کار در این گام مقایسه ساختار منطقی داده‌های ایجاد شده در گام‌های قبلی و ساختار ایجاد شده در این گام می‌باشد. قبل از این که ساختار قبلی را عوض کنیم و بعضی از موجودیت‌ها را در آن حذف کنیم اقدامات زیر را انجام می‌دهیم.

۱. از درستی کار منطقی‌سازی و این که چیزی فراموش نشده باشد اطمینان حاصل می‌کنیم.
۲. از در نظر گرفتن کلیه فرم‌های ورودی و گزارشات مطمئن می‌شویم.
۳. ساختاری که از روش منطقی‌سازی به دست می‌آید شامل زیرموجودیت‌ها و نحوهٔ ارتباط بین موجودیت‌ها که می‌تواند خط‌چین و یا خط ممتد باشد نخواهد بود.
۴. از نگهداری اطلاعات اضافی و اطلاعاتی که به راحتی می‌تواند از سایر اطلاعات نگهداری شده استخراج شود اجتناب کنیم.

به طور مثال جدول :

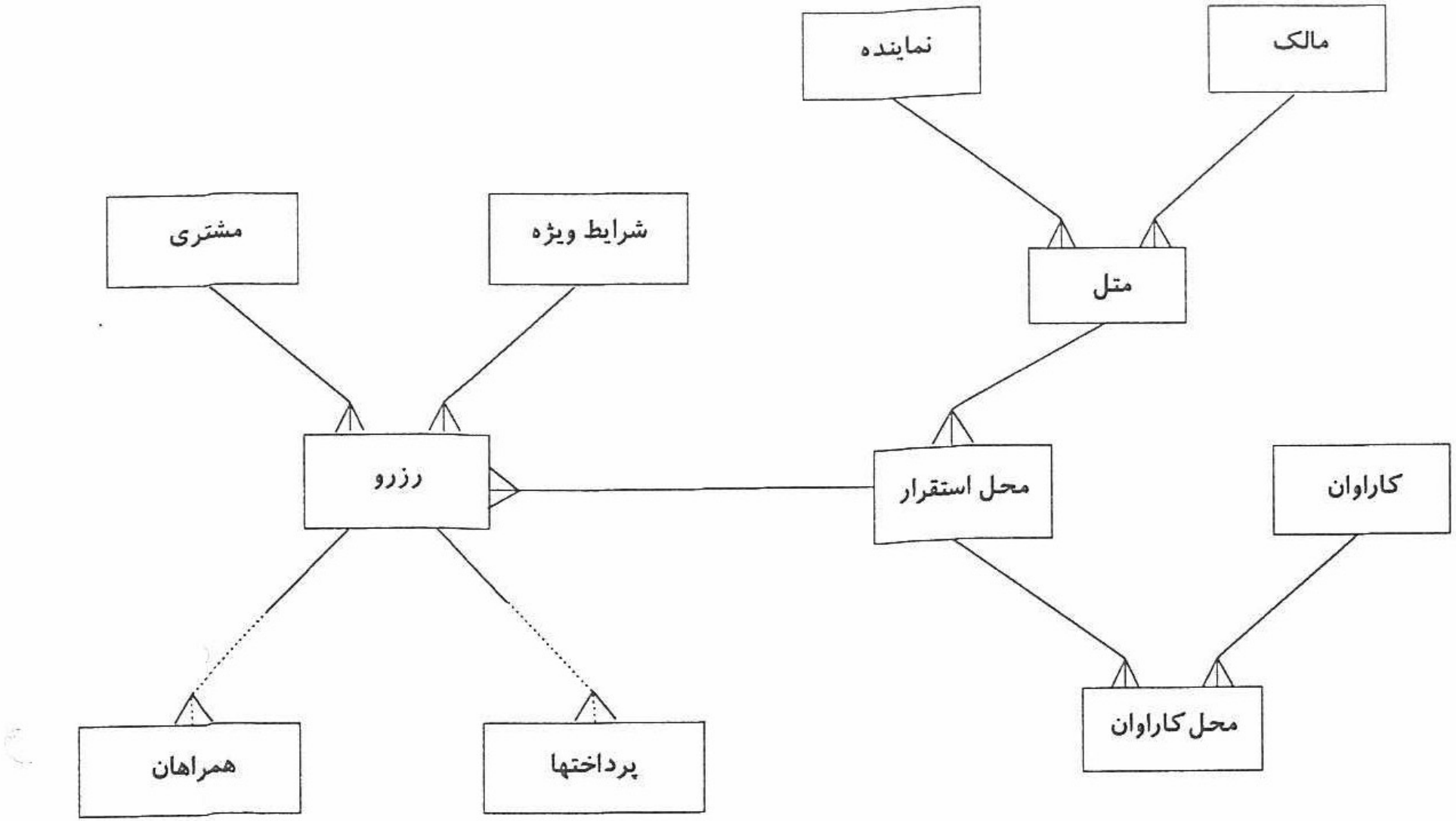
کد بازاریاب
کد محصول
کد منطقه
ماه
تعداد فروخته شده
ارزش اجناس فروش رفته

را در نظر بگیرید این جدول نشان می‌دهد که هر بازاریاب از هر محصول در هر منطقه چه مقدار فروش داشته است. این اطلاعات را به سادگی می‌تواند از اطلاعات فروش ثبت شده در سیستم به دست آورد و احتیاجی به ثبت و نگهداری و به روزرسانی آن نیست. پس از بررسی مدل ایجاد شده در این گام و ساختار منطقی می‌توان به نتایج زیر رسید:

۱. در ساختار جدید به دست آمده موجودیت رزرو وابسته به موجودیت متل می‌باشد. در صورتی که در مدل قبلی موجودیت رزرو وابسته به موجودیت محل استقرار بود. علت این اختلاف نبودن اطلاعات محل استقرار در فرم رزرو می‌باشد که در صفحات بعدی فرم رزرو اضافه می‌گردد و در این جا نادیده گرفته

شده است. پس بهتر است که ساختار قبلی را درنظر بگیریم یا این که بعداً توسط فرم‌های دیگری اطلاعات محل استقرار وارد شود که در این صورت هم موجودیت متل و هم موجودیت محل استقرار در نظر گرفته می‌شود.

۲. موجودیت نامه / مشتری که برای ارتباط دادن موجودیت نامه و موجودیت مشتریان می‌باشد در مدل قبلی وجود نداشته است. می‌توان براساس لیست نیازمندی‌ها به این مطلب پی ببریم که آیا ثبت سوابق نامه‌های ارسالی مورد نیاز هست یا حیر که اگر مورد نیاز باشد این موجودیت را حفظ کنیم در غیر این صورت ساختار قبلی را حفظ می‌کنیم و این موجودیت را نادیده می‌گیریم.



شکل ۱۱.۲۹ ساختار منطقی داده‌ها

فصل دوازدهم

نمونه‌سازی (Prototyping)

اهداف فصل

هدف از نمونه‌سازی، ساختن یک مدل کوچکتر از سیستم است تا قابلیت‌ها و امکانات آن به کاربران نشان داده شود و اگر اشتباهاتی در طراحی سیستم صورت گرفته است قبل از شروع به کار پیاده‌سازی و در مرحله طراحی مشخص شده و اصلاح گردد.

انواع نمونه‌سازی

مدل کوچکتر سیستم را می‌توان برای مقاصد گوناگونی ساخت و به کار برد. در ذیل انواع نمونه که برای کاربردهای متفاوتی ساخته می‌شوند، تعریف شده است.

۱. نمونه‌سازی تجربی

در این جا هدف ساختن مدلی از کاری است که می‌خواهد انجام شود. با بررسی مدل می‌توانیم به این نتیجه برسیم، که آیا اصولاً این کار عملی است یا خیر. به طور مثال اگر بخواهیم اطلاعات یک سیستم فروش را از کامپیوتر شخصی به روی کامپیوتر بزرگ منتقل کنیم با ساخت یک مدل با اطلاعات کمتر و غیرواقعی می‌توان این کار را تجربه کرد و مزایا و معایب آن را مشخص نمود.

۲. نمونه‌سازی اختصاصی

در این نوع نمونه‌سازی هدف ساختن مدلی از سیستم جدیدی می‌باشد که بتوان قابلیت‌ها و امکانات آن را به کاربران نشان داد تا کاربران از درستی طراحی سیستم و مطابقت آن با نیازهای خود اطمینان حاصل کنند.

۳. نمونه‌سازی گام به گام

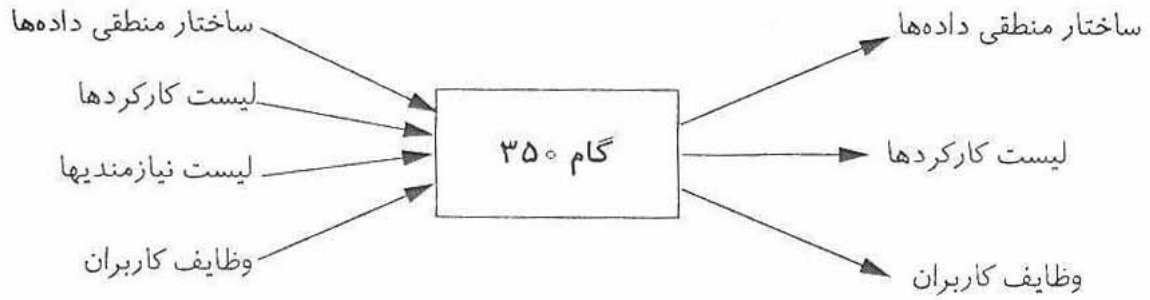
در این نوع نمونه‌سازی تجزیه و تحلیل سیستم در ابتدا صورت گرفته است ولی سیستم به صورت گام به گام پیاده‌سازی می‌شود و هر بار قسمتی از کل سیستم پیاده‌سازی می‌گردد تا در پایان، سیستم تکمیل شود. در حین پیاده‌سازی انجام اصلاحات کوچک امکان‌پذیر است. ولی تغییرات اساسی امکان‌پذیر نیست زیرا سیستم از قبل طراحی شده است.

نمونه‌سازی تکاملی

این نوع نمونه‌سازی با روش گام به گام تفاوت دارد به این صورت که تجزیه و تحلیل

سیستم اساساً صورت نگرفته است ولی با تکامل سیستم پیش می‌رود در این رویکرد ابتدا تعدادی از نیازمندی‌ها تجزیه و تحلیل شده سپس برای همان نیازمندی‌ها طراحی صورت می‌گیرد و سپس برنامه‌سازی می‌شود و پس از تست تعداد دیگری از نیازمندی‌ها بررسی می‌گردد و این چرخه تا به آخر ادامه پیدا می‌کند تا سیستم کامل شود.

این چهار نوع نمونه‌سازی را می‌توان از دیدگاه دیگری نیز بررسی کرد. در این دیدگاه ۲ نوع اول صرفاً جهت نمایش به کاربر درنظر گرفته شده و پس از نمایش به کاربران دیگر مورد استفاده قرار نمی‌گیرد و فی‌الواقع به دور افکنده می‌شود اما بعضی از تحلیل‌گران از نمونه‌سازی اختصاصی به عنوان زیربنای ساخت سیستم استفاده می‌کنند تا باعث تسریع در ایجاد سیستم گردند.



شکل ۱۲.۱ ورودی‌ها و خروجی‌های گام ۳۵۰

جایگاه نمونه‌سازی در SSADM

بهترین هنگام استفاده از نمونه‌سازی در SSADM گام‌های ۱۲۰ و ۱۶۰ برای تعیین نیازمندی‌ها و گام ۳۵۰ برای تأیید نیازمندی‌های شناسایی توسط کاربران می‌باشد. در این گام پس از ساخت یک نمونه از پایگاه داده‌ها مطابق ساختار منطقی داده‌ها، حقوق کاربران و وظایف آنان در رابطه با سیستم جدید و کارکردهایی که در سیستم جدید درنظر گرفته شده به کاربران نمایش داده شده و اگر کمبودهایی وجود داشته باشد این نکات به عنوان نیازمندی‌های جدید در لیست کارکردها و لیست نیازمندی‌ها منعکس می‌گردد و اگر در پاره‌ای از موارد تحلیل‌گر در طراحی سیستم و جهت‌گیری آن مرتکب اشتباه شده باشد کل کار تجزیه و تحلیل از ابتدا مجدداً صورت می‌گیرد.

مسأله دیگری که باید به آن اشاره شود اختیاری بودن و اجباری نبودن این گام می‌باشد که البته امروزه با وجود زبان‌های نسل چهارم این کار به سادگی امکان‌پذیر

می‌باشد و فواید زیادی نیز دارد.

تعیین محدودهٔ نمونه‌سازی

معمولاً مدیریت تعیین می‌کند چه فرم‌های ورودی و گزارشاتی در این مرحله مورد استفاده قرار گیرد و نمونه آن ساخته شود می‌توان در مورد فرم‌های ورودی و گزارشات انتخابی با کاربران نیز مشورت کرد تا اگر گزارشات و فرم‌های دیگری مورد نظر آنها باشد در طراحی و ساخت نمونه مورد استفاده قرار گیرند تنها مسأله محدودکننده در این قسمت زمان و هزینه می‌باشد. البته نمونه گزارشاتی که طبق فرمت تعیین شده برای سازمان‌های دیگری ایجاد می‌شود خصوصاً باید ایجاد گردد.

در هنگام نمایش نمونه تمام اقلام اطلاعاتی موردنظر کاربران با ساختار منطقی داده‌ها و اقلام اطلاعاتی موجودیت‌ها مطابقت می‌گردد و اگر اطلاعاتی با استفاده از فرمول خاص براساس این اقلام به وجود می‌آید آن فرمول در شرح و وظایف کارکرد مربوطه ثبت می‌گردد.

مزایا و معایب نمونه‌سازی

ایجاد نمونه سیستم جدید باعث می‌گردد تا کاربران بدانند سیستم جدید به چه صورت خواهد بود و چه کارهایی انجام می‌دهد تا نواقص آن قبل از این که وارد مرحله پیاده‌سازی شود شناخته و برطرف گردد. به علاوه مشارکت کاربران باعث ایجاد احساس تعهد در آنان نسبت به سیستم می‌گردد اما چون معمولاً نمونه بسیار سریع ایجاد می‌شود و در اختیار کاربران قرار می‌گیرد این تصور غلط در کاربران به وجود می‌آید که سیستم اصلی نیز به همین سرعت حاضر خواهد شد که می‌توان با گفت‌وگو و بیان مشکلات چنین تصوری را از بین برد.

ایجاد مستندات

در راهنمای SSADM ایجاد پنج نوع مستندات مختلف برای این مرحله پیش‌بینی گردیده که بسیار زیاد به نظر می‌رسد. آنچه مهم است ثبت نظرات ارائه شده از طرف کاربران می‌باشد که می‌توان این اطلاعات را در فرم مشابه کل ۲-۱۲ ثبت کرد.

فرم بررسی نمونه

سیستم: سیستم رزرواسیون متل　　نمونه شماره ۳　　**تاریخ:** ۸۰/۱۲/۱

نام تحلیل‌گر: حسین رمضانی　　**نام کاربر:** حسین ممقانی

هدف:

نمایش کارکرد شماره ۴۱ کارمند امور مشتریان آقای ممقانی

نتایج:

- ایجاد راهنماجهت قلم اطلاعاتی متل
- تکفکی رنگها در صفحهٔ نمایش

اقدامات لازم:

- ایجاد راهنما
- تفکیک رنگها
- نمایش مجدد فرم به همکاران آقای ممقانی

۲ - ۱۲ فرم بررسی نمونه

فصل سیزدهم

تاریخچه زندگی موجودیت

(Entity Life History)

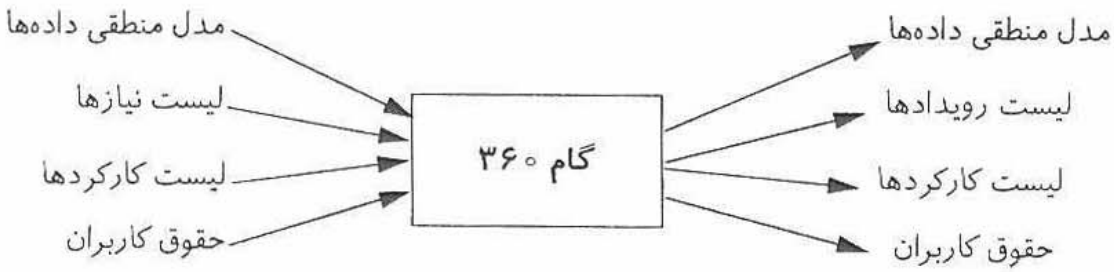
اهداف فصل

هدف اصلی در این گام تعیین کارکردهایی می‌باشد که درنظر گرفته نشده‌اند. به طور کلی می‌توان گفت اکثر تحلیل‌گران به این نتیجه می‌رسند که نتوانسته‌اند بیش از هشتاد درصد کارکردها را تشخیص دهند، مخصوصاً کارکردهایی که در رابطه با حذف نمونه (رکورد) در موجودیت‌های مختلف می‌باشند چون در ابتدا نگرانی تحلیل‌گر ایجاد نمونه و پیدا کردن موجودیت‌های جدید می‌باشد. در این گام می‌خواهیم کارکردهایی که از نظر تحلیل‌گر مخفی مانده‌اند را شناسایی کنیم.

ابزاری که در این گام مورد استفاده واقع می‌شود نمودار تاریخچه زندگی موجودیت (Entity Life History) است.

غالباً گفته می‌شود که این ابزار دیدگاه سومی از سیستم را عرضه می‌کند. دو دیدگاه دیگر نمودار جریان داده‌ها و ساختار منطقی داده‌ها می‌باشد. دیدگاهی که مراحل مختلف زندگی رکوردها یا نمونه‌های (Occurances) یک موجودیت، از زمان ایجاد تا حذف شدن آن را نمایش می‌دهد.

برای رسم نمودار در گام ۳۶۰ باید ورودی‌ها و خروجی‌های گام را شناسایی کنیم. در شکل ۱-۱۳ ورودی‌ها و خروجی‌های این گام نشان داده شده است. مهم‌ترین ورودی مسلماً ساختار منطقی داده‌ها می‌باشد که موجودیت‌ها را نشان می‌دهد و بعد لیست کارکردها می‌باشد. می‌توانیم در شناسایی رویدادها از فهرست و لیست نیازها و نمودار جریان داده‌ها کمک بگیریم. پس از شناسایی رویدادها می‌توانیم تغییرات لازم را در لیست کارکردها و نمودار جریان داده‌ها نیز اعمال کنیم. در این گام فهرست رویدادها نیز ایجاد می‌شود تا بتوان براساس آن ساختار منطقی داده‌ها را اصلاح کرد. رسم نمودار تاریخچه موجودیت (ELH) از کارهای بسیار مهمی است که انجام می‌شود تا زوایای پنهان سیستم را روشن کند. ممکن است به دلیل سختی روش‌ها انجام این کار دشوار به نظر برسد، به همین دلیل بسیاری این گام را نادیده می‌گیرند اما بهتر آن است که به جای حذف این مرحله از روش‌های ساده‌تری استفاده شود.



شکل ۱۳.۱ ورودی‌ها و خروجی‌های گام ۳۶۰

از کجا آغاز کنیم

برای شروع این گام ابتدا رویدادها (Events) را شناسایی می‌کنیم تا براساس آن بتوان نمودار تاریخچه زندگی موجودیت را ترسیم کرد. پس ابتدا رویداد را تعریف می‌کنیم.

- رویداد عبارت است از آنچه باعث ایجاد به روزرسانی و یا حذف رکوردها (Occurance) یک موجودیت می‌شود.

پس برای رسم نمودار تاریخچه زندگی موجودیت رویدادهایی را که بر زندگی موجودیت از ابتدا تا انتها تأثیر می‌گذارد بر حسب توالی آن در نظر می‌گیریم.

ترسیم این نمودار کاملاً اختیاری است و برای موجودیت‌هایی که رویدادهای کمی روی آن تأثیر می‌گذارد رسم آن لزومی ندارد. در ذیل چگونگی رسم نمودار به صورت گام به گام آورده شده است.

۱. شناسایی رویدادها
۲. تکمیل ماتریس موجودیت /کارکرد
۳. ترسیم نمودار
۴. بررسی نمودار
۵. بررسی رویداد حذف و اثر آن بر موجودیت
۶. تعیین کارکردهایی که هر رویداد بر آن تأثیر می‌گذاردو ثبت کارکردهای به دست آمده در لیست کارکردها
۷. ایجاد فهرست رویدادها

رویداد

می‌توان گفت آنچه در خارج از سیستم انجام می‌شود و باعث به هنگام‌سازی (حذف، به روزرسانی، ایجاد) نمونه‌های هر موجودیت می‌شود را رویداد می‌گویند. این

رویدادها معمولاً بر اثر کنش‌های موجودیت خارجی صورت می‌گیرد. در این جا باید به تفاوت رویداد و پردازش توجه کنیم که رویداد باعث شروع پردازش می‌گردد در حالی که پردازش باعث به روزرسانی (ایجاد، حذف، به روز رسانی) می‌شود.

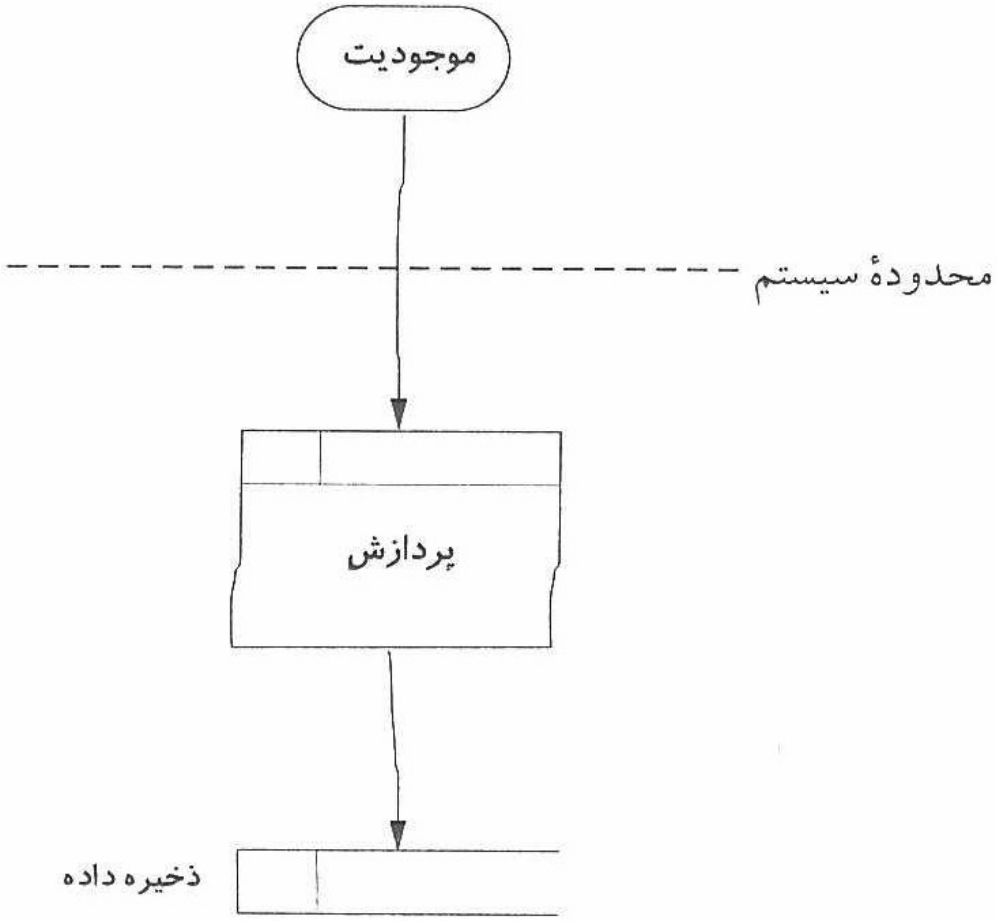
رویدادها را می‌توان به صورت زیر دسته‌بندی کرد:

۱. کلیه رویدادهایی که منشأ خارجی دارند و باعث شروع یک پردازش می‌شوند که در نهایت به بهنگام‌سازی یک ذخیره داده منتهی می‌گردد که عمومی‌ترین شکل رویداد می‌باشد. به طور مثال می‌توان سفارش یک مشتری را در نظر گرفت رویداد «ثبت سفارش مشتری» پردازش «ثبت سفارش مشتری» را شروع کرده و سفارش در یک ذخیره داده ثبت می‌گردد.

۲. رویدادهایی را که با گذشت زمان ایجاد می‌شود و باعث بهنگام‌سازی موجودیت‌ها می‌گردند. به طور مثال یک سیستم کنترل موجودی را در نظر بگیرید که در پایان هر روز به بررسی موجودی کالاها می‌پردازد و برای کالاهایی که موجودی آنها زیر نقطه سفارش می‌باشد درخواست خرید صادر می‌کند و یا رویداد پایان سال مالی که باعث آغاز پردازش‌های مربوطه می‌گردد.

۳. رویدادهایی که منشأ داخلی دارند، به عنوان مثال فرض کنید در یک سیستم مدیریت کالا پس از ورود اطلاعات رسید کالا سند حسابداری آن نیز ثبت می‌گردد، بنابراین ثبت سند حسابداری یک رویداد داخلی می‌باشد.

رویدادها به تنهایی انجام نمی‌شوند و هر رویداد با موجودیت مربوطه ارتباط دارد و بر آن تأثیر می‌گذارد که در این جا مقصود از اثر رویداد بر موجودیت آن است که هر رویداد می‌تواند باعث ایجاد، حذف به روزرسانی نمونه‌های یک موجودیت گردد.



شکل ۱۳.۲ نحوه تأثیرگذاری موجودیت خارجی بر پردازش‌های سیستم (رویداد)

شناسایی رویدادها

چگونه می‌توان رویدادها را شناسایی کرد. اولین راهی که به ذهن می‌رسد استفاده از نمودار جریان داده‌ها در آخرین سطح می‌باشد. می‌توان پردازش‌هایی را که موجب به روزرسانی موجودیت‌ها می‌شود را شناسایی کرد و سپس با دنبال کردن آنها به این مطلب رسید که بروز چه رویدادهایی باعث آغاز این پردازش می‌گردد اما از این جا شروع نمی‌کنیم. زیرا اولاً مقصود از نمودار تاریخچه زندگی، مشخص کردن پردازش‌هایی است که دیده نشده‌اند. ثانیاً پردازش‌ها در نمودار جریان داده‌ها به صورت کلی شرح داده شده‌اند و تاریخچه زندگی موجودیت به جزئیات به روزآوری می‌پردازد. ثالثاً استفاده از منابع جدید باعث ایجاد سؤالات جدید نسبت به سیستم می‌گردد، پس با توجه به مطالب بالا باید از جای دیگری آغاز کنیم.

ساختار منطقی داده‌ها می‌تواند اطلاعات مناسبی برای شناسایی رویدادها در اختیار بگذارد و برای هر موجودیت پرسش‌های زیر را در نظر می‌گیریم:

۱. چه رویدادهایی باعث ایجاد نمونه‌های جدید موجودیت می‌شود.

۲. چه رویدادهایی باعث حذف آن می‌گردد. براساس مذاکره با کاربران به این نکته رسیده‌ایم که تمام اطلاعات رزرو تا سه سال در سیستم باقی بماند پس حذف یک نمونه رزرو سه سال پس از تاریخ رزرو صورت می‌گیرد.

۳. چه رویدادهایی باعث به روزآوری موجودیت می‌گردد. در این جا به اقلام اطلاعاتی موجودیت رزرو توجه می‌کنیم و فرض می‌کنیم که اقلام اطلاعاتی تاریخ شروع و تاریخ خاتمه و نام متل و محل استقرار آن تغییر نکند اگر این اقلام تغییر نکنند پس هزینه رزرو نیز تغییر نمی‌کند.

شکل ۴-۱۳ اقلام اطلاعاتی موجودیت را نمایش می‌دهد تنها قلم اطلاعاتی که می‌تواند تغییر کند Status یا وضعیت می‌باشد که می‌تواند برحسب مورد یکی از وضعیت‌های زیر را نمایش دهد:

P رزرو موقت

R پذیرفته نشد (رد شد)

F پرداخت کامل

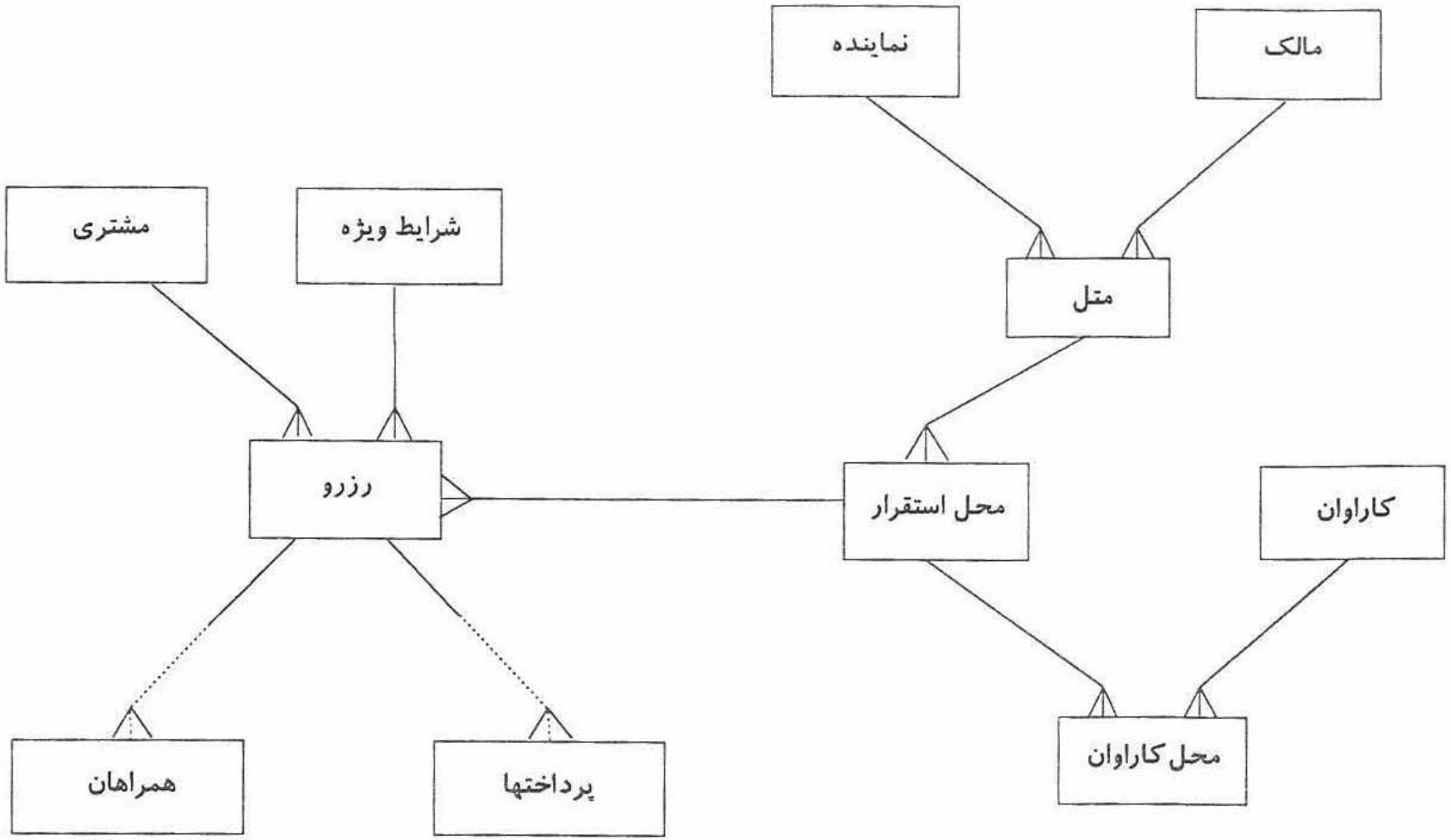
D بیعانه پرداخت شده

N عدم پرداخت (پرداخت صورت نگرفته است)

C ابطال

S گزارش متل

که این قلم اطلاعاتی نشان‌دهنده موقعیت رزرو می‌باشد و چگونگی آن را نشان می‌دهد.



شکل ۱۳.۳ ساختار منطقی سیستم رزرواسیون متل

اقلام اطلاعاتی موجودیت		
نام سیستم: سیستم رزرواسیون مکانیزه	**وضعیت موجود:**	**وضعیت پیشنهادی:**
نام موجودیت: رزرو	**تاریخ**	
	تعداد	

کلید	نام قلم اطلاعاتی	فرمت
P	شماره رزرو	عددی
F	شماره مشتری	عددی
F	کد شرایط ویژه	کاراکتری (۲)
F	کد متل	کاراکتری (۳)
	محل استقرار	کاراکتری (۴)
	تاریخ آغاز	تاریخ
	تاریخ خاتمه	تاریخ
	هزینه	کاراکتری (۴)
	وضعیت	کاراکتری (۱)

شکل ۱۳.۴ اقلام اطلاعاتی موجودیت رزرو

وضعیت ابطال در صورتی پیش می‌آید که اگر تا هفت روز پس از تاریخ پرداخت نهایی یا هفت روز پس از تاریخ پرداخت بیعانه پرداخت دیگری توسط مشتری صورت نگیرد رزرو باطل می‌شود پس به ازای هر یک از حالت‌هایی که این قلم اطلاعاتی می‌گیرد، رویداد برای آن موجودیت تعریف می‌شود. رویدادهایی از قبیل پرداخت، ابطال، عدم پرداخت، گزارش متل. حال این کار را برای تمامی موجودیت‌ها تکرار می‌کنیم و یک جدول می‌سازیم که یک بعد آن رویدادها و بعد دیگرش موجودیت‌ها می‌باشد و براساس اثری که روی موجودیت گذاشته می‌شود از حروفی مانند I به معنی ایجاد نمونه و M به معنی به روزآوری و D به معنی حذف استفاده می‌شود. شکل ۵-۱۳ قسمتی از این جدول را برای سیستم رزرواسیون نمایش می‌دهد.

البته بعضی از کتاب‌ها به جزئیات بیشتری می‌پردازد و از حروفی مانند T به معنی ایجاد ارتباط با موجودیت بالاسری یا G به معنی ایجاد ارتباط با موجودیت وابسته یا C

به معنی آزاد شدن از موجودیت بالاسری استفاده می‌کنند و چنین حالت‌هایی را هم نشان می‌دهند. آنچه در جدول مشاهده می‌شود این است که بعضی از رویدادها بر روی یک موجودیت حالت دوگانه دارد و به صورت دوگانه تأثیر می‌گذارد.

رویداد / موجودیت	مشتری	رزرو	همراهان	پرداخت
درخواست رزرو	I/M	I	I	I
تغییر اطلاعات مشتری	M			
۴ سال پس از رزرو	D			
پرداخت		M		M
عدم پرداخت بیعانه		M		
عدم پرداخت نهایی		M		
پرداخت ناکافی		M		M/I
دریافت گزارش متل		M		
۳ سال پس از ورود		D	D	D
اضافه شدن همراهان			I	
حذف افرادی از همراهان			D	
تغییر اطلاعات همراهان			M	
ابطال		M		

شکل ۱۳.۵ ماتریس موجودیت / رویداد (Entity Access Matrix)

اولین موجودیت که دچار این وضعیت می‌باشد موجودیت مشتری است. رویداد رزرو یا یک نمونه در این موجودیت ایجاد می‌نماید یا آن را به روز می‌کند و برای سادگی قضیه فرض می‌کنیم که اپراتور از مشتری سؤال می‌کند که اولین بار است از این طریق متل رزرو می‌کند یا خیر؟ که اگر مشتری بگوید اولین بار است یک نمونه در این موجودیت اضافه شده و در غیر این صورت به روزآوری انجام می‌شود.

موجودیت دومی که دچار این حالت است موجودیت پرداخت است. چون رویداد پرداخت این موجودیت را متأثر می‌کند و بهنگام می‌سازد. در این موجودیت اقلام اطلاعاتی آخرین تاریخ پرداخت بیعانه و آخرین تاریخ پرداخت نهایی و تاریخ پرداخت

بیعانه و تاریخ پرداخت نهایی وجود دارد در ابتدا که رویداد رزرو نمونه مذکور را ساخته دو قلم اطلاعاتی آخرین تاریخ پرداخت بیعانه و تاریخ پرداخت نهایی دیگر پر می‌شود و پس از پرداخت بیعانه و پرداخت نهایی توسط مشتری دو قلم اطلاعاتی به هنگام می‌شوند. حال اگر مشتری مبلغی پرداخت نماید که متناسب با مبلغ نهایی نمی‌باشد در این صورت کارمند مربوطه با مشتری تماس گرفته و از او می‌خواهد که مشخص کند که این پرداخت ناکافی به عنوان پیش‌پرداخت تلقی گردد و مابقی نیز پرداخت می‌شود یا کلاً هزینه مربوطه را به مشتری عودت نماید و رزرو را باطل کند که در صورت اول سیستم یک نمونه دیگر از موجودیت پرداخت ایجاد می‌کند که مبلغ آن درست به اندازه مابه‌التفاوت مبلغ پرداختی و کل هزینه می‌باشد.

بنابراین رویداد پرداخت ناکافی باعث بهنگام‌سازی یک نمونه و ایجاد نمونه دیگر می‌شود. حال اگر دو مورد اخیر را مقایسه کنیم تفاوت‌هایی با یکدیگر دارند در مورد اول این رویداد است که تأثیری دوگانه دارد اما مورد دوم این موجودیت است که دو نقش را ایفا می‌کند.

نمودار تاریخچه زندگی موجودیت

هدف از این کار رسم نموداری است که تصویری از زندگی یک موجودیت را نمایش می‌دهد و از زمان ایجاد تا زمان حذف را دربرمی‌گیرد و این نمودار تمام مراحلی را که طی می‌شود، نمایش می‌دهد این نمودار را می‌توان به روش‌های مختلف رسم کرد که این روش‌ها عباتند از:

۱. نمودار جکسون
۲. نمودار جریان رویدادها
۳. نمودار انتقال حالت
۴. نمودار حالت‌های محدود

روش استاندارد در SSADM روش جکسون می‌باشد بدین لحاظ ابتدا به تشریح این روش می‌پردازیم. سپس روش ساده‌تری را که بیشتر در این کتاب مورد استفاده قرار می‌گیرد تشریح می‌شود:

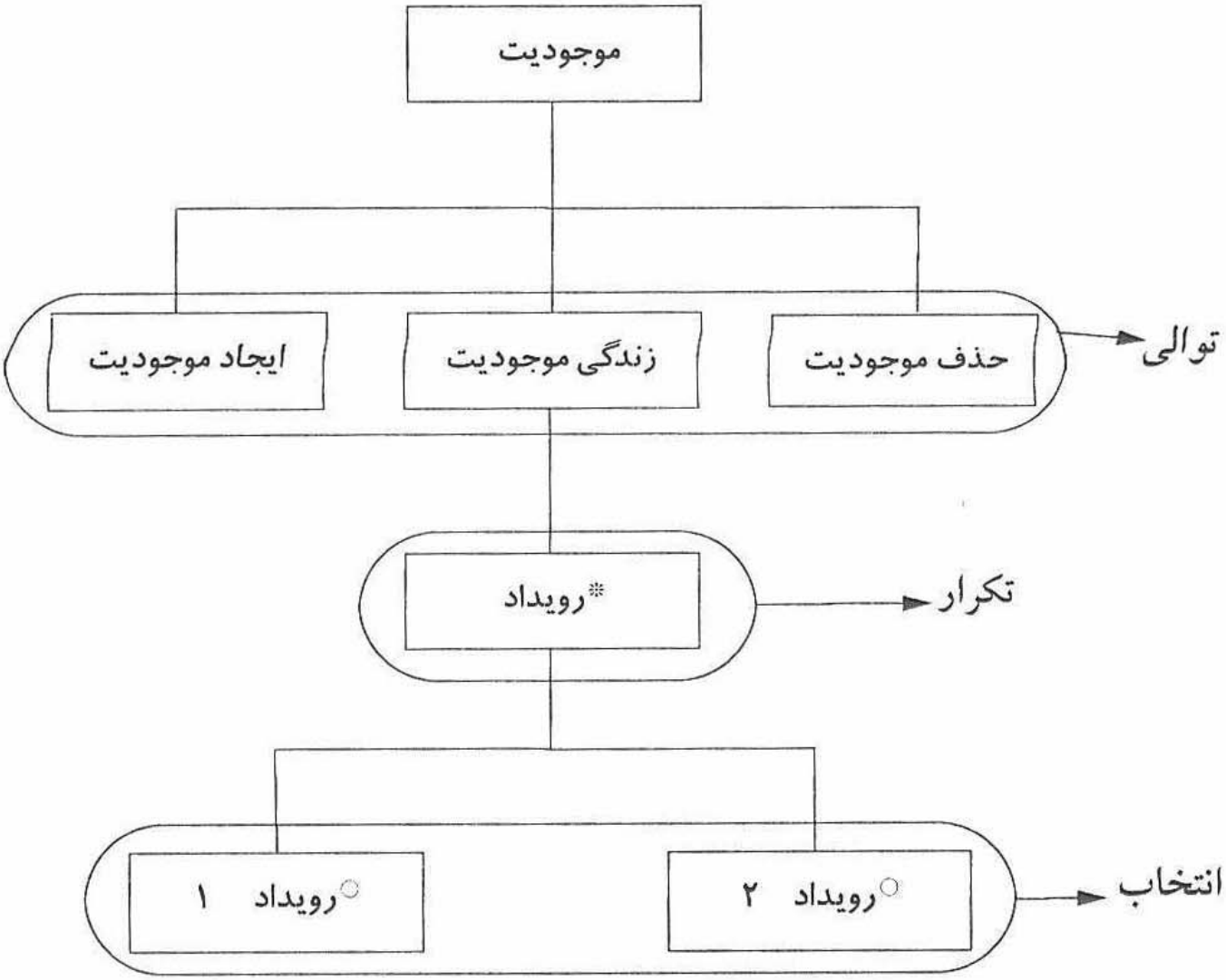
در این روش سه ساختار موجود است:

۱. توالی

۲. انتخاب

۳. تکرار

شکل زیر یک نمودار تاریخچه زندگی است که شامل سه ساختار فوق می‌باشد.



شکل ۱۳۶ نمودار تاریخچه موجودیت رسم شده به روش جکسون

توالی رویدادها توسط مستطیل‌هایی که در یک سطح هستند، نمایش داده می‌شوند و تکرار رویداد توسط مستطیلی که ستاره دارد و انتخاب توسط مستطیل‌هایی که با علامت «○» مشخص شده‌اند.

ساختار از بالا به پایین و از چپ به راست خوانده می‌شود و از ایجاد نمونه تا حذف آن را شامل می‌شود.

ساختار توالی ترتیب رویدادها را نشان می‌دهد. در صورتی که انتخاب نشان می‌دهد که موجودیت یکی از چندین وضع را می‌توان انتخاب کند و تکرار نشان‌دهنده این است که یک رویداد چندین بار می‌تواند تکرار گردد ولی به جای روش جکسون می‌توان از

روش نمودار جریان رویدادها استفاده کرد که فهم آن توسط کاربران راحت‌تر می‌باشد و همین ساختارها را به صورت دیگری نشان می‌دهد شکل ۷-۱۳ نشان‌دهندهٔ همین ساختارها توسط روش جریان داده‌ها می‌باشد.

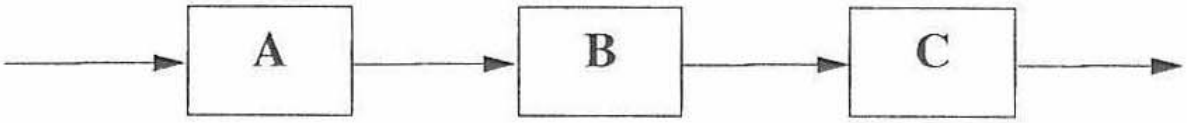
قسمت اول شکل ۷-۱۳ توالی را نشان می‌دهد که نشان‌دهندهٔ آن است که رویداد C بعد از رویداد B و بعد از رویداد A انجام می‌شود و قسمت دوم انتخاب را نشان می‌دهد. که یکی از رویدادهای A یا B یا C انتخاب شده و عمل می‌شوند و قسمت سوم تکرار را نمایش می‌دهد و نشان می‌دهد که یک رویداد چندین بار می‌تواند عمل شود.

نمودار تاریخچه زندگی موجودیت همراهان در شکل ۸-۱۳ نمایش داده شده است. این نمودار براساس جدول موجودیت / رویداد شکل ۵-۱۳ رسم گردیده است همان‌طور که در جدول ملاحظه می‌کنید ایجاد این موجودیت بر اثر دو رویداد «اضافه شدن همراهان» و یا «درخواست رزرو» صورت می‌گیرد و حذف آن نیز بر اثر دو رویداد «حذف همراهان» یا «گذشت ۳ سال از رزرو» صورت می‌گیرد. پس از ایجاد نمونه و قبل از حذف آن نیز رویداد «تغییر مشخصات همراهان» ممکن است چندین بار روی آن اثر گذارد.

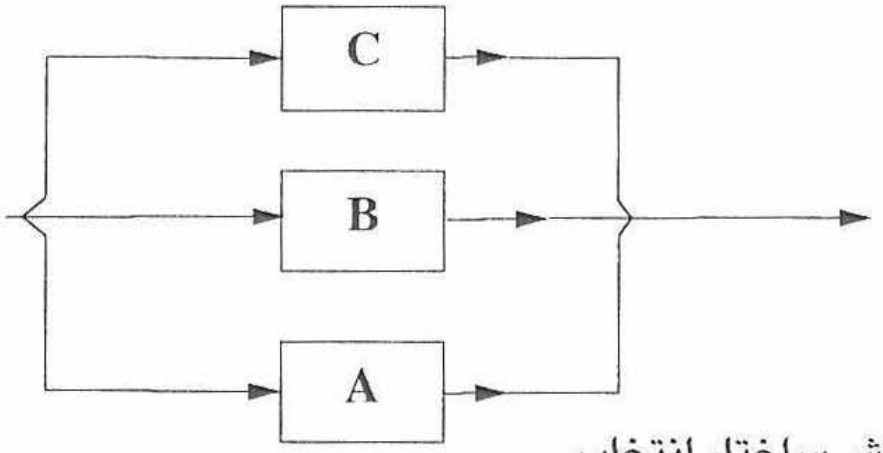
در ترسیم نمودارها بهتر است ابتدا رویدادهایی که باعث ایجاد نمونه می‌شود رسم گردد و سپس رویدادهایی که باعث حذف نمونه می‌گردد و بعد از آن رویدادهایی که بین این دو مرحله تأثیر می‌گذارند و گاهی بسیار پیچیده می‌باشند.

تأثیرات رویداد بر نمونه

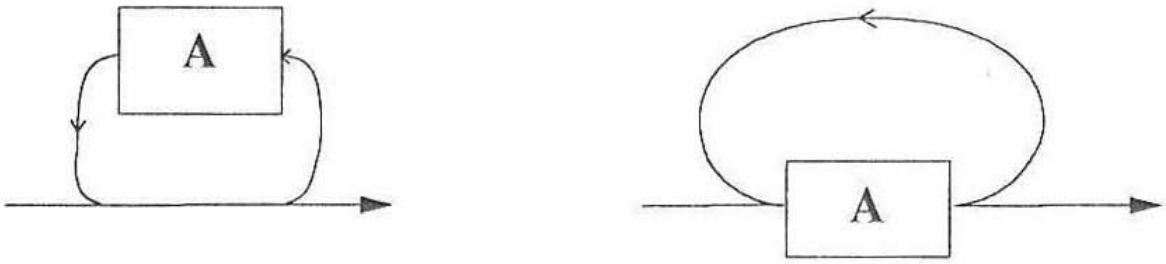
شکل ۹-۱۳ نمودار تاریخچه زندگی موجودیت مشتری را نمایش می‌دهد. چیز جالبی که در این نمودار می‌باشد تکرار رویداد «درخواست رزرو» می‌باشد. وقتی درخواست رزرو برای اولین بار برای یک مشتری می‌آید نمونه مشتری ایجاد می‌گردد با رزروهای بعدی موجودیت مشتری بهنگام می‌شود. البته رویداد «تغییر مشخصات مشتری» نیز این موجودیت را بهنگام می‌سازد.



نمایش ساختار توالی

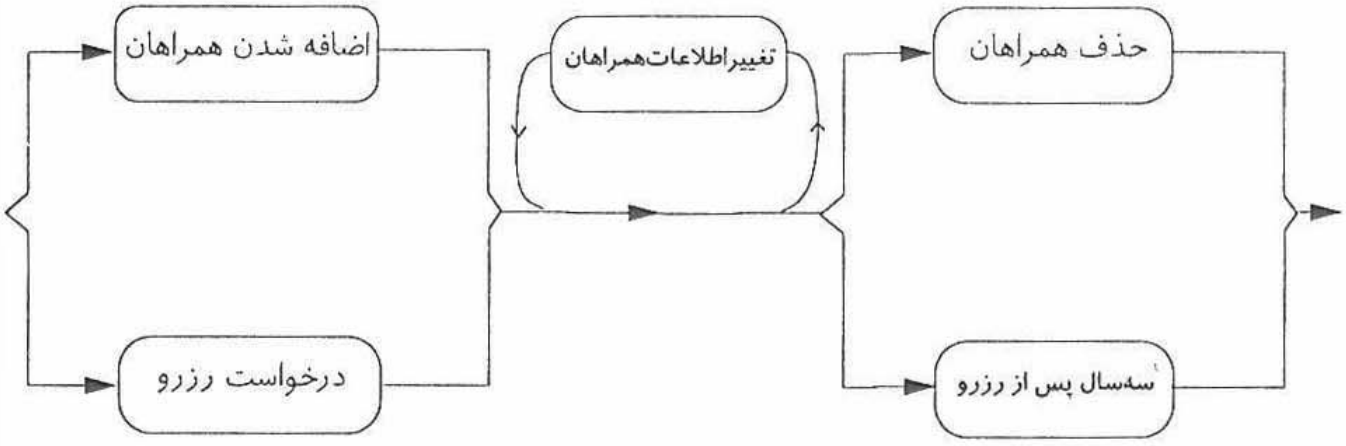


نمایش ساختار انتخاب



نمایش ساختار تکرار

شکل ۱۳.۷ ساختارهای مختلف به روش جریان رویدادها



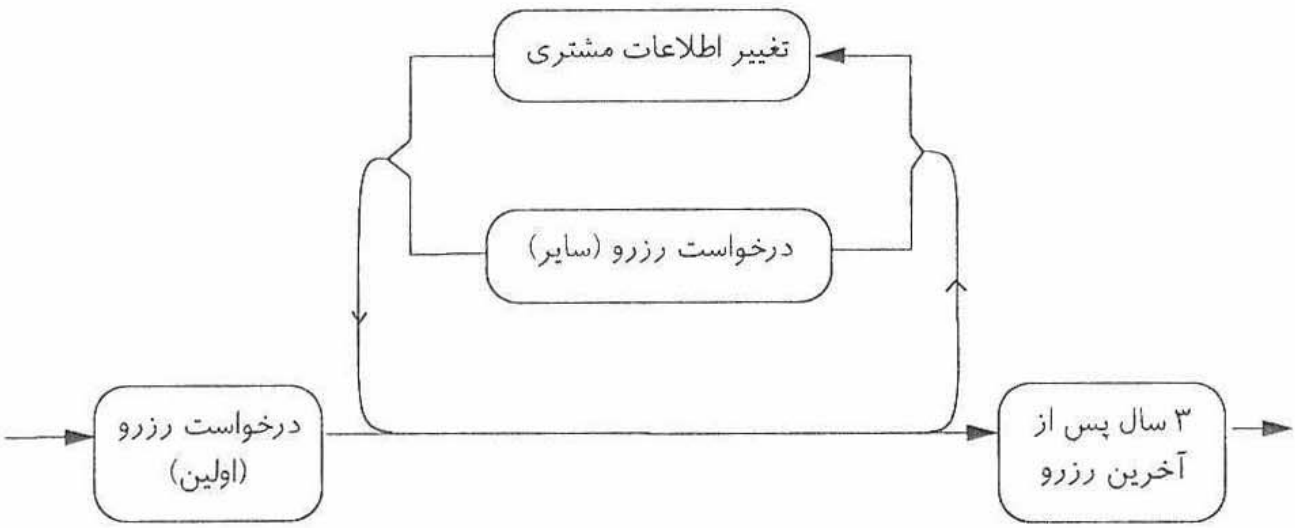
شکل ۱۳.۸ نمودار تاریخچه زندگی موجودیت همراهان (روش جریان رویدادها)

پس می‌توان گفت رویداد «درخواست رزرو» تأثیری دوگانه بر این موجودیت دارد. بار اول آن را ایجاد می‌کند و در سایر موارد آن را بهنگام می‌سازد پس نمودار باید بین اجرای بار اول این رویداد با سایر موارد را تفاوت بگذارد و آن را نشان دهد. در این جا به این صورت عمل شده که در یک پرانتز نوشته شده است اولین بار یا نوشته شده است سایر که بدین ترتیب بین اثر بار اول این رویداد با اثرش در سایر موارد تمایز قایل شده‌ایم.

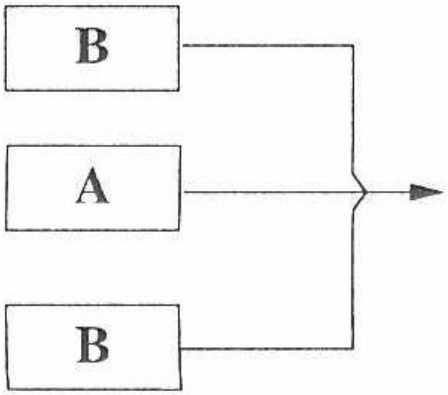
ممکن است راه حل دیگر مسأله آن باشد که به جای این که نمونه‌های موجودیت مشتری به روزرسانی گردد تمام مشتریان را مشتری جدید فرض کنیم و نمودار را بر آن اساس ترسیم کنیم در این صورت احتیاجی به نمایش تکرار این رویداد نخواهد بود نمودار و تاریخچه این موجودیت ساده‌تر خواهد بود ولی اگر احتمال آن وجود داشته باشد که یک مشتری همزمان ۲ درخواست رزرو فعال داشته باشد دیگر نمی‌توان چنین فرض کرد.

اگر نمودار تاریخچه زندگی بسیار درهم و شلوغ باشد استفاده از آن مشکل خواهد بود. یک راه جلوگیری از درهم شدن نمودار نشان دادن یک رویداد در چندین مستطیل است که می‌توان با کشیدن یک خط کوچک در بالای مستطیل تکرار آن را نمایش داد.

می‌توان نمودارها با به شکل دیگری نیز رسم کرد که شاید حتی بعضی از آنها برای نمایش نزد کاربران مناسب‌تر باشند ولی این مسأله می‌تواند باعث تفاوت نمودارهایی که توسط اشخاص متفاوت رسم می‌گردد شود که در این صورت فهم چنین نمودارهایی برای دیگران مشکل خواهد بود.



شکل ۱۳.۹ تاریخچه زندگی موجودیت مشتری

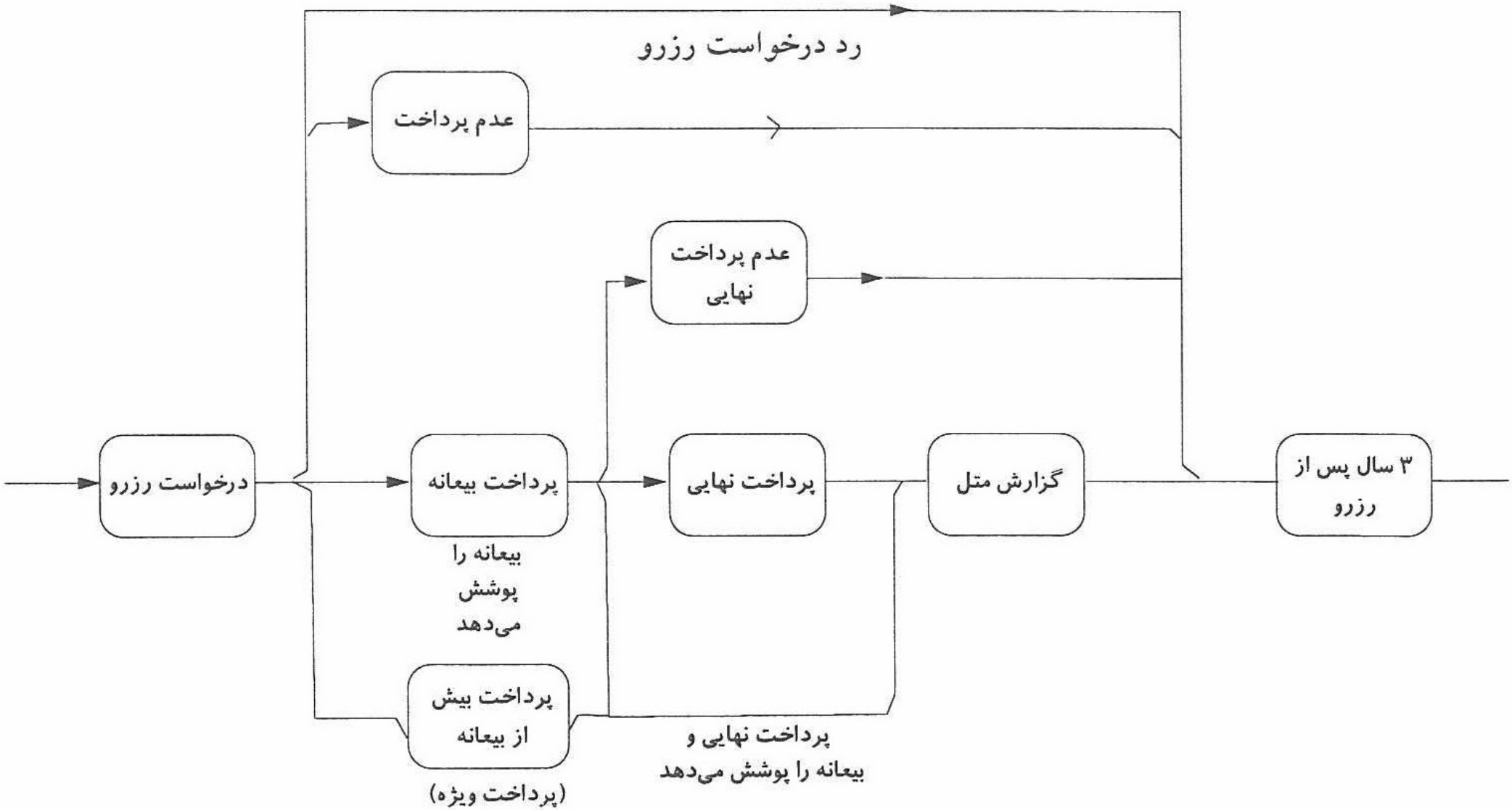


شکل ۱۳.۱۰ نمایش تکرار یک رویداد

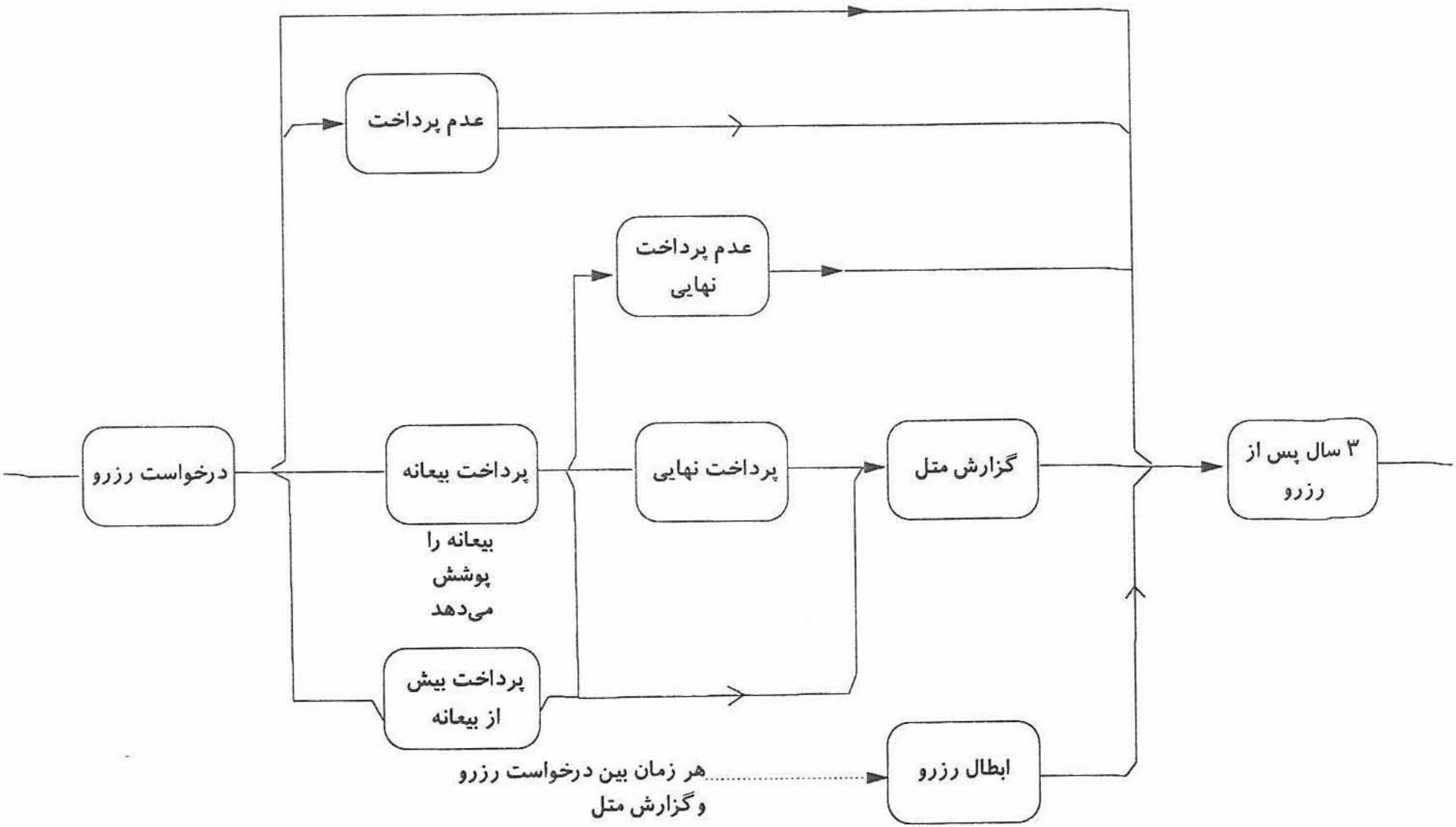
خروج‌ها

اگر ما رویداد ابطال را در نظر نگیریم و نمودار تاریخچه زندگی موجودیت رزرو به صورت شکل ۱۱-۱۳ خواهد که در بالای نمودار حالتی که درخواست رزور توسط سیستم پذیرفته نمی‌شود نمایش داده شده است. سیستم چنین تقاضاهایی را ثبت می‌کند تا بتواند آمار متل‌هایی که تقاضای زیادی برای آنها موجود است تهیه کند. در شکل همین‌طور رویداد پرداخت‌های ویژه (پرداخت‌هایی که از مبلغ بیعانه مورد نظر بیشتر است اما کل هزینه را پوشش نمی‌دهد) نیز در نظر گرفته شده است و همین‌طور «رویداد عدم پرداخت» چه در مورد بیعانه و چه در مورد پرداخت نهایی نشان داده شده و رفتار این موجودیت در مواجه با آن ترسیم گردیده است.

حال اگر بخواهیم رویداد ابطال را نشان دهیم به دلیل آن که رویداد ابطال در هر لحظه از زندگی موجودیت می‌تواند اتفاق افتد نمودار آن بسیار پیچیده می‌گردد، پس بهتر است از روش دیگری برای نمایش نمودار استفاده گردد. این روش در شکل ۱۲-۱۱ نمایش داده شده است و باعث می‌گردد از شلوغ شدن زیاد نمودار جلوگیری گردد.



شکل ۱۳.۱۱ نمودار تاریخچه زندگی موجودیت رزرو



شکل ۱۳.۱۲ نمودار تاریخچه زندگی موجودیت رزرو به همراه در نظر گرفتن رویداد «ابطال رزرو»

نقش موجودیت (Entity Role)

حال می‌خواهیم وضعیتی را در نظر بگیریم که یک موجودیت با توجه به یک رویداد کارهای مختلفی انجام می‌دهد و نقش‌های متفاوتی دارد. یعنی در اثر یک رویداد نمونه‌های یک موجودیت هم بهنگام‌سازی شده و هم ایجاد می‌گردد. برای بررسی مطلب سیستم مکانیزه رزرواسیون را در نظر می‌گیریم. در این سیستم رفتار موجودیت محل کاراوان را بررسی می‌کنیم. این موجودیت ارتباط بین دو موجودیت کاراوان و محل استقرار را برقرار می‌کند و در آن اطلاعات محل استقرار کاراوان در متل مربوطه نگهداری می‌گردد و شامل اقلام اطلاعاتی زیر است:

شماره کاراوان

کد متل

کد قسمت

تاریخ استقرار

تاریخ ترک محل

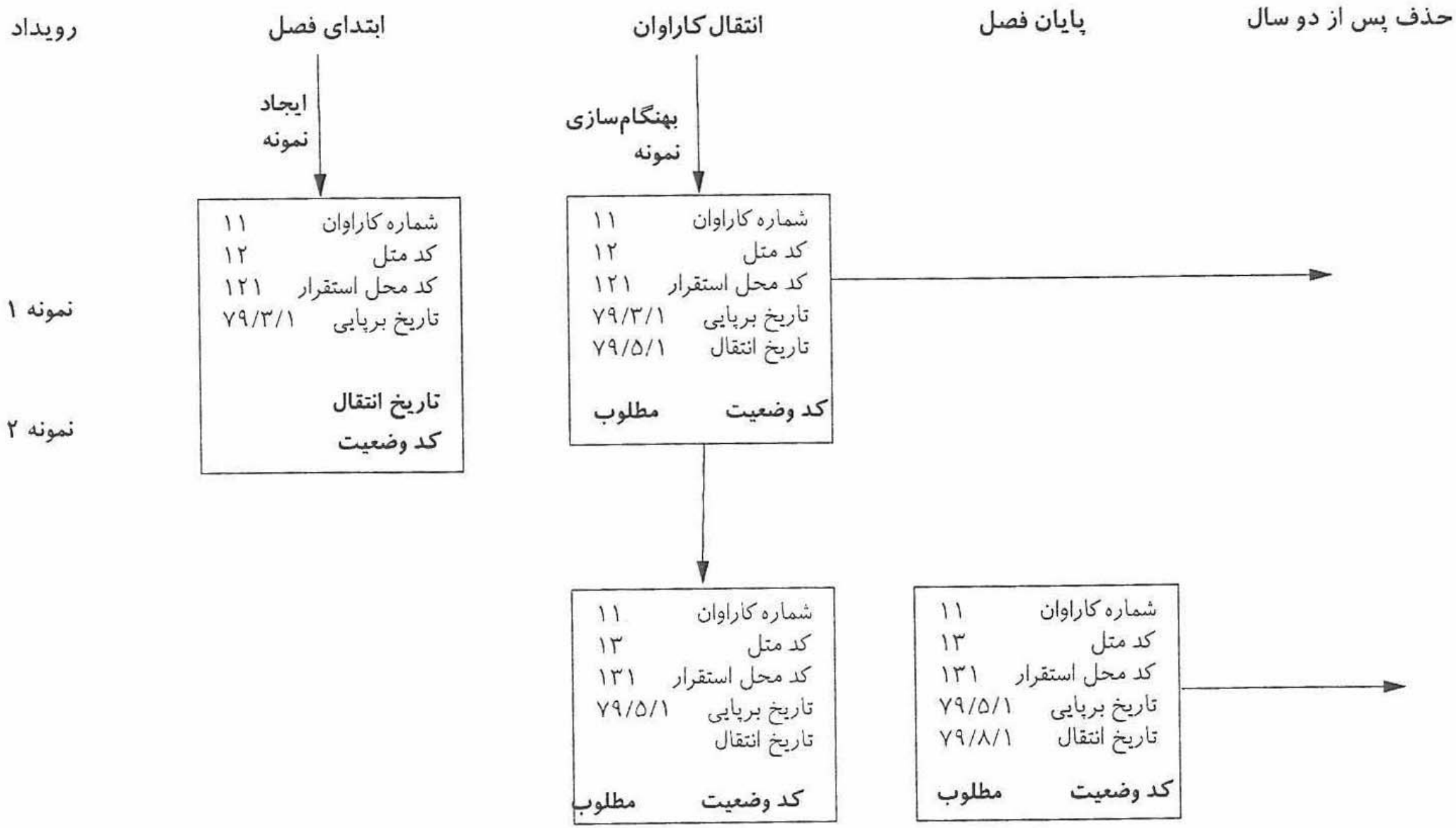
کد وضعیت

کاراوان‌ها معمولاً در ابتدای فصل در محل مربوطه استقرار می‌یابند و تا پایان فصل تعطیلات در محل مستقر هستند.

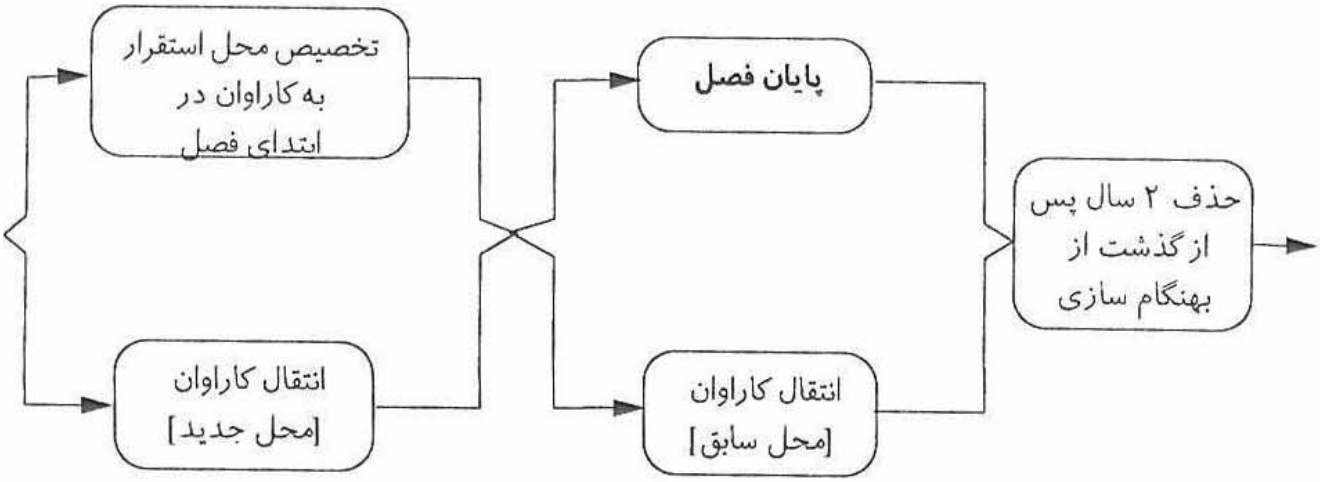
توجه داشته باشید که تاریخ استقرار کاراوان جزئی از کلید می‌باشد زیرا که کاراوان ممکن است تا سال بعد در همان محل باقی بماند. کد پایان فصل وضعیت کاراوان را در پایان فصل مشخص می‌سازد تا کاراوان‌هایی که احتیاج به تعمیر و بازبینی دارند مشخص گردند بنابراین در وضعیت نرمال هر نمونه موجودیت محل کاراوان با سه رویداد ایجاد یا تخصیص کاراوان به محل استقرار و بهنگام‌سازی وضعیت آن در انتهای فصل و حذف اطلاعات دو سال پس از تاریخ بهنگام‌سازی (حذف نمونه) مواجه می‌گردد حال اگر در یک فصل بعضی از متل‌ها با تقاضای زیادی از جانب مشتریان روبرو شوند و متل‌هایی نیز وجود داشته باشد که مشتریان کمتری تمایل به آنها دارند در این صورت شرکت بعضی از کاروان‌ها به محل‌های پرمشتری منتقل می‌کند و برای این که سیستم چنین وضعیتی را ثبت کند باید محل سابق استقرار کراوان نیز ذخیره گردد و سیستم به این صورت عمل می‌کند که تاریخ حرکت کاراوان از محل را در نمونه ثبت کرده و کد وضعیت آن را بهنگام می‌سازد و یک نمونه دیگر جهت استقرار کاراوان در محل جدید با

تاریخ جدید استقرار ایجاد می‌کند شکل ۱۳.۱۳ چنین وضعیتی را نشان می‌دهد.

بنابران رویداد انتقال کاراوان باعث بهنگام‌سازی یک نمونه و ایجاد نمونه دیگری می‌گردد که در نهایت و با گذشت ۲ سال از تاریخ بهنگام‌سازی نمونه حذف می‌گردد. در شکل ۱۳.۱۴ نمودار تاریخچه زندگی موجودیت کاراوان را نمایش می‌دهد.



شکل ۱۳.۱۳ اثر رویداد انتقال کاراوان بر نمونه‌ها



شکل ۱۳.۱۴ تاریخچه زندگی موجودیت محل کاراوان

توجه کنید که این رویداد دو اثر بر این موجودیت دارد یکبار باعث ایجاد نمونه می‌گردد و بار دیگر باعث بروزآوری نمونه دیگری می‌شود و در نمودار نیز دو بار تکرار شده است. بنابراین می‌گوییم که موجودیت در مقابل رویداد نقشی دوگانه ایفا می‌کند و هر دو نمونه با مشخص کردن نقش هر نمونه در داخل کروشه [bracket] مشخص می‌شود پس علاوه بر این که رویداد می‌تواند موجودیت را به دو صورت ایجاد یا بهنگام‌سازی متأثر کند مانند رویداد رزرو که می‌تواند یک نمونه جهت مشتریان جدید ایجاد کند و یا نمونه موجود اطلاعات مشتری را که قبلاً ایجاد شده بهنگام سازد موجودیت نیز می‌تواند دارای دو نقش متفاوت باشد.

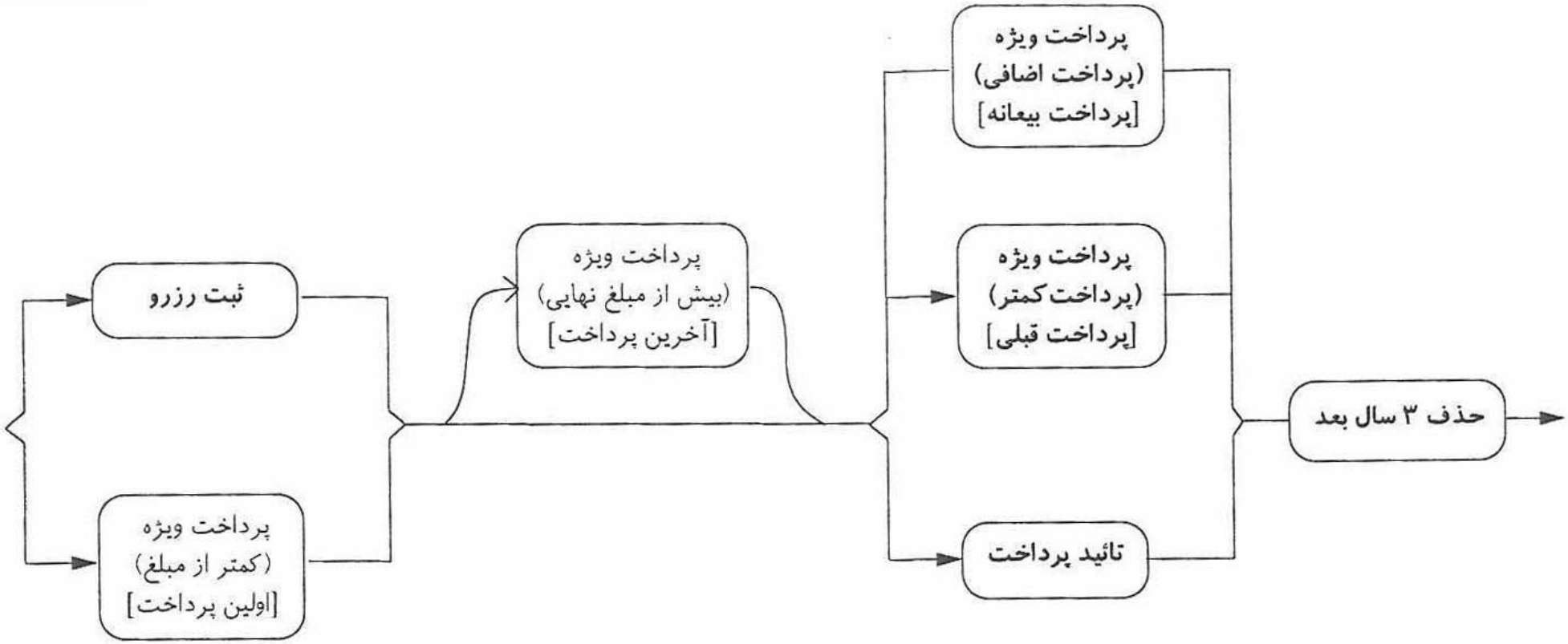
بنابراین می‌توان گفت که تأثیر رویداد و موجودیت این طور متمایز می‌شوند که اولی بر روی یک نمونه موجودیت تأثیر می‌کند و دومی بر روی دو نمونه موجودیت برای نشان دادن این تفاوت در هنگام رسم نمودار تاریخچه موجودیت می‌توان از نماد پرانتز برای نمایش یکی و از نماد کروشه برای دیگری استفاده کرد در کار با این علائم باید توجه گردد که استفاده از آنها زمانی که نقش رویداد و موجودیت بهم پیچیده است بسیار مشکل است.

شکل ۱۳.۱۵ را در نظر بگیرید این نمودار تاریخچه زندگی موجودیت پرداخت را نشان می‌دهد. یادآوری می‌شود که در سیستم رزرو فرض بر این است که به ازای هر رزرو دو پرداخت صورت می‌گیرد یکی جهت پرداخت مبلغ بیعانه و دیگری جهت مبلغ نهایی پس از رزرو کردن کاراوان دو نمونه در این موجودیت ایجاد می‌شود و پس هر پرداخت توسط مشتری یکی از این دو نمونه بروزآوری می‌شود و در نهایت پس از

گذشت ۳ سال از تاریخ پرداخت حذف می‌گردد و حال اگر وضعیتی را در نظر بگیرید که مشتری مبلغی کمتر از مبلغ بیعانه یا مبلغ پرداخت نهایی را پرداخت که ما آن را مبلغ ویژه می‌نامیم. در این صورت بسته به مورد نمونه‌ای که برای پرداخت نهایی یا پرداخت بیعانه ایجاد شده برابر مبلغ پرداختی بروزآوری می‌شود و سپس نمونه دیگری به اندازه مبلغ مابه‌التفاوت ایجاد می‌گردد و برای نمونه قبلی اتفاق خاصی صورت نمی‌گیرد تا پس از مدت معین حذف گردد با این روش می‌توان تمام مبالغ پرداختی توسط مشتری را ثبت و نگهداری کرد. حال اگر مشتری مبلغی بیش مبلغ مورد نظر جهت بیعانه را بپردازد همین کار انجام می‌شود با این تفاوت که نمونه مربوط به پرداخت نهایی اصلاح می‌شود و مبلغ مابه‌التفاوت را نشان می‌دهد.

زیرزندگی

نمودار تاریخچه زندگی موجودیت می‌تواند بسیار گسترده و پیچیده شود. به این منظور که نمودارها را ساده‌تر و کوچک‌تر سازیم می‌توانیم از مفهوم زیرزندگی، با استفاده از این مفهوم می‌توان قسمتی از نمودار را به صورت جداگانه و جدا از نمودار اصلی نمایش داد.



شکل ۱۳.۱۵ نمودار تاریخچه زندگی موجودیت پرداخت

فرض کنید موجودیت کاراوان دارای اقلام اطلاعاتی زیر باشد:

۱. شماره کاراوان

۲. نام تولیدکننده

۳. ارزش

۴. ظرفیت

۵. تاریخ خرید

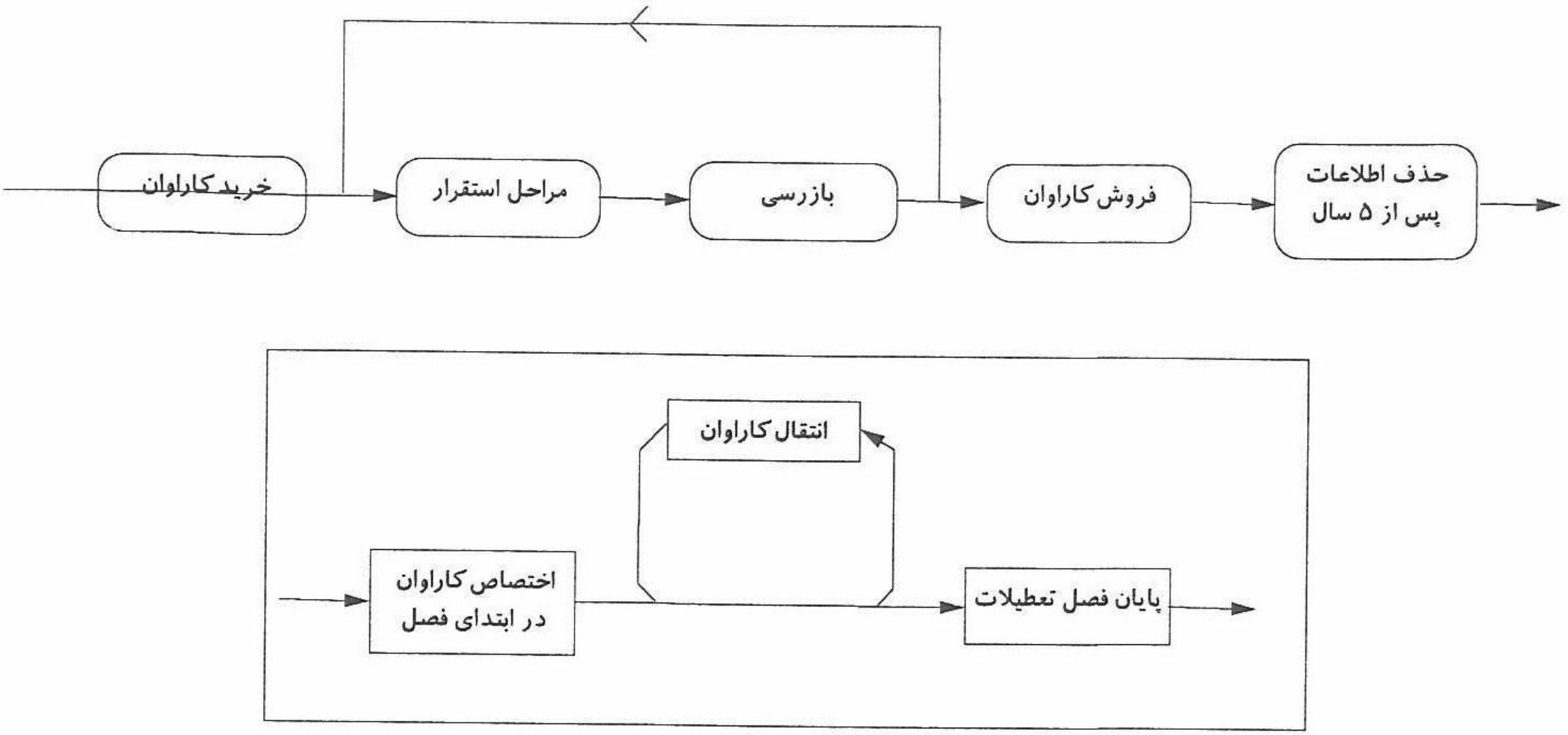
۶. تاریخ فروش

۷. تاریخ آخرین بازرسی

۸. تاریخ آخرین استقرار

۹. وضعیت

اگر تاریخچه زندگی موجودیت را به شکل ۱۳.۱۶ نمایش دهیم، پیچیدگی نمودار کاهش پیدا می‌کند و ساده‌تر می‌شود.



زیر زندگی مراحل استقرار

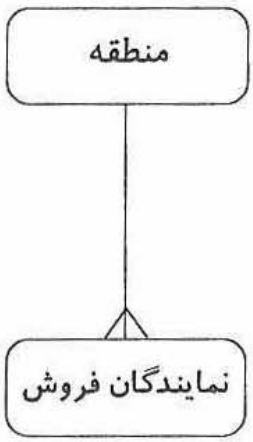
۱۳.۱۶ تاریخچه زندگی موجودیت کاراوان

اقلام وابسته، رویداد حذف و زیرموجودیت‌ها

پس از کامل کردن نمودار تاریخچه زندگی موجودیت قدم بعدی در نظر گرفتن تأثیر هر رویداد بر سایر موجودیت‌ها می‌باشد. اگر تمامی اثرات رویدادها به صورت کامل در نظر گرفته شده باشد ممکن است انجام این کار لزومی نداشته باشد. اما از آنجا که همیشه می‌تواند چیزهایی نادیده گرفته شود بهتر است این کار انجام گیرد به ویژه آن که ممکن است ارتباط یک موجودیت با موجودیت بالاسری یا پایین‌دستی آن نادیده گرفته شده باشد به همین دلیل در این جا ابتدا اقلام اطلاعاتی که اطلاعات آنها وابسته به سایر اقلام از قبیل سرجمع‌ها می‌باشد در نظر گرفته می‌شود و سپس به تأثیر رویداد حذف یک موجودیت بر سایر موجودیت‌ها و تأثیر رویدادها بر زیرموجودیت‌های می‌پردازیم.

شکل ۱۳.۱۷ را در نظر بگیرید فرض کنید یکی از اقلام اطلاعاتی موجودیت منطقه، کل فروش منطقه باشد و برابر مجموع فروش کلیه نمایندگان فروش در منطقه باشد بنابراین هر رویدادی که میزان فروش نماینده را تحت تأثیر قرار دهد بر این قلم اطلاعاتی نیز تأثیر می‌گذارد پس برای بررسی اثر رویدادها بر اقلام وابسته مراحل زیر را در نظر می‌گیریم:

۱. فهرست کلیه اقلام وابسته را تعیین می‌کنیم
۲. کلیه رویدادهایی را که بر آن تأثیر می‌گذارد را شناسایی می‌کنیم
۳. اثر رویدادها را در نمودار تاریخچه موجودیت نشان می‌دهیم



شکل ۱۳.۱۷ ارتباط موجودیت‌های منطقه‌ای نمایندگان فروش

پس از بررسی اقلام وابسته به بررسی تأثیر رویداد حذف بر سایر موجودیت‌ها به ترتیب زیر می‌پردازیم و قبل از هر چیز به سؤالات زیر پاسخ می‌دهیم.

۱. آیا حذف نمونه از یک موجودیت وابسته هیچ تأثیری بر نمونه‌های موجودیت بالاسری دارد

۲. آیا حذف موجودیت بالاسری باعث حذف یا تغییر موجودیت وابسته می‌گردد

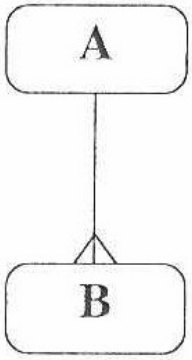
در حالت کلی حذف موجودیت وابسته تأثیری به موجودیت بالاسری ندارد اما فرض کنید:

موجودیت بالاسری شامل یک قلم اطلاعاتی باشد که برای هر نمونه نشان‌دهندهٔ تعداد یا متوسط نمونه‌ها در موجودیت وابسته باشد در این صورت حذف هر نمونه مسلماًبر نمونه موجودیت بالاسری مربوطه تأثیر می‌گذارد ولی لزوماً باعث حذف نمونه موجودیت بالاسری نمی‌گردد اما اگر بخواهیم اطلاعات کلیه مشتریانی که سفارشات آنها حذف گردیده را حذف کنیم در این صورت با حذف آخرین سفارش اطلاعات مشتری نیز حذف می‌گردد.

در پاسخ به سؤال دوم با توجه به شکل ۱۳.۱۸ باید سه حالت زیر را درنظر گرفت:

۱. آیا حذف یک نمونه از موجودیت A باعث حذف کلیه نمونه‌ها از موجودیت B می‌گردد.

۲. آیا کلیه نمونه‌های موجودیت Bکه وابسته به موجودیت A هستند می‌باید قبل از آن حذف گردند.



شکل ۱۳.۱۸ ارتباط دو موجودیت A و B

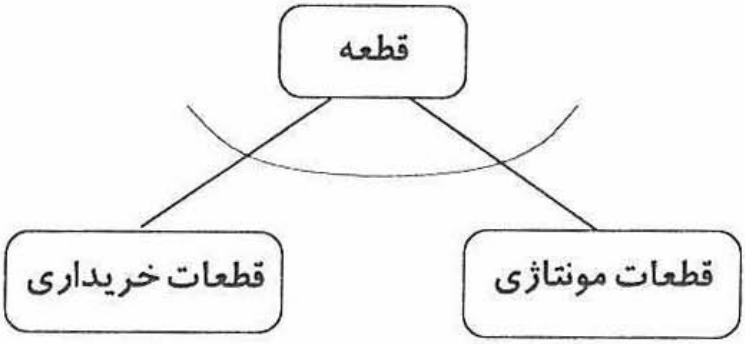
اگر حذف نمونه B وابسته به زمان باشد این زمان باید کوتاه‌تر از زمان حذف موجودیت A باشد و پردازشی که باعث حذف نمونه موجودیت A می‌شود ابتدا از حذف کلیه نمونه‌های مربوطه در موجودیت B اطمینان حاصل کند.

۳. آیا پس از حذف نمونه A نمونه‌های وابسته به موجودیت B نیز حذف می‌شوند یا همچنان باقی می‌ماند در این صورت مقدار کلید خارجی در موجودیت B به چه

مقداری تبدیل می‌شود.

در مورد رویداد حذف باید به سراغ کلیه موجودیت‌ها رفته و ابتدا به صورت موجودیت وابسته آنها را بررسی کنیم سپس همین موجودیت‌ها را به صورت موجودیت بالاسری بررسی کنیم.

در پایان به زیرموجودیت می‌پردازیم هر نمونه در زیرموجودیت باید براساس اطلاعات موجودیت اصلی ایجاد شده باشد و ایجاد آن بدون ایجاد نمونه مربوطه در موجودیت بالاسری بی‌معنی می‌باشد به مثال زیر توجه کنید:



شکل ۱۳.۱۹ ساختار موجودیت قطعه و دو زیرموجودیت آن

با توجه به شکل بالا ایجاد نمونه‌ها و رکوردها در موجودیت قطعات خریداری شده بدون درج اطلاعات در موجودیت قطعه بی‌معنی می‌باشد ولی حالت برعکس می‌تواند درست باشد یعنی اطلاعات قطعه می‌تواند ثبت گردد بدون آن که اطلاعات مربوط به خرید یا مونتاژ آن ثبت گردد و برای سایر رویدادها به غیر از حذف یا ایجاد اطلاعات زیرموجودیت می‌تواند بهنگام شود بدون آن‌که تأثیری بر موجودیت اصلی داشته باشد.

ایجاد فهرست رویدادها

پس از شناسایی کلیه رویدادها باید پردازش‌های متناسب را در کارکرد بروزرسانی بگنجانیم و بدین ترتیب در فرمی به صورت زیر علاوه بر ثبت و مستندسازی رویدادها نقش پردازش را در کارکرد متعلق به آن نشان دهیم.

این فرم تا حد زیادی شبیه ماتریس رویداد / موجودیت است اما به صورت فرم درآمده تا به صورت واضح و روشن ارتباط رویداد با کارکرد را نشان دهد.

فهرست رویدادها

سیستم: سیستم رزرواسیون متل　　**تاریخ** ۱۳۸۰/۰۵/۰۱　　**صفحه** ۱

شماره رویداد	نام رویداد	موجودیتها	نوع پردازش	نام کارکرد
۱	ثبت رزرو	مشتری	I/M(ایجاد/اصلاح)	
		رزرو	I (ایجاد)	
		همراهان	I (ایجاد)	
		پرداختها	I (ایجاد)	
۲	اصلاح مشخصات مشتری	مشتری	M (اصلاح)	اصلاح مشخصات مشتری
۳	۴ سال پس از آخرین رزرو	مشتری	D (حذف)	بازسازی‌اطلاعات (پایان سال)
۴	پرداخت	رزرو	M (اصلاح)	پرداخت
		پرداخت	M (اصلاح)	

۱۳.۲۰ فرم فهرست رویدادها

فصل چهاردهم

تعیین مسیر دسترسی

(Access Path)

اهداف فصل

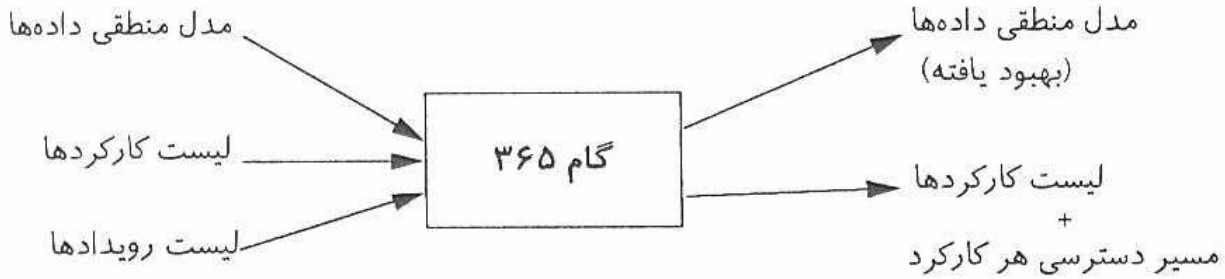
در SSADM در گام ۳۳۰ کارکردها مشخص می‌گردند و در آن گام کارکردها به تنهایی و با در نظر گرفتن صفحات ورودی و خروجی مشخص می‌شوند سپس در گام ۳۴۰ ساختار منطقی داده‌ها با توجه به این صفحات و کارکردها ایجاد می‌گردد و در گام ۳۶۰ تاریخچه زندگی موجودیت‌ها (E.L.H) ایجاد می‌گردد چیزی که می‌تواند از قلم بیفتد و فراموش گردد، اقلام اطلاعاتی می‌باشد که در صفحات خروجی نمایش داده نمی‌شود در این گام می‌خواهیم از طریق پیمایش ساختار منطقی داده‌ها و حرکت در آن و در نظر گرفتن اقلام اطلاعاتی ورودی به اقلام خروجی و صفحات گزارش دسترسی پیدا کنیم و بدین ترتیب از درستی ساختار منطقی داده‌ها اطمینان حاصل کنیم پس اهداف زیر را به طور کلی در این فصل دنبال می‌کنیم.

۱. بررسی و تشریح چگونگی حرکت در ساختار منطقی داده‌ها و تعیین مسیر دستیابی برای رسیدن به اطلاعات مورد نیاز هر کارکرد
۲. بررسی و ارزیابی ساختار منطقی داده‌ها به منظور حصول اطمینان از وجود کلیه اقلام اطلاعاتی لازم برای هر کارکرد
۳. تعریف پردازش‌های هر کارکرد و مشخص کردن چگونگی به روزآوری موجودیت‌ها تا بتوان در مراحل بعد مشخصات برنامه‌ها را براساس آن تعریف کرد
۴. بررسی کارآیی سیستم و تعیین زمان دسترسی به اطلاعات با توجه به اطلاعات موجود در حجم دادهها

مسیرهای دسترسی در گام ۳۶۵ تعیین می‌گردد. ورودی‌های این گام عبارتند از:

۱. ساختار منطقی داده‌ها
۲. لیست کارکردها
۳. فهرست رویدادها
۴. شرح موجودیت‌ها به همراه تخمینی از تعداد نمونه‌های هر موجودیت

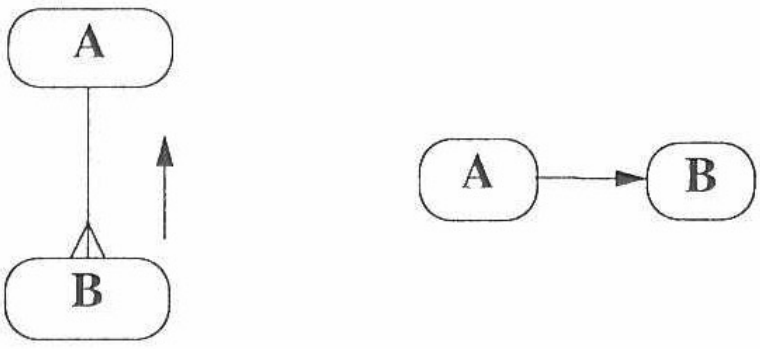
و خروجی آن تعیین مسیر دستیابی برای هر کارکرد می‌باشد و احتمالاً تغییراتی که در ساختار منطقی داده‌ها جهت بهسازی و رفع مشکلات آن ایجاد می‌شود.



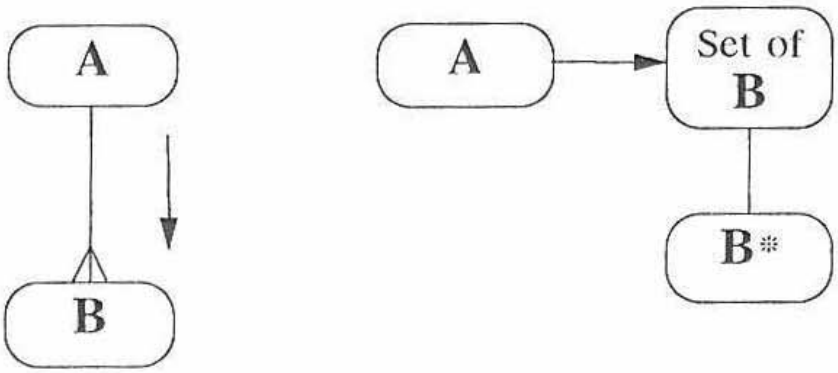
شکل ۱۴.۱ ورودی‌ها و خروجی‌های این گام

نمودار مسیر دسترسی چیست؟

نمودار مسیر دسترسی طریقه پیمایش ساختار منطقی داده‌ها را برای یک کارکرد نمایش می‌دهد. مقصود از پیمایش ساختار منطقی داده‌ها چگونگی حرکت بین موجودیت‌ها و رسیدن به موجودیت‌های موردنظر و اطلاعات آن می‌باشد. می‌توانیم از یک نوع ساده به صورت شکل ۲-۱۴ استفاده کنیم. در این شکل دو موجودیت وابسته A و B نمایش داده شده‌اند. به موجودیتی که موجودیت B به آن وابسته است (یعنی موجودیت A) موجودیت بالاسری (Master) و به موجودیت B موجودیت وابسته (detail) گفته می‌شود جهت فلش نشان‌دهندهٔ جهت حرکت می‌باشد و همواره فرض می‌کنیم که می‌توان از یک موجودیت وابسته به موجودیت بالاسری دست یافت از طرف دیگر در حرکت از موجودیت بالاسری به موجودیت وابسته هر بار یک نمونه از موجودیت بالاسری در نظر گرفته می‌شود(اصطلاحاً خوانده می‌شود) ولی احتمالاً بیش از یک نمونه از موجودیت وابسته به آن موجود می‌باشد که طریقه نمایش آن در شکل ۳-۱۴ آمده است.



شکل ۱۴.۲ چگونگی حرکت از موجودیت وابسته به موجودیت بالاسری و چگونگی نمایش در نمودار مسیر دسترسی

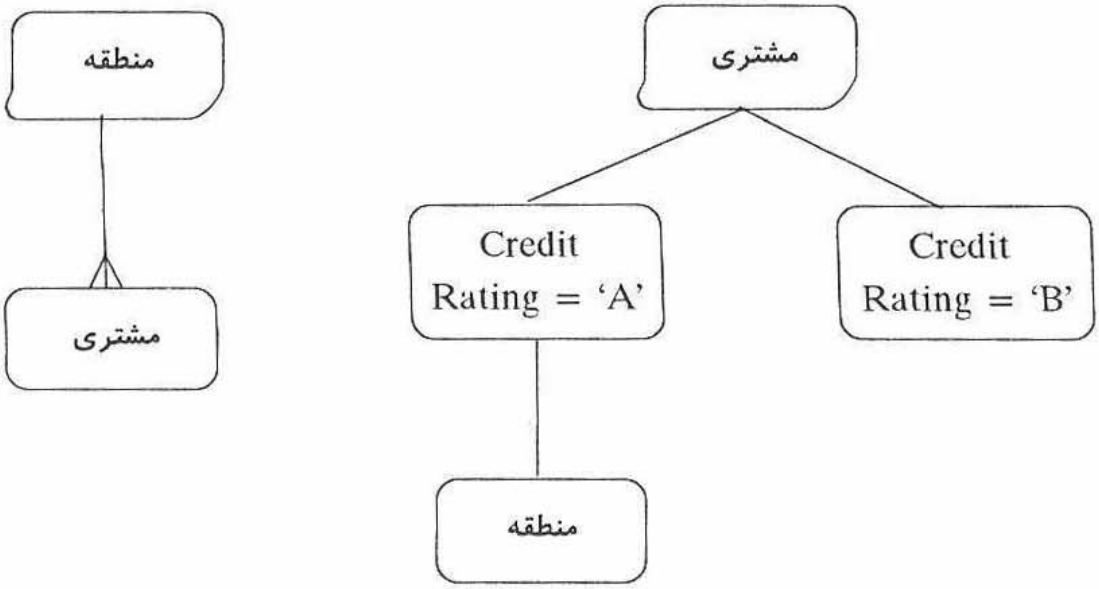


شکل ۱۴.۳ چگونگی حرکت از موجودیت بلاسری به موجودیت وابسته و چگونگی نمایش آن در نمودار مسیر دسترسی

یعنی پس از خواندن یک نمونه از موجودیت بالاسری به یک مجموعه از نمونه‌های موجودیت وابسته دسترسی پیدا می‌کنیم و این مجموعه توسط یک مستطیل گرد گوشه که در آن عبارت "set of" و نام موجودیت است نشان داده می‌شود که با یک خط عمودی به موجودیت وابسته متصل شده است و موجودیت وابسته با یک علامت ستاره مشخص می‌شود تا نشان داده شود که از موجودیت B (موجودیت وابسته) می‌تواند صفر یا چند نمونه (Occurance) موجود باشد که به یک نمونه از موجودیت A وابسته است.

اگر از علایم جکسون همان‌طور که در فصل قبل گفته شد استفاده کنیم این مجموعه را می‌توانیم به صورت یک عملگر تکرار در نظر بگیریم.

عملگری دیگری که علائم جکسون از آن استفاده می‌کنند، عملگر انتخاب است. از این عملگر هنگامی استفاده می‌شود که بخواهیم از بین دو یا چند وضعیت یکی را انتخاب کنیم شکل ۴-۱۴ چنین موردی را نمایش می‌دهد این شکل نشان می‌دهد که پس از بررسی نمونه موجودیت مشتری بسته به مقدار قلم اطلاعاتی رده اعتباری اگر مقدار رده اعتباری "A" باشد می‌توانیم به موجودیت منطقه برسیم در غیر این صورت نمی‌توان به این موجودیت رسید، پس براساس مقدار یک قلم اطلاعاتی مسیرهای مختلفی طی می‌گردد و برای نمایش این مطلب که فقط یکی از دو حالت انتخاب می‌شود، در گوشه سمت چپ مستطیل یک علامت "o" می‌گذاریم.



شکل ۱۴.۴

چگونگی رسم نمودار مسیر دسترسی

ابتدا کارکردها را به دو دسته کارکردهای پرس‌وجو و کارکردهای بهنگام‌سازی تقسیم می‌کنیم و در این گام تنها به تعیین مسیر دسترسی در مورد کارکردهای پرس‌وجو می‌پردازیم. برای رسم نمودار، کارهای زیر را انجام می‌دهیم:

۱. شناسایی خروجی‌ها و گزارشات: یعنی اقلام اطلاعاتی که کارکرد تولید می‌کند را درنظر می‌گیریم.

۲. شناسایی ورودی‌ها: یعنی برای آغاز پردازش کدامیک از اقلام اطلاعاتی مورد نیاز می‌باشد. برای کارکردهای لحظه‌ای مسلماً اطلاعات ورودی را کاربر وارد می‌کند و برای سایر کارکردها اطلاعات می‌تواند توسط سیستم عامل یا اپراتور وارد گردد. این اطلاعات می‌تواند اطلاعاتی از قبیل ساعت و یا تاریخ باشد.

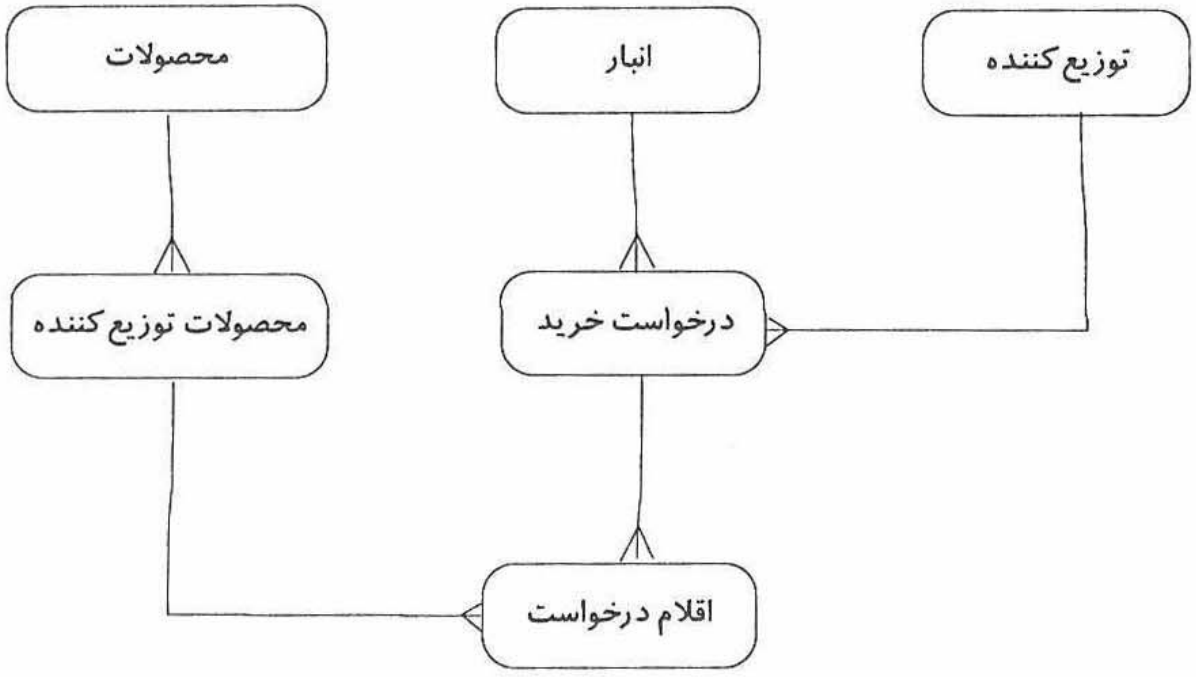
۳. شناسایی موجودیت‌های مورد نیاز: پس از شناسایی اقلام اطلاعاتی خروجی، موجودیت‌هایی که این اقلام اطلاعاتی در آنها وجود دارد را شناسایی می‌کنیم بعضی از اقلام ممکن است در چندین موجودیت مشترک باشند فعلاً فقط نام آنها را ثبت می‌کنیم ولی اگر از یک Case tool استفاده شود و موجودیت‌ها به صورت فرم نرمال از نوع سوم باشد انتظار می‌رود که این ابزار خود بتواند آن را شناسایی کند.

۴. شناسایی روابط مسیر: معمولاً ورودی‌های هر کارکرد نقطه شروع مسیر را تعیین



شکل ۱۴.۵ موجودیت‌های مثال مربوطه

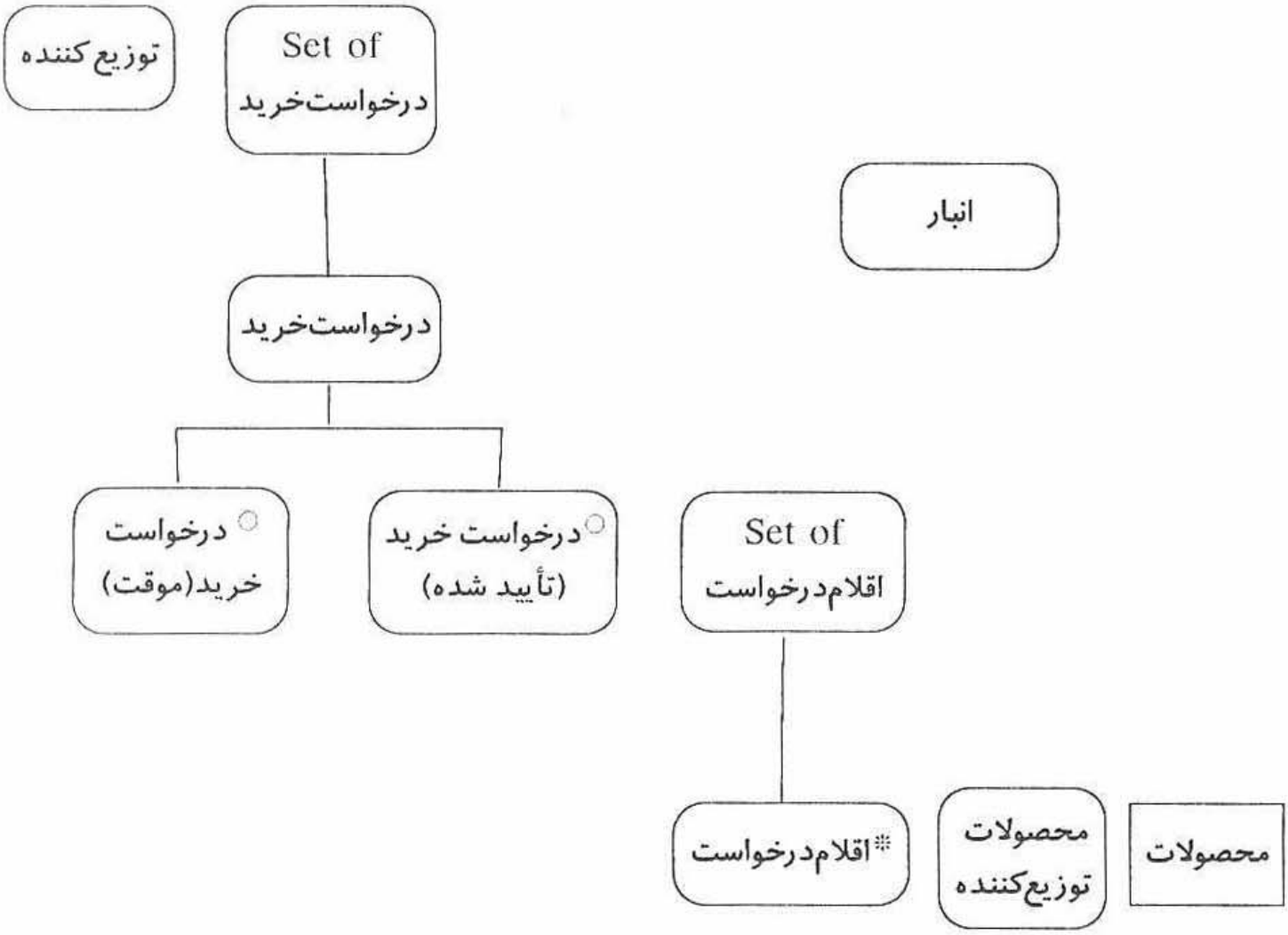
۴. بررسی روابط: رابطه‌هایی را که بین موجودیت‌های شناسایی شده وجود دارند و در تهیه گزارش مورد استفاده قرار می‌گیرند، در نظر می‌گیریم یا به عبارتی قسمتی از ساختار منطقی داده‌ها را که مورد استفاده است در نظر می‌گیریم. در مورد این مثال روابط بین موجودیت‌ها به صورت زیر است:



شکل ۱۲.۶ ساختار منطقی داده‌ها (مثال مربوطه)

۵. شناسایی عملگرهای انتخاب و تکرار: در این قسمت عملگرهای انتخاب و تکرار و این که روی چه موجودیت‌هایی تأثیر می‌گذارند را شناسایی می‌کنیم و آن را به نمودار اضافه می‌نماییم. در این مثال از دو عملگر تکرار یکی در مورد موجودیت درخواست خرید که از موجودیت توزیع‌کننده به آن می‌رسیم و

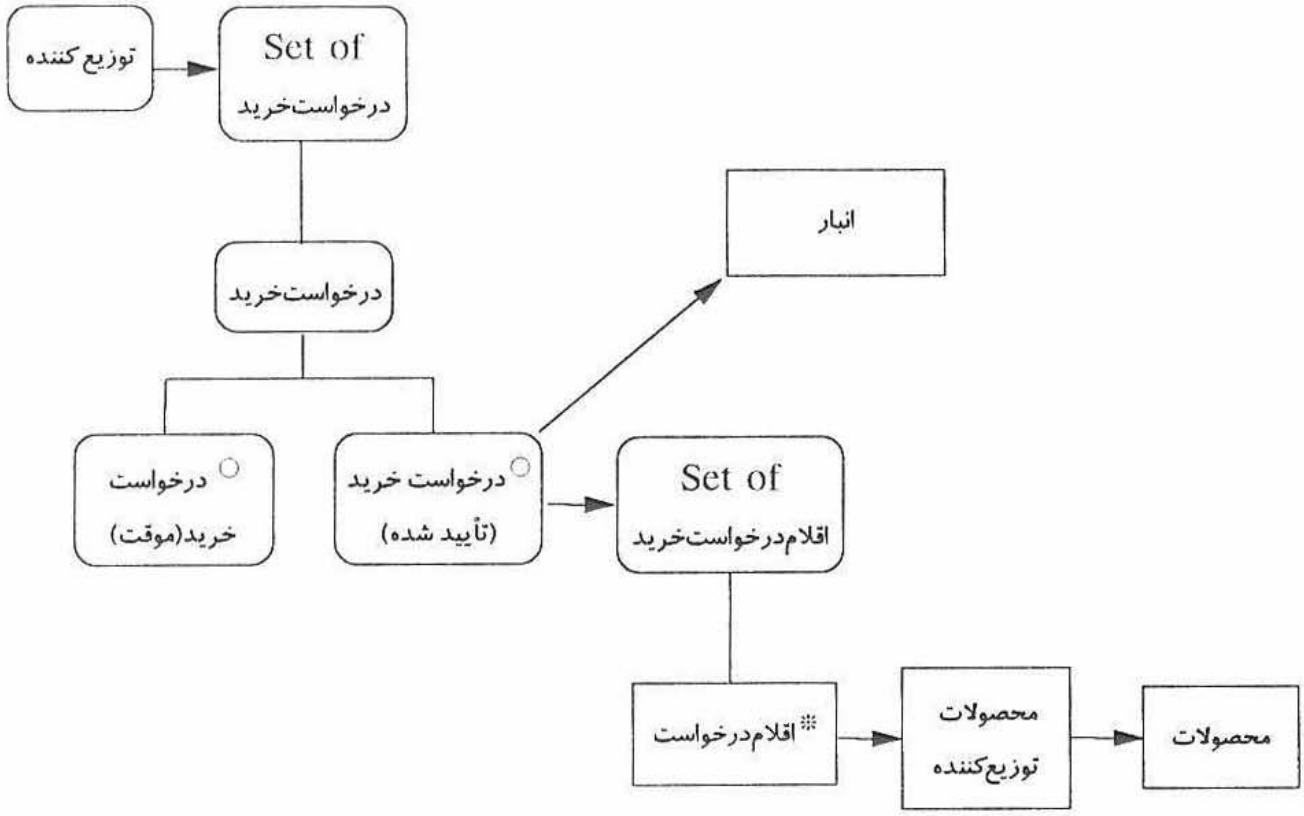
دیگری در مورد موجودیت اقلام درخواست خرید که از موجودیت درخواست خرید به آن می‌رسیم و از یک عملگر انتخاب استفاده شده که نشان می‌دهد با توجه به نوع درخواست خرید بین درخواست خریدهای تأیید شده و موقت انتخاب صورت می‌گیرد.



شکل ۱۴.۷

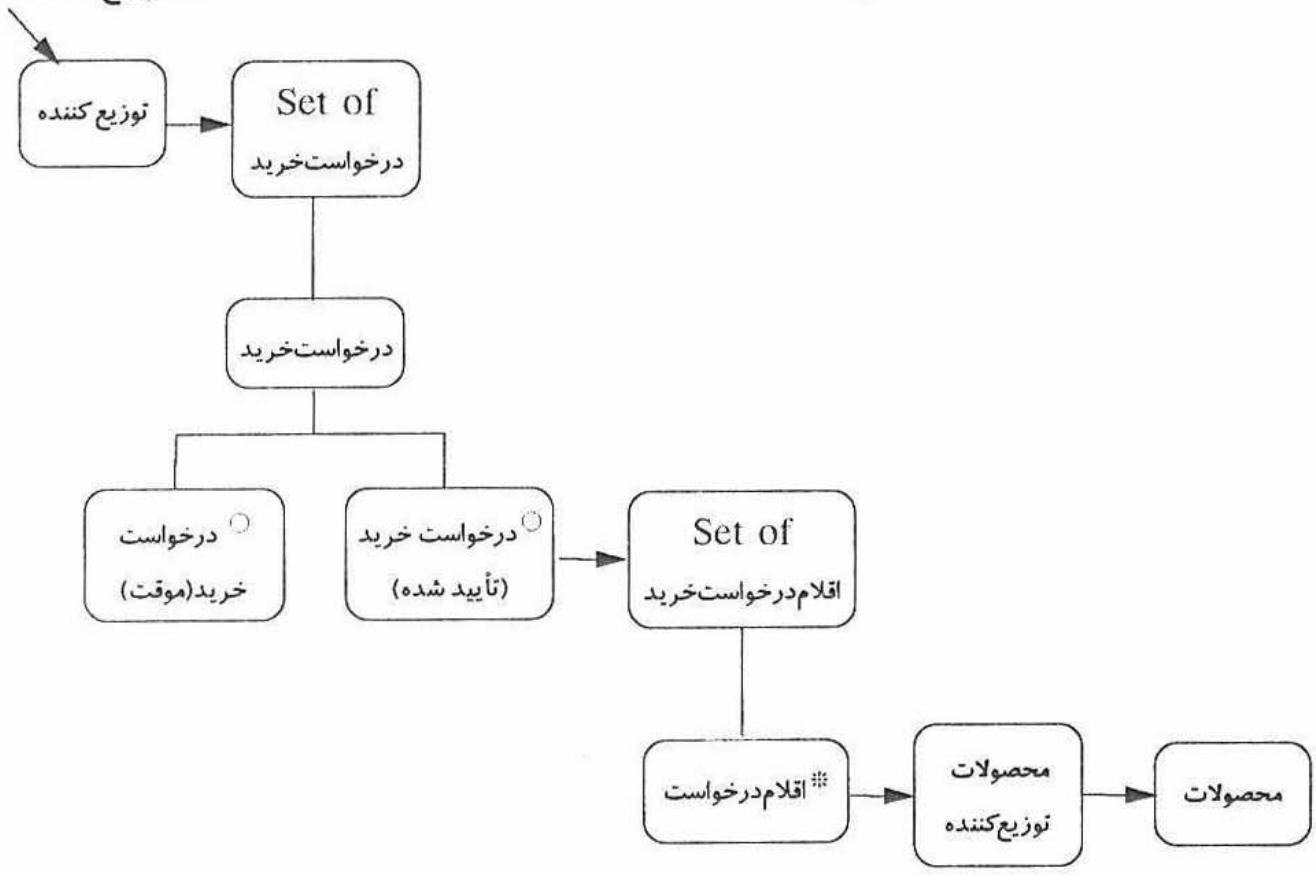
۶. ترسیم ارتباطات یک به یک: حال خطوطی را که نشانگر مسیر دستیابی بین مستطیل‌های مختلف است رسم می‌کنیم که در مورد مثال، عبارتند از:

- ارتباط توزیع‌کننده با درخواست خرید
- ارتباط درخواست خرید با اقلام درخواست
- ارتباط درخواست خرید با انبار
- ارتباط درخواست خرید با محصولات توزیع‌کننده
- ارتباط محصولات توزیع‌کننده با محصولات



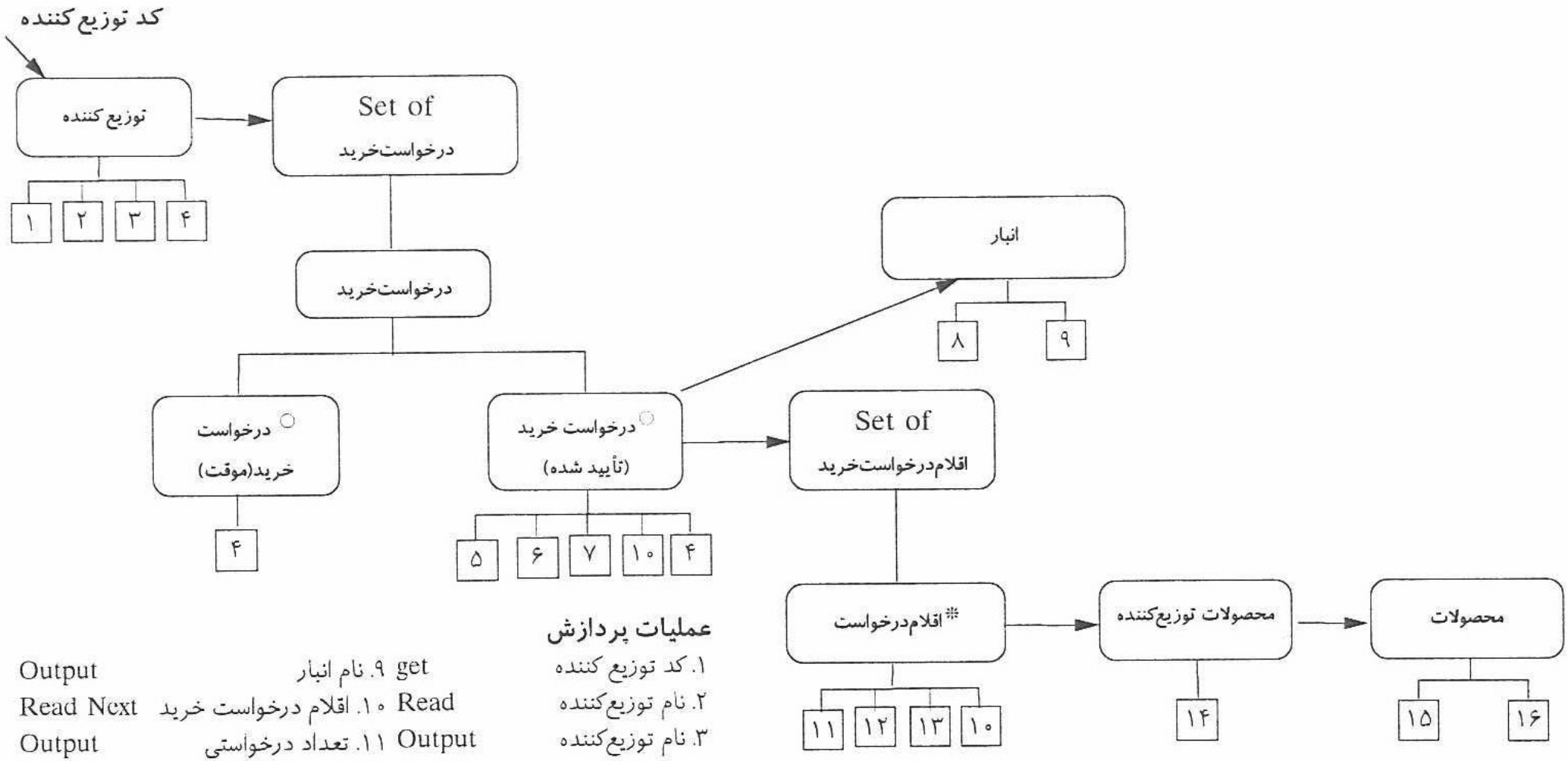
شکل ۱۴.۸

۶. مشخص کردن نقاط ورودی: در این قسمت موجودیتی را که از آن موجودیت مسیر را شروع می‌کنیم به همراه کلید موجودیت را در نمودار مشخص می‌کنیم که در مورد مثال، کد توزیع‌کننده می‌باشد.



شکل ۱۴.۹

۷. اضافه کردن عملیات: در این قسمت عملیات مربوط را اضافه می‌کنیم همان‌طور که گفته شد در این کارکرد ابتدا شماره توزیع‌کننده را از ورودی می‌گیریم پس می‌توانیم آن را توسط عمل "get" نشان دهیم. سپس نمونه مربوط به موجودیت توزیع‌کننده (رکورد مربوطه از این موجودیت) را می‌خوانیم که آن را توسط عمل "Read" نشان می‌دهیم سپس اطلاعات مربوط به توزیع‌کننده از قبیل آدرس و غیره را توسط عمل "output" نشان می‌دهیم و بعد از آن درخواست‌های خرید این توزیع‌کننده را خوانده و مجدداً آن را با عمل "Read" نشان می‌دهیم ولی چون می‌تواند چندین درخواست وجود داشته باشد و باید تک‌تک آنها را در نظر بگیریم و به دلیل آن‌که از موجودیت بالاسری به موجودیت وابسته رسیده‌ایم آن را با عمل "Read Next" نمایش می‌دهیم و سپس با عمل "Read" دیگر نام و شماره انبار را می‌خوانیم و آن را در اختیار کارکرد نمایش می‌دهیم که این کار را با عمل "output" نشان می‌دهیم، سپس اقلام درخواست خوانده می‌شود "Read Next" که تعداد تأیید شده آن را در اختیار کارکرد قرار می‌دهیم (با عمل "output") سپس موجودیت محصولات هر توزیع‌کننده را خوانده "Read" و سپس موجودیت محصول را با عمل "Read" خوانده و شماره محصول و نام آن را در اختیار کارکرد قرار می‌دهیم تا به صورت output نمایش دهد.



عملیات پردازش

۱. کد توزیع کننده	get	۹. نام انبار	Output
۲. نام توزیع‌کننده	Read	۱۰. اقلام درخواست خرید	Read Next
۳. نام توزیع‌کننده	Output	۱۱. تعداد درخواستی	Output
۴. درخواست خرید	Read Next	۱۲. تعداد تأیید شده	Output
۵. تاریخ درخواست	Output	۱۳. محصولات توزیع‌کننده	Read
۶. ساعت درخواست	Output	۱۴. محصولات	Read
۷. انبار درخواست	Read	۱۵. کد محصول	Output
۸. کد انبار	Output	۱۶. نام محصول	Output

شکل ۱۰. ۱۴

۹. در پایان شرط ادامه عملگرهای تکرار و انتخاب را تعیین می‌کنیم که در این جا همان‌طور که مشخص شده است شرط تکرار در مورد موجودیت درخواست خرید تا آنجا ادامه پیدا می‌کند که کد توزیع‌کننده عوض شود یا در مورد اقلام درخواست خرید تا آنجا ادامه می‌یابد که شماره درخواست عوض شود و شرط عمل انتخاب نیز به وضعیت درخواست مربوط می‌گردد.

نمودار تطابق اثر (Effect Correspondence Diagram)

در این بخش می‌خواهیم چگونگی رسم نمودار مسیر دسترسی برای کارکردهای بروزرسانی را بیان کنیم. این نمودار را برای چنین کارکردهای در SSADM نسخه ۴۰۲ به بعد نمودار تطابق اثر می‌نماند و اساس آن را بر تأثیر رویدادها بر موجودیت‌ها می‌گذارد.

این نمودار نشان می‌دهد که رویداد روی چه موجودیت‌هایی تأثیر می‌گذارد و این تأثیر چگونه می‌باشد؟ و طرز عمل کارکرد را نشان می‌دهد. برای رسم نمودار کارهای زیر را انجام می‌دهیم.

۱. شناسایی موجودیت‌ها:

در این قسمت ابتدا سعی می‌کنیم که موجودیت‌هایی را که کارکرد بر روی آن تأثیر می‌گذارد را شناسایی کنیم. مناسب‌ترین منبع برای پیدا کردن این موجودیت‌ها ماتریس رویداد / موجودیت می‌باشد که با نگاه کردن به آن می‌توانیم موجودیت‌هایی که کارکرد روی آن تأثیر می‌گذارند را پیدا کنیم.

سایر منابعی که می‌تواند در مورد شناسایی موجودیت‌ها به ما کمک کند عبارتند از:

- نمودار تاریخچه زندگی موجودیت
- لیست کارکردها
- ساختار منطقی داده‌ها

هر موجودیت را به صورت یک مستطیل گردگوشه نمایش می‌دهیم و در هنگام رسم نمودار زیر موجودیت‌ها را به صورت جداگانه نمایش می‌دهیم.

۲. اضافه کردن آثار رویداد:

برای ثبت این آثار باید به دو نکته توجه کنیم:

نکته اول: همان‌طور که در فصل پیش گفته شد یک رویداد ممکن است آثار مختلفی را بر روی یک موجودیت بگذارد و بسته به شرایط در حالتی نمونه موجودیت را به

روزآوری کند و در حالتی دیگر آن را ایجاد نماید. این حالت را می‌توان با یک عملگر انتخاب نمایش داد (برای نمایش نام موجودیت را نوشته و هر حالت را درون پرانتز نمایش می‌دهیم.

نکته دوم: یک رویداد ممکن است بیش از یک نمونه از موجودیت را متأثر کند، یعنی به طور مثال یک نمونه موجودیت را به روزآوری کند و نمونه دیگر را ایجاد نماید و برای نشان دادن آن هر اثر را با یک مستطیل که نام موجودیت در آن نوشته شده و نام اثر در کروشه قرار دارد نشان می‌دهیم (همان‌طور که در فصل پیش در بخش نقش موجودیت شرح داده شد).

۳. مشخص کردن نقاط ورودی:

ورودی کارکرد را مشخص می‌کنیم و به این ترتیب موجودیتی که از آن موجودیت نمودار شروع می‌شود، تعیین می‌گردد. ممکن است چندین نقطه ورودی موجود باشد در این صورت کلیه نقاط را مشخص می‌کنیم.

۴. تعیین ارتباطات یک به یک:

در این جا موجودیت‌هایی را که رابطه یک به یک با هم دارند، مشخص می‌کنیم. مقصود از رابطه یک به یک آن است که به ازای یک نمونه موجودیت اول، یک نمونه از موجودیت دوم موجود باشد.

۵. مشخص‌سازی عملگرهای تکرار:

موجودیت‌هایی که با یکدیگر رابطه یک به چند دارند را در این قسمت مشخص می‌کنیم. این حالت وقتی وجود دارد، که موجودیت اول موجودیت بالاسری و موجودیت دوم موجودیت وابسته باشد و هنگامی آن را با استفاده از عملگر تکرار مشخص می‌کنیم که به ازای یک نمونه از موجودیت اول چندین نمونه از موجودیت دوم موجود باشد و این عملگر را به صورت یک مستطیل که عبارت "set of" و نام موجودیت در آن نوشته شده و مستطیل دیگری که به آن مربوط است و فقط نام موجودیت و علامت "O" در آن می‌باشد، مشخص می کنیم.

۶. اضافه کردن شرط‌ها:

وضعیتی که تا برقراری آن عمل تکرار ادامه می‌یابد را مشخص می‌کنیم و در نمودار نشان می‌دهیم. معمولاً این وضعیت مقدار یک قلم اطلاعاتی است که تا عوض شدن آن عمل تکرار ادامه می‌یابد.

۷. مانند نمودار مسیر دستیابی در این قسمت عملیاتی که روی موجودیت‌ها صورت می‌گیرد را نشان می‌دهیم.

عملیاتی که معمولاً از آن استفاده می‌شود عبارتند از:

۱. «نام موجودیت» Create با این عمل نشان می‌دهیم که یک نمونه موجودیت ایجاد می‌گردد.

۲. «نام قلم اطلاعاتی» Set با این عمل نشان می‌دهیم که مقدار یک قلم اطلاعاتی عوض می‌گردد و به مقدار نشان داده شده تغییر می‌کند.

۳. «نام موجودیت» Tie to این عمل نشان می‌دهد که یک نمونه وابسته در موجودیت وابسته به یک نمونه از موجودیت بالاسری متصل می‌شود (قلم اطلاعاتی کلید خارجی با کلید موجودیت بالاسری پر می‌شود)

۴. «نام موجودیت» Cut from با این عمل نشان می‌دهیم که ارتباط یک نمونه وابسته از موجودیت وابسته با موجودیت بالاسری قطع می‌گردد.

۵. «نام موجودیت» Gain برقراری ارتباط بین یک نمونه موجودیت بالاسری با یک یا چند نمونه در موجودیت وابسته را نشان می‌دهیم در حالتی که نمونه‌های موجودیت وابسته بیش از یک باشد. با عمل «نام موجودیت» Gain Set of نشان می‌دهیم.

۶. «نام موجودیت» Lose قطع ارتباط نمونه موجودیت بالاسری با یک یا چند نمونه از موجودیت وابسته را نشان می‌دهیم.

۷. «نام قلم اطلاعاتی» Fail if SI نشان می‌دهد که اگر مقدار قلم اطلاعاتی برابر مقدار تعیین شده نباشد، عمل به روزرسانی متوقف گردد.

۸. «نام موجودیت» Write نشان می‌دهد که اطلاعات مربوطه در موجودیت نام برده شده ثبت گردد.

رسم نمودار یک مثال

در این مثال رویداد ورود درخواست خرید اقلام پیشنهادی را در نظر می‌گیریم و می‌خواهیم نمودار تطابق اثر آن را رسم کنیم. برای رسم آن کارهای زیر را انجام می‌دهیم:

۱. ماتریس موجودیت / رویداد را در نظر می‌گیریم براساس آن می‌توان

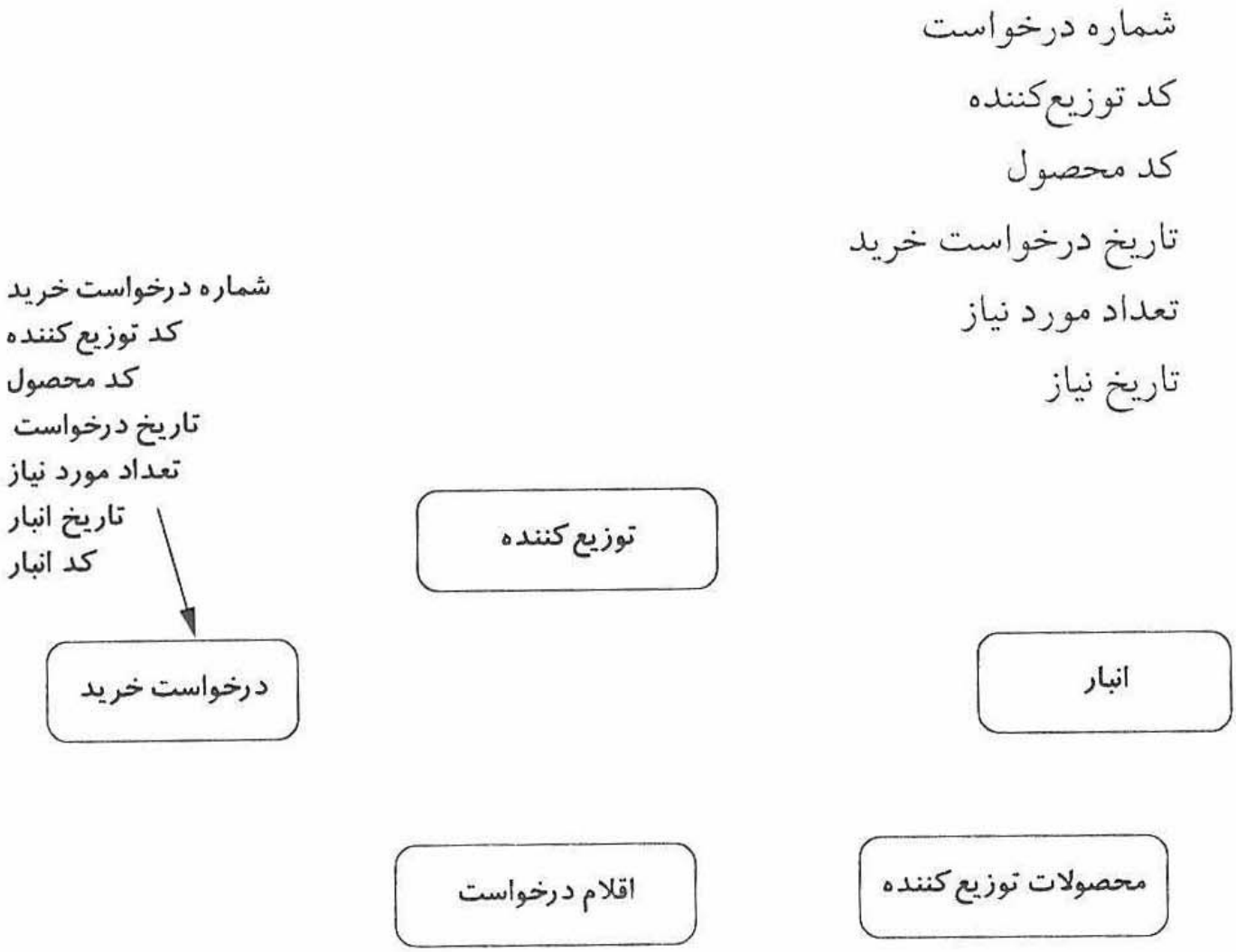
موجودیت‌هایی که رویداد روی آنها تأثیر می‌گذارد را پیدا کنیم. فرض کنید قسمتی از ماتریس که این رویداد و موجودیت‌های مربوطه را نشان می‌دهد، به صورت زیر باشد:

رویداد / موجودیت	ورود اطلاعات درخواست خرید
توزیع‌کننده	M
انبار	M
درخواست خرید	I
اقلام درخواست	I
اجناس توزیع‌کننده	I

جدول ۴.۱۱

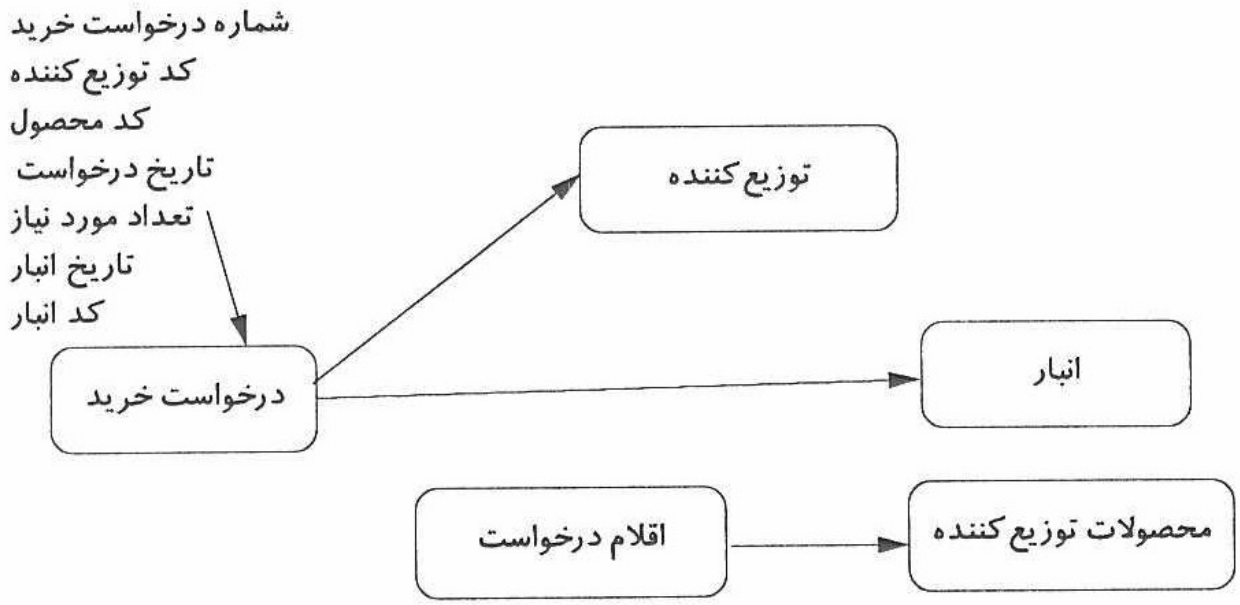
۲. اثرات مختلف رویداد بر موجودیت را بررسی می‌کنیم. در این مثال چون رویداد با موجودیت‌ها به صورت دوگانه برخورد نمی‌کند و همین‌طور حالتی وجود ندارد که رویداد روی دو نمونه از موجودیت تأثیر بگذارد، پس اثر خاصی وجود ندارد. اگر رویداد با موجودیتی به صورت دوگانه برخورد کند (به طور مثال در یک حالت یک نمونه را ایجاد نماید و در حالت دیگری آن را به روزآوری کند) آن را با عملگر انتخاب نشان می‌دهیم.[۱]

۳. سپس اطلاعات ورودی و موجودیت ابتدای مسیر را شناسایی می‌کنیم. که در این مثال ورودی‌ها عبارتند از:



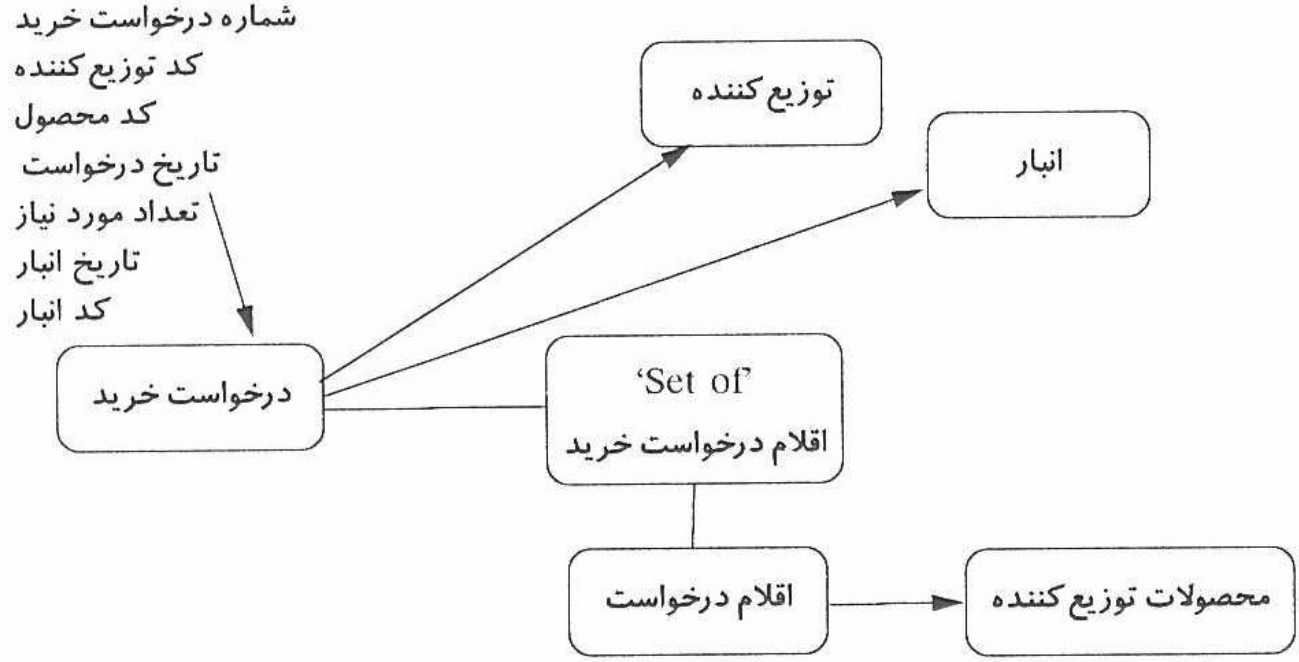
شکل ۱۴.۱۲

۴. تعیین سایر موجودیت‌هایی که می‌توان براساس روابط بین موجودیت‌ها به آن رسید و حالت یک به یک دارند در این جا موجودیت درخواست با توزیع‌کننده رابطه یک به یک دارند زیرا هر درخواست برای یک توزیع‌کننده صادر می‌گردد و چون هر درخواست برای یک انبار صادر می‌گردد پس موجودیت درخواست با موجودیت انبار رابطه یک به یک دارند و همین‌طور چون هر قلم درخواست *برای یک محصول توزیع‌کننده ایجاد شده پس این موجودیت با موجودیت* محصول توزیع‌کننده ارتباط یک به یک دارد.



شکل ۱۴.۱۳

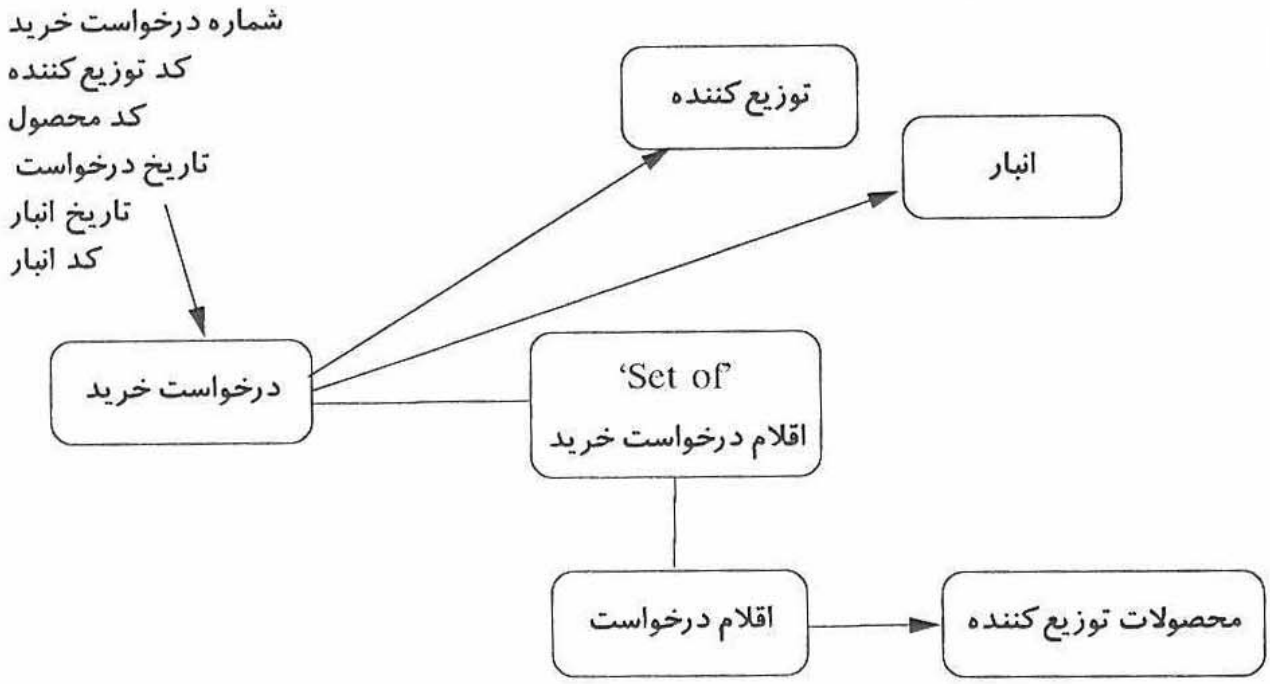
۵. مشخص کردن عملگرهای تکرار: همان‌طور که گفته شد وقتی از یک موجودیت بالاسری به یک موجودیت وابسته می‌رسیم. اگر به ازای هر نمونه موجودیت بالاسری چندین نمونه از موجودیت وابسته وجود داشته باشد آن را با عملگر تکرار نمایش می‌دهیم. در این مثال وقتی می‌خواهیم از درخواست خرید به اقلام درخواست برسیم آن را با عملگر تکرار نمایش می‌دهیم.



شکل ۱۴.۱۴

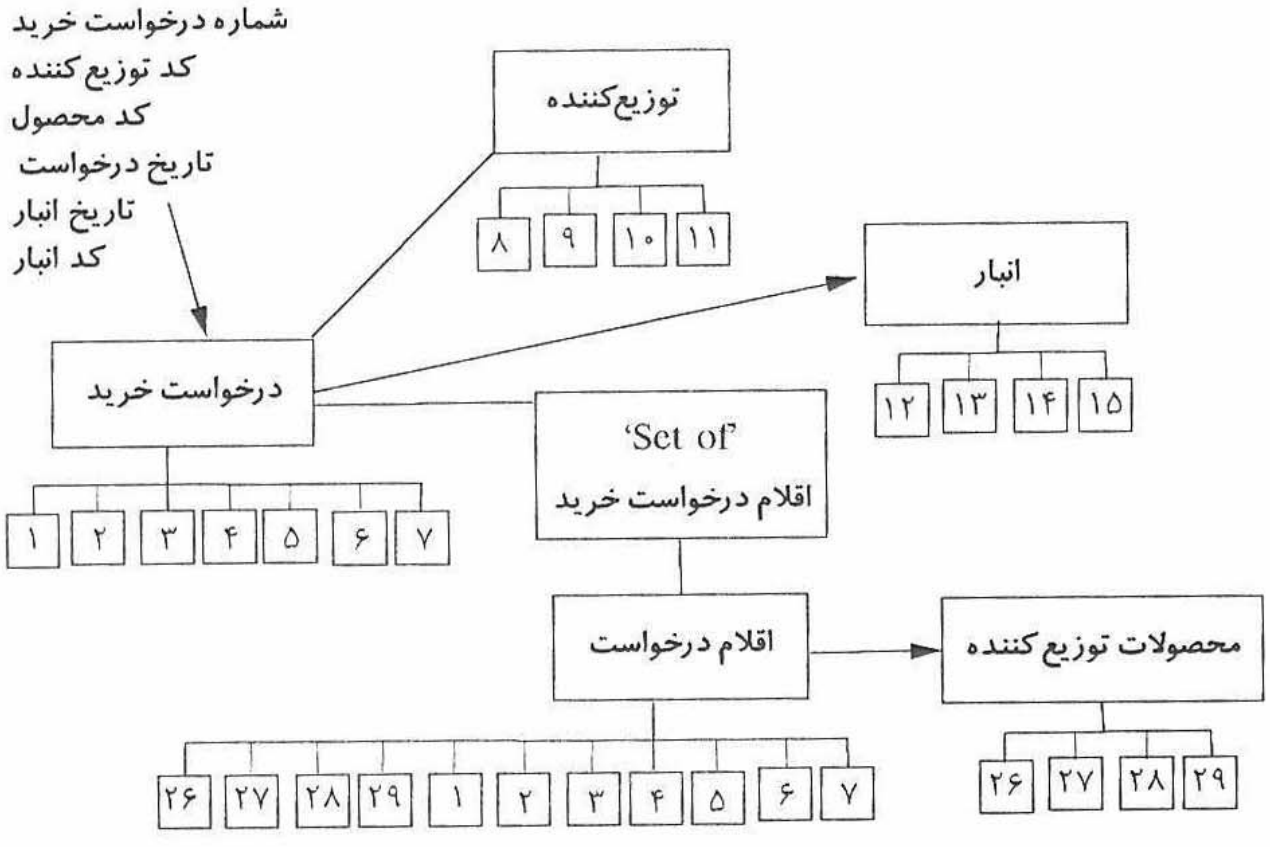
۶. اضافه کردن شرط: در مورد هر عملگر تکرار شرطی را که تا برقراری آن عمل تکرار ادامه پیدا می‌کند، نشان می‌دهیم و در این نمودار فقط در یک جا از عملگر تکرار استفاده شده است، پس شرط را برای همان عملگر تعیین و به نمودار

اضافه می‌کنیم و شرط تکرار در این جا آن است که این عمل تا آنجا که شماره درخواست عوض نشده ادامه پیدا کند.



شکل ۱۴.۱۵

۷. آخرین کار برای رسم نمودار تعیین عملیاتی است که روی موجودیت‌ها انجام می‌شود و اضافه کردن آن به نمودار می‌باشد. حال نمودار کامل را رسم می‌کنیم.



شکل ۱۴.۱۶

۱. «درخواست»	Create
۲. «شماره درخواست»	Set
۳. «تاریخ درخواست»	Set
۴. درخواست خرید به انبار	Tie
۵. درخواست خرید به توزیع‌کننده	Tie
۶. ۲= وضعیت درخواست	Set
۷. «درخواست خرید»	Write
۸. توزیع‌کننده درخواست	Read
۹. <> وضعیت توزیع‌کننده	Fail if
۱۰. ۲= وضعیت توزیع‌کننده	Set
۱۱. توزیع‌کننده	Write
۱۲. انبار درخواست خرید	Read
۱۳. ۱ <> وضعیت انبار	Fail if
۱۴. ۱= وضعیت انبار	Set
۱۵. انبار	Write
۱۶. اقلام درخواست	Create
۱۷. «کد محصول»	Set
۱۸. «شماره درخواست»	Set
۱۹. «تعداد مورد نیاز»	Set
۲۰. «تاریخ نیاز»	Set
۲۱. «کد توزیع»	Set
۲۲. «اقلام درخواست خرید» به درخواست خرید	Tie
۲۳. «اقلام درخواست خرید» به محصولات توزیع‌کننده	Tie
۲۴. ۲ وضعیت اقلام درخواست خرید	Set
۲۵. اقلام درخواست	Write
۲۶. محصولات توزیع‌کننده برای این کد محصول	Read
۲۷. ۱ <> وضعیت کد محصول	Fail if sl
۲۸. ۱= وضعیت کد محصول	Set
۲۹. محصولات توزیع‌کننده	Write

فصل پانزدهم

طراحی واسط گرافیکی

(Produce GUI Design)

اهداف فصل

کاربر امروز، انتظار دارد که با کامپیوتر از طریق صفحاتی به گفتگو بپردازد که این صفحات مانند صفحات سیستم عامل Windows پر از پنجره‌ها و Iconها و منوهای مختلف باشد. این واسط گرافیکی امکاناتی را در اختیار کاربر قرار می‌دهد که از قابلیت‌های روش‌های سنتی تعریف صفحات ورودی و خروجی فراتر می‌رود. به عنوان مثال کاربر را قادر می‌سازد تا به طور همزمان کارهای مختلفی را انجام دهد و با منابع مختلف اطلاعاتی ارتباط داشته باشد. برای این که قادر باشیم از چنین امکاناتی استفاده کنیم ابتدا باید روش کار کاربران را در این محیط بشناسیم و بدانیم کدام یک از پنجره‌ها در چه هنگامی باز باشند و کدام یک پس از استفاده از اطلاعات آنها بسته شود. البته این نکته را نیز یادآوری می‌شویم که امکانات بهتر لزوماً باعث طراحی بهتر نمی‌شود و گاهی باعث ضعف نیز می‌گردد. به علاوه برای طراحی جدید، تخصص‌های جدیدی نیز مورد نیاز می‌باشد، به ویژه از افرادی که تجربه کار در محیط گرافیکی داشته و با طراحی در این محیط آشنایی دارند باید استفاده گردد.

از همین رو است که در نسخه ۴.۳ SSADM طراحی صفحات ورودی و خروجی با استفاده از این واسط گرافیکی مورد توجه قرار گرفته و روش‌هایی ارائه داده شده تا بتوانیم براساس آن طراحی بهتری براساس خواست و نیاز کاربر امروز انجام شود. این روش‌ها نشان می‌دهد که طراحی پنجره‌ها و ارتباطات بین آنها چگونه باشد و منابع اطلاعاتی مختلف چگونه در دسترس کاربر قرار گیرد.

حال این سؤال مطرح می‌گردد که در کدام مرحله این کار بایستی صورت گیرد. SSADM پیشنهاد می‌کند که در مرحله مشخص‌سازی نیازها بعد از آن که کارکردهای اصلی تشخیص داده شدند و ورودی‌ها و خروجی‌های آنها تعیین گردید. طراحی گرافیکی صفحات ورودی و خروجی صورت گیرد زیرا طراحی در محیط گرافیکی خود می‌تواند کارکردهای جدیدی را شناسایی کند که کارکردهای جدید نیز در لیست کارکردهای وارد می‌شود اما طراحی در این مرحله معایب زیر را دارد:

۱. زمان مورد نیاز این مرحله را افزایش می‌دهد

۲. تخصص‌های مورد نیاز را افزایش می‌دهد

به همین دلیل برخی معتقد هستند که اگر امتیاز خاصی برای انجام این گام در این مرحله موجود نباشد انجام این گام می‌تواند به تعویق بیفتد و در مرحله پنج یا شش که

طراحی منطقی و یا فیزیکی صورت می‌گیرد، انجام شود.

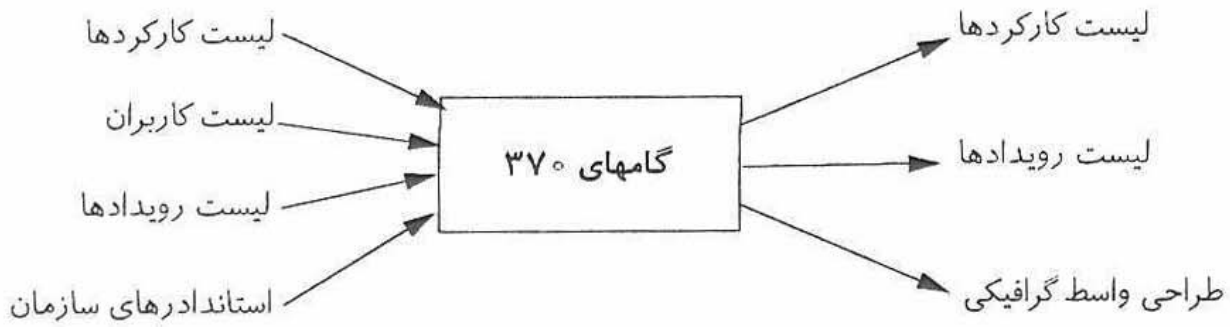
ورودی‌های این مرحله عبارتند از:

- استانداردهای طراحی صفحات گرافیکی سازمان

اگر استانداردهایی در سازمان در زمینه طراحی صفحات گرافیکی وجود دارد، از آن به منظور هماهنگی با سایر سیستم‌های موجود در سازمان استفاده می‌کنیم.

- لیست کارکردها به همراه اقلام ورودی و خروجی هر کارکرد
- لیست رویدادها
- لیست کاربران

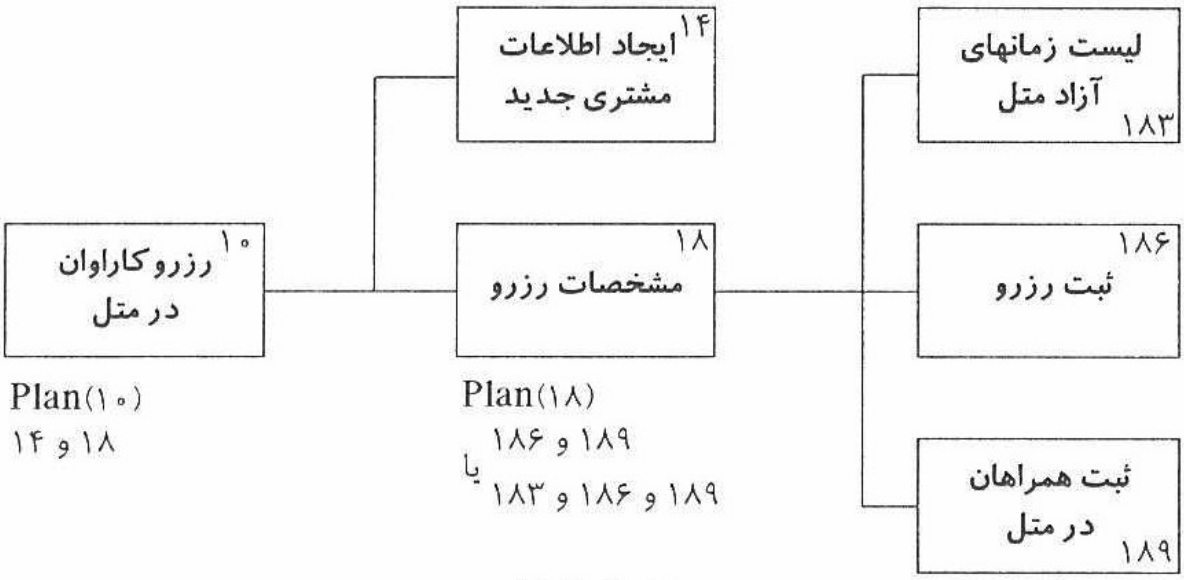
خروجی‌های این گام نیز عبارتند از:



شکل ۱۵.۱ نمودار ورودی‌ها و خروجی‌های گام ۳۷۰

مدل‌سازی فعالیت Task Modelling

مدل فعالیت کلیه کارهایی را که کاربر در زمان هر فعالیت انجام می‌دهد را تشریح می‌کند، بدین ترتیب که هر فعالیت را به کارهای کوچکی که برای انجام آن فعالیت لازم است تقسیم می کند و این کارها را در یک نمودار به عنوان نمودار مدل سلسله مراتبی نمایش می‌دهد. در شکل ۱۵.۲ کارهایی که کاربر در هنگام انجام فعالیت رزرو یک کاراوان انجام می‌دهد، نشان داده شده است و برای سادگی مطلب فرض می‌کنیم مشتری پرداخت‌ها را در همان ابتدا انجام دهد.



شکل ۱۵.۲

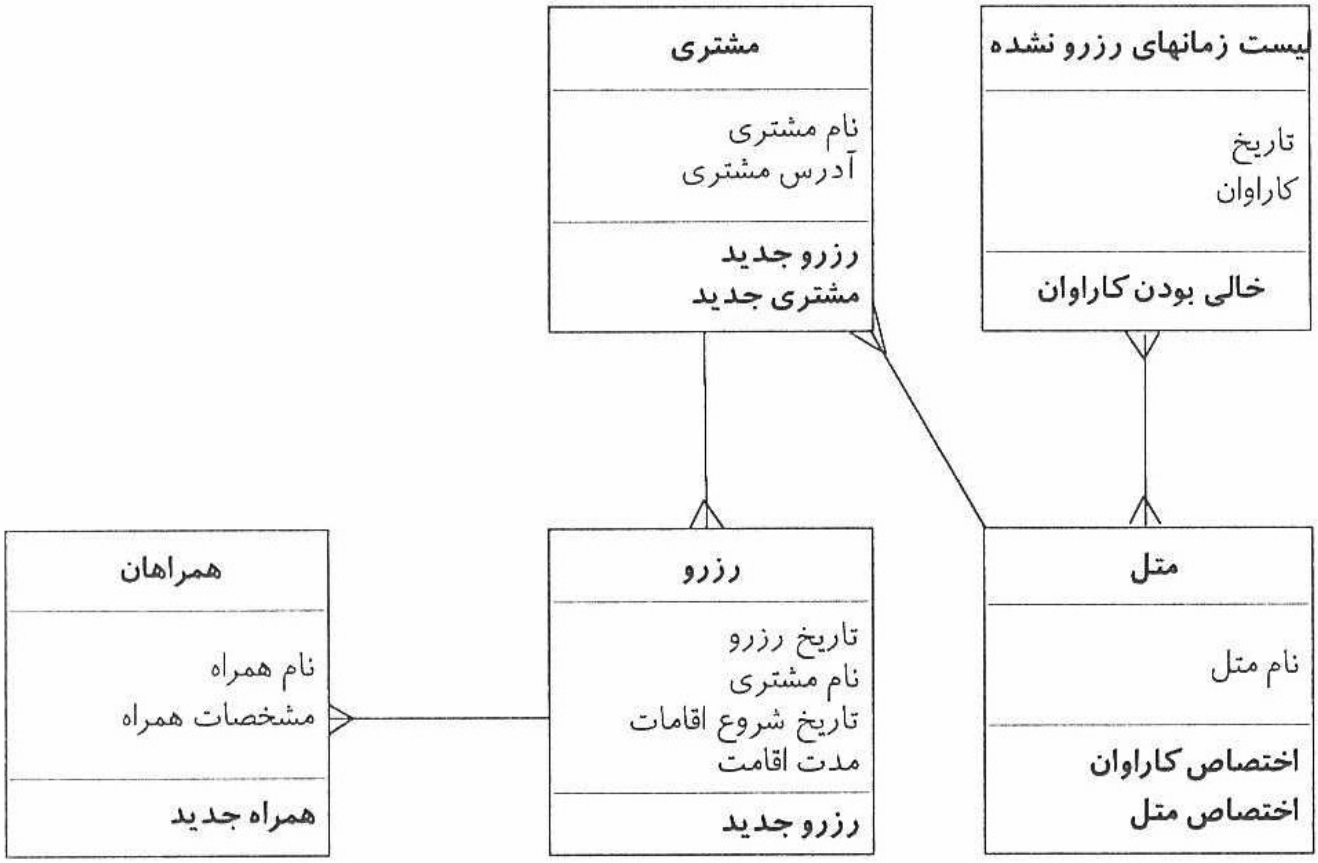
نمودار از چپ به راست خوانده می‌شود و در حاشیه هر مستطیل حالت‌های مختلفی که ممکن است اتفاق بیفتد را نشان می‌دهد. حال اگر به مثال رزرو برگردیم کاربر ابتدا مشخصات مشتری را وارد می‌کند، اگر جدید باشد (همان‌طور که گفته شد می‌تواند از مشتری بپرسد که قبلاً کاراوانی رزرو کرده است یا خیر) یا مستقیماً به تعیین مشخصات رزرو می‌پردازد. برای تعیین مشخصات رزرو کاربر می‌تواند یا ابتدا لیست تاریخ‌هایی را که کاراوان‌ها در آن زمان‌ها آزاد هستند را برای مشتری تهیه کند تا از بین این تاریخ‌ها مشتری یکی را انتخاب کند یا می‌تواند مستقیماً به ثبت رزرو و سپس ثبت اطلاعات همراهان بپردازد.

رسم چنین نمودارهایی به تحلیل‌گر کمک می‌کند که بتواند مراحل مختلف انجام یک فعالیت را درک کند. برای هر یک از فعالیت‌ها می‌توان چندین نمونه از حالت‌های واقعی را در نظر گرفت و این نمونه را سناریو فعالیت (Task Scenario) می‌نامند و توسط این نمونه که حالت‌های واقعی و عملی را نشان می‌دهند با مدل ایجاد شده مقایسه نمود. اشکالات مدل ایجاد شده را پیدا کرد. نمودار مدل سلسله مراتبی فعالیت در دهۀ ۵۰ وقتی که جامعه‌شناسان می‌خواستند که فعالیت‌های کاری افراد را تجزیه و تحلیل کنند، ساخته شد.

مدل اشیاء کاربر User Object Model

مدل اشیاء کاربر عبارت است از موجودیت‌هایی که کاربر تشخیص می‌دهد و همچنین کارهایی که بنابر تشخیص کاربر باید روی هر موجودیت انجام پذیرد تا یک

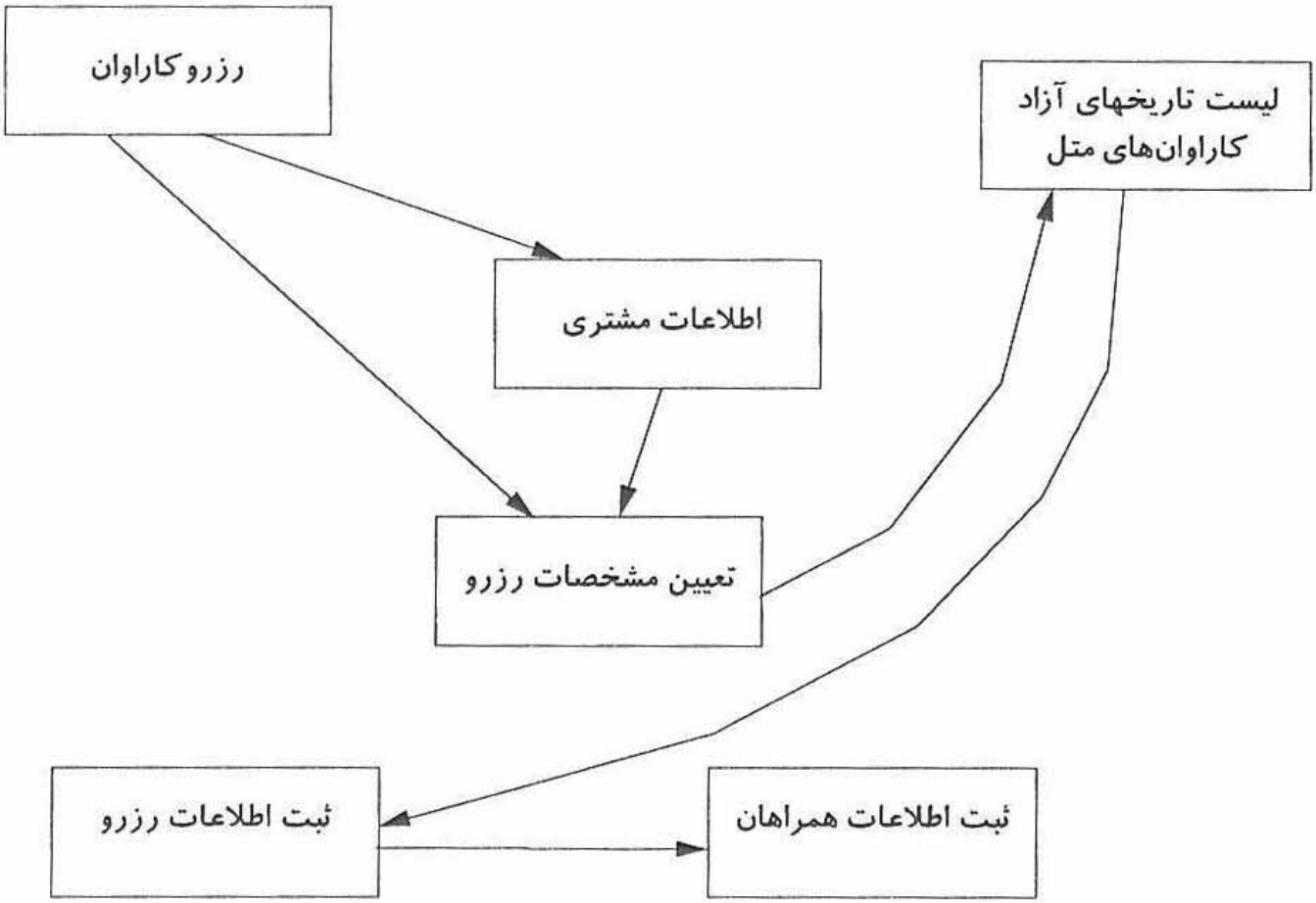
فعالیت انجام گردد. اگر بار دیگر مثال رزرو را در نظر بگیریم از نظر کاربر موجودیت‌های مشتری، رزرو، همراهان و لیست زمان‌های رزرو نشده کاراوان‌ها موجود است که در شکل ۱۵.۳ این موجودیت‌ها و اقلام اطلاعاتی آنها و کارهایی که بر روی هر یک از موجودیت‌ها صورت می‌گیرد نشان داده شده است.



شکل ۱۵.۳

مدل پیمایش کارکردی Function Navigation Model

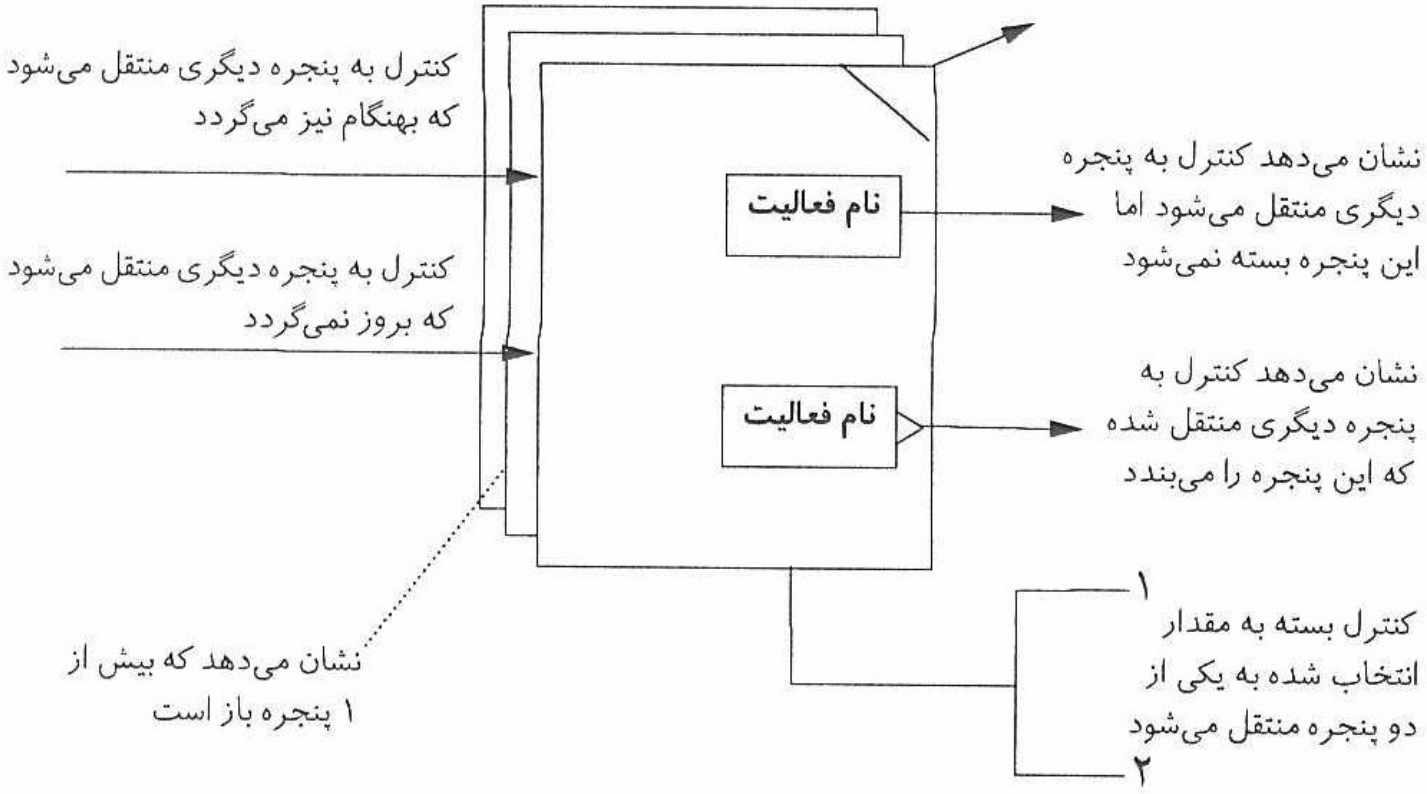
مدل سلسله مراتبی نشان می‌دهد که برای انجام فعالیت چه کارهایی باید انجام شود. اما مدل پیمایش کارکردها نشان می‌دهد که برای انجام فعالیت‌ها چگونه باید حرکت کنیم و ترتیب و توالی کارها چگونه است؟ در مثال رزرو ابتدا صفحهٔ ورود اطلاعات رزرو ظاهر می‌شود، سپس از مشتری پرسیده می‌شود که آیا قبلاً رزرو داشته است یا خیر؟ در صورتی که جواب مشتری منفی باشد، صفحهٔ دیگری ظاهر می‌شود و اطلاعات مشتری ثبت می‌گردد، سپس به صفحهٔ اطلاعات رزرو برمی‌گردیم و لیست زمان‌های آزاد کاروان‌ها را مشخص می‌کنیم که کاربر زمان مورد نظر را انتخاب می‌کند و بعد مشخصات رزرو را ثبت می‌کنیم و در مرحله بعد مشخصات همراهان را ثبت می‌نماییم. شکل ۳-۱۵ نمودار این مدل است که چگونگی توالی کارهای مرتبط با این فعالیت را نشان می‌دهد.



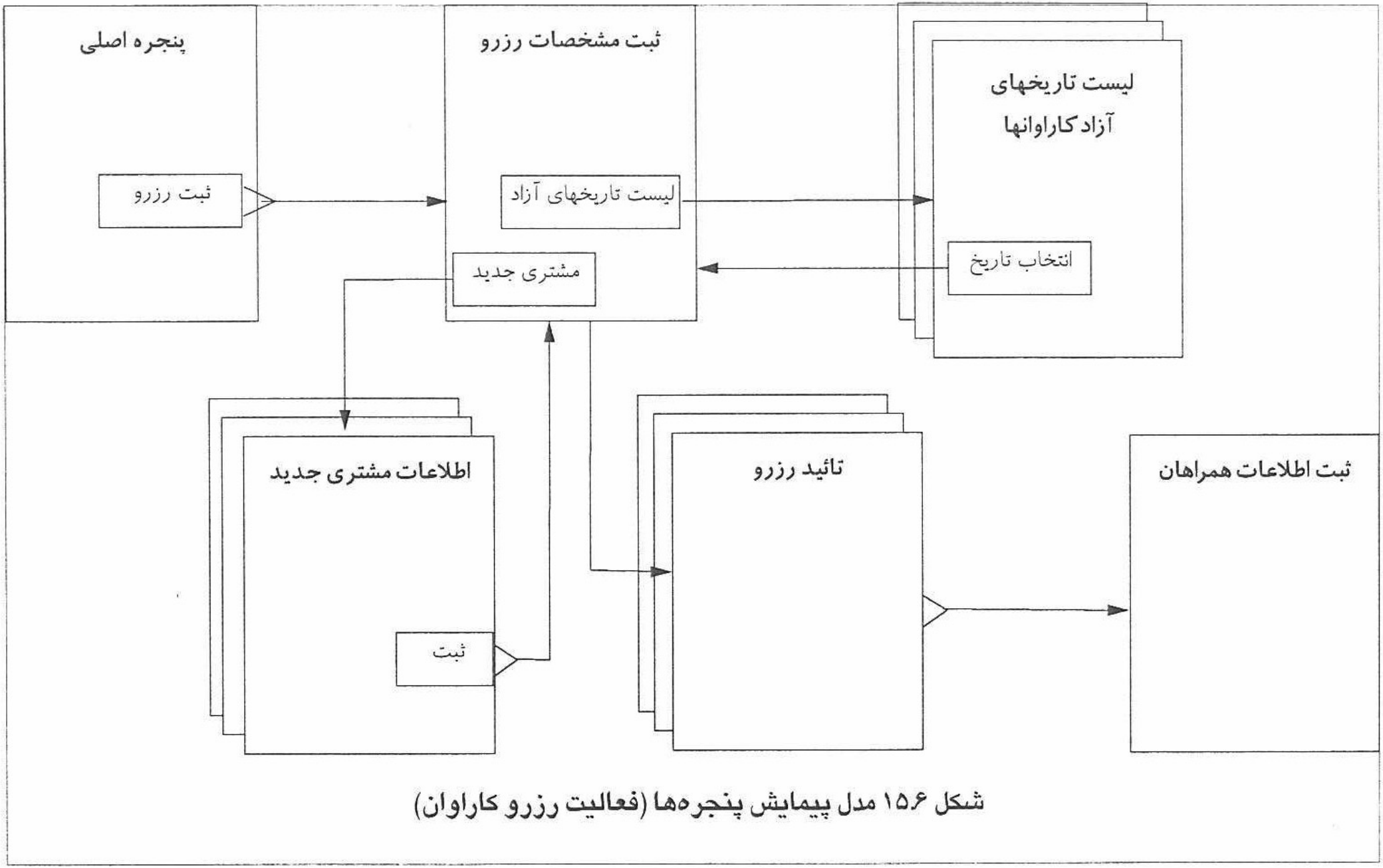
شکل ۱۵.۴

مدل پیمایش پنجره‌ها Window Navigation Model

تنها کاری که حالا باقی می‌ماند تبدیل مدل پیمایش کارکردها به طراحی صفحات گرافیکی می‌باشد. این کار با ایجاد مدل پیمایش پنجره‌ها صورت می‌گیرد. علامت‌هایی که در تهیه این مدل از آن استفاده می‌شود در شکل ۱۵-۵ نشان داده شده است. هر پنجره یکی از کارکردها (کارهای مختلف فعالیت) را نشان می‌دهد و اقلام اطلاعاتی هر کارکرد در یک پنجره به نمایش درمی‌آید. در شکل ۱۵-۶ مدل پیمایش پنجره‌ها برای فعالیت رزرو مشخص شده است.



شکل ۱۵.۵ علامت‌هایی که در مدل پیمایش پنجره‌ها از آن استفاده می‌شود



شکل ۱۵۶ مدل پیمایش پنجره‌ها (فعالیت رزرو کاراوان)

فصل شانزدهم

آخرین گام در مرحلهٔ سوم

اهداف فصل

در این فصل مهم‌ترین اهداف عبارتند از:

۱. اطمینان از عملکرد درست و رسیدن به اهداف مربوطه

مهمترین وظیفه در این مرحله نهایی کردن لیست کارکردها می‌باشد و باید اطمینان یابیم که کلیه کارکردها در نظر گرفته شده‌اند و چیزی از قلم نیفتاده است و هر کارکرد نیز زمان پاسخ‌دهی مناسب را دارا می‌باشد.

۲. جمع‌آوری و تهیه مشخصات نیازها:

در این گام تمام مستندات ایجاد شده جمع‌آوری گردیده و تمام آنها امضاء می‌شوند.

لیست کارکردها

لیست کارکردها از این جهت که شامل کلیه پردازش‌های سیستم جدید است، بسیار مهم بود و یکی از اصلی‌ترین مستندات این مرحله می‌باشد. برای هر کارکرد مستندات زیر را تهیه می‌کنیم:

۱. شرح کارکرد:

شامل شرح هر کارکرد و جزئیات پردازش مربوطه و زمان پاسخ‌دهی کارکرد می‌باشد.

۲. صفحات ورودی و گزارشات:

این قسمت شامل کلیه اقلام اطلاعاتی لازم و طرح هر صفحهٔ ورودی و گزارش می‌باشد که مطابق استانداردهای کاربران تنظیم می‌گردد.

۳. نمودار مسیر دسترسی:

در این قسمت مسیرهای دستیابی به اطلاعات مورد نظر برای هر کارکرد تهیه می‌گردد. البته می‌توان برای پرس‌وجوهای ساده این نمودار را تهیه نکرد.

۴. شرح مسیر دسترسی:

شامل شرح فعالیت‌های مسیر دسترسی می‌باشد.

۵. تهیه لیست خطاها:

شامل کلیه خطاهایی می‌باشد که در زمان اجرای کارکرد با آن مواجه می‌شویم.

در بسیاری از موارد می‌توان مشخصات برنامه‌ها را مستقیماً از روی اطلاعات کارکرد به شرح فوق تهیه کرد.

گام آخر

در این گام دو وظیفه اصلی وجود دارد، وظیفه اول آن است که لیست نیازمندی‌ها را مجدداً مرور کنیم و نیازمندی‌های کارکردی و غیرکارکردی، را بررسی کنیم و با بررسی شرح کارکرد از مطابقت عملکرد کارکرد با نیازمندی‌ها مطمئن شویم. اطمینان از برآورده شدن نیازمندی‌های غیرکارکردی مشکل‌تر است. فرض کنید که در لیست نیازمندی‌ها تصمیم بر این گرفته شده است که اطلاعات به صورت رمز شده (کد شده) در پایگاه اطلاعاتی وجود داشته باشد و طبیعی است که در طراحی سیستم ما فقط به پردازش‌های سیستم توجه داشته باشیم و برای رمزگذاری و کد کردن اطلاعات از نرم‌افزارهای موجود در بازار استفاده نماییم. بنابراین چنین نیازی در مرحله انتخاب نرم‌افزار در نظر گرفته می‌شود.

اما وظیفه مهم در این گام بررسی زمان سرویس‌دهی سیستم می‌باشد که به معنی آن است که زمان پاسخ‌دهی کارکردهای لحظه‌ای چه مقدار باشد؟ و چه مدت طول می‌کشد تا اطلاعات از پایگاه داده‌ها استخراج گرددد و به کاربر نشان داده شود؟ حداکثر زمان پاسخ‌دهی در این مرحله تعیین می‌گردد. زمان پاسخ‌دهی یکی از پارامترهای مهم برای تولیدکنندگان سیستم می‌باشد، زیرا یکی از معیارهای تأیید سیستم است.

در تعیین زمان سرویس‌دهی سیستم باید مسائل زیر را رعایت کنیم:

۱. در گام ۳۳۰ زمان سرویس‌دهی را برای هر کارکرد تعیین کردیم، در عمل می‌توان احتمال بسیار داد که برای تعداد زیادی از کارکردها یک زمان پاسخ‌دهی در نظر گرفته شده باشد. در حالی که نوع کارکردها با هم متفاوت است پس زمان پاسخ‌دهی آنها نیز متفاوت می‌باشد. پس بهتر است کارکردها را دسته‌بندی کنیم. بدین صورت که کارکردهای مهم (آنهایی که کاربرد بیشتری دارند) را در یک دسته قرار دهیم و کارکردهایی را که گاهاً مورد استفاده قرار می‌گیرند در دسته دیگری قرار گیرند. و برای هر گروه زمان پاسخ‌دهی خاص خود را انتخاب کنیم و اگر این دسته‌بندی قبل از آغاز پروژه صورت گیرد بسیار بهتر است. با این حال ممکن است کارکردهایی به دلیل شرایط خاص نتواند زمان پاسخ‌دهی مناسب را داشته باشد، بنابراین بهتر است برای آن کارکرد راهی برای رسیدن به زمان مناسب تعیین کرد.

۲. باید مشخص کنیم که مقصود از زمان سرویس‌دهی چیست؟ آیا صرفاً زمان

بازیابی اطلاعات می‌باشد یا زمان ارزیابی اطلاعات ورودی هم در آن ملحوظ است و یا حتی شامل زمان به روزرسانی و ایجاد اطلاعات نیز می‌گردد، پس بهتر است در مورد زمان سرویس‌دهی مشخص‌تر صحبت کرد و اگر می‌گوییم زمان سرویس‌دهی اطلاعات، زیر ۳ ثانیه می‌باشد مطلب بسیار کلی است بهتر است این طور بگوییم که کارکردها در هشتاد درصد موارد، زیر ۲ ثانیه به کاربر پاسخ می‌دهد و در ۱۵ درصد موارد بین ۲ تا ۵ ثانیه و در سایر موارد زمان سرویس‌دهی بالای ۵ ثانیه می‌باشد.

۳. آنچه در گام ۳۳۰ به عنوان زمان سرویس‌دهی در نظر گرفته شده است بدون در نظر گرفتن قیمت سخت‌افزارها و نرم‌افزارهای مورد نیاز می‌باشد و طبیعتاً در این مرحله کاربر دوست دارد که حداقل زمان سرویس‌دهی را داشته باشد اما با مطرح شدن قیمت سخت‌افزارها و نرم‌افزارهای لازم ممکن است کاربران در تصمیم خود تجدید نظر کنند. البته این مورد را می‌توان در مرحله بعد که سخت‌افزارها انتخاب می‌گردند تعیین کرد که ممکن است این کار باعث طولانی‌تر شدن سایر مراحل می‌گردد.

جمع‌آوری مشخصات نیازمندی‌ها

در اینجا مستنداتی را که مشخصات نیازمندی‌ها را تشکیل می‌دهند جمع‌آوری می‌کنیم که شامل موارد زیر است:

۱. لیست نیازمندی‌ها: در این مرحله تمام نیازمندی‌های کارکردی بایستی توسط کارکردهای متناسب پوشش داده شده باشد و آن نیازمندی‌های غیرکارکردی که منتظر انتخاب سخت‌افزار و نرم‌افزارها باقی می‌ماند باید مشخص شده باشد.

۲. مدل منطقی داده‌ها برای سیستم جدید که شامل مستندات زیر می‌باشد تهیه می‌گردد:

- نمودار ساختار منطقی داده‌ها
- شرح موجودیت‌ها و اقلام اطلاعاتی آنها
- شرح روابط موجودیت‌ها

۳. لیست کارکردها که درباره آن بحث شد.

۴. کاربران سیستم جدید و نقش و وظایف هر یک از آنها

مستندات فوق جزئیات سیستم جدید را مشخص می‌کند و باید در اختیار تیم طراحی و برنامه‌نویسی قرار گیرد تا بتواند بقیه راه را ادامه دهد.

و البته مستندات دیگری از قبیل لیست رویدادها بهتر است مستندات فوق را همراهی کند تا تیم از زاویه دیگری نیز بتواند سیستم را بررسی کند.

فصل هفدهم

تعیین گزینه‌های فنی و انتخاب گزینه مناسب

(Technical System Option)

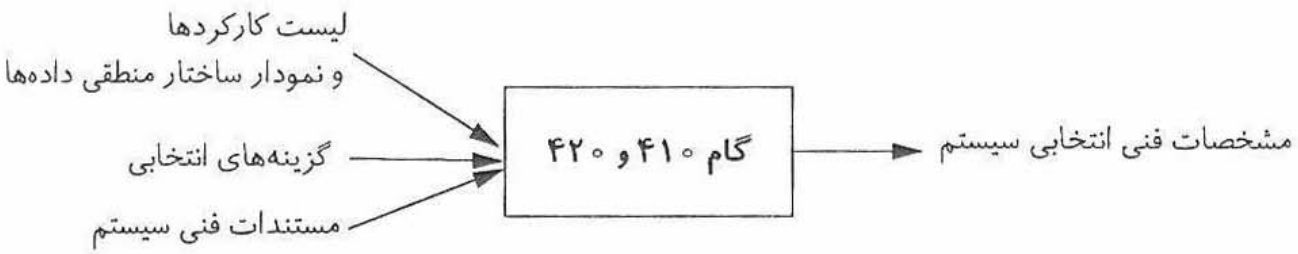
اهداف فصل

در این فصل جزئیات گام‌های مرحله چهارم از مراحل SSADM تشریح می‌گردد. در این مرحله نرم‌افزارها و سخت‌افزارهای مختلفی که سیستم به کمک آنها می‌تواند پیاده‌سازی گردد بررسی شده و در نهایت یک الگوی مناسب از جهت سخت‌افزار و نرم‌افزار جهت پیاده‌سازی سیستم انتخاب می‌شود. تعیین گزینه‌های فنی سیستم تا حد زیادی به تعیین گزینه‌های تجاری شباهت دارد. در این مرحله در گام ۴۱۰ گزینه‌های فنی سیستم تعیین می‌شود و در گام ۴۲۰ یکی از این گزینه‌ها توسط کمیته راهبری پروژه جهت پیاده‌سازی سیستم انتخاب می‌گردد.

قبل از انتخاب گزینه مناسب، کمیته راهبری باید آنچه سیستم قرار است انجام دهد، آگاهی یابد. کمیته می‌تواند این اطلاعات را با مراجعه به مستندات مرحله تعیین مشخصات سیستم یعنی عمدتاً از لیست کارکردها و ساختار منطقی داده‌ها به دست آورد.

زمان پاسخ‌دهی و حجم اطلاعات سیستم دو پارامتر بسیار مهم و تعیین‌کننده برای انتخاب سخت‌افزار و نرم‌افزار سیستم می‌باشد، زیرا نرم‌افزارها و سخت‌افزارهای لازم بایستی با توجه به حجم اطلاعاتی برآورد شده و کارآیی لازم را داشته باشند و بتوانند در زمان تعیین شده به اطلاعات دستیابی پیدا کنند. مهمترین ورودی‌های این مرحله، گزینه انتخابی و مشخصات نیازمندی‌ها براساس راه‌حل انتخابی ایجاد شده و سایر مستندات فنی سیستم از قبیل استراتژی قسمت کامپیوتر در انتخاب مدیریت بانک اطلاعاتی و سخت‌افزارها میباشد. سایر ملاحظاتی که در تعیین گزینه‌های فنی می‌تواند نقش داشته باشند، عبارت است از:

- وجود مکان مناسب جهت نصب سخت‌افزار پیشنهادی خصوصاً اگر تجهیزات سخت‌افزاری مربوط به کامپیوترهای بزرگ باشد.
- داشتن امنیت لازم خصوصاً جهت کاربردهای خاص از قبیل کاربردهای نظامی
- جود پشتیبانی لازم برای نرم‌افزار و سخت‌افزار انتخابی
- سهولت تغییر در نرم‌افزارها و سخت‌افزارها
- وجود تخصص‌های لازم جهت بهره‌گیری از سخت‌افزارها و نرم‌افزارها



شکل ۱۷.۱ ورودی‌ها و خروجی‌های گام‌های ۴۱۰ و ۴۲۰

گام ۴۱۰ تعیین گزینه‌های فنی سیستم

در این گام حداکثر شش گزینه برای سیستم در نظر گرفته می‌شود و پس از هماهنگی با کاربران حداکثر ۲ یا ۳ گزینه برای بررسی نهایی و اتخاذ تصمیم به کمیته راهبری پروژه ارسال می‌گردد. در این گام راه‌های مختلف توسعه سیستم در محیط‌های متفاوت را در نظر می‌گیریم و آن را از زوایای مختلف با یکدیگر مقایسه می‌کنیم. برای تعیین هر گزینه می‌توان عوامل زیر را در نظر گرفت:

۱. محیط

در این جا سخت‌افزارها و نرم‌افزارهای ممکن و در دسترس با توجه به پارامترهای زیر مقایسه می‌شوند.

- امکان مشارکت اطلاعات یا سایر سیستم‌ها
- امکان تبادل اطلاعات سیستم با تکنولوژی‌های جدیدتر (مطابقت با استانداردهای مطرح)
- امکان کار ۲۴ ساعته سیستم و نیازمندی‌های آن
- امکان شروع به کار سریع و برگرداندن سریع اطلاعات در زمان بروز خرابی در سیستم
- امکان توسعه سیستم با توجه به زمان‌بندی مورد نظر
- هماهنگی با سایر نرم‌افزارها و سخت‌افزارهای موجود در سازمان

در زمان تهیه گزارش امکان‌سنجی نیازمندی‌های اصلی فنی سیستم از قبیل این که روی چه نوع کامپیوترهایی (Pc-Mini-Main) سیستم توسعه یابد و یا چه نوع بانک اطلاعاتی (سلسله مراتبی ـ رابطه‌ای ـ شبکه‌ای و غیره) استفاده گردد، مشخص می‌شود و در این مرحله فقط بین سخت‌افزارها و نرم‌افزارهای موجود و در دسترس مقایسه صورت می‌گیرد. از آنجایی که در بسیاری از موارد (خصوصاً در مورد کامپیوترهای بزرگ) به

دلیل سیاست‌های خاص سازمان استفاده از یک نرم‌افزار یا سخت‌افزار خاص اجتناب‌ناپذیر است بنابراین در بسیاری از موارد بیش از یک راه‌حل، موجود نیست.

ـ بار مالی

در این جا گزینه‌های مختلف از جهت هزینه‌ها و بار مالی که برای سازمان دارند، با یکدیگر مقایسه می‌شوند.

ـ تأثیرگذاری بر سازمان

می‌توان تأثیر هر گزینه بر سازمان و افراد آن را از جهت دوره‌های آموزشی مورد نیاز و مدت و هزینه آنها جهت هر گزینه برآورد و مقایسه کرد.

ـ زمان

می‌توان تأثیر هر گزینه بر زمان توسعه سیستم را مشخص کرده و سپس با هم مقایسه کرد.

پس از تعیین گزینه‌های مختلف این گزینه‌ها جهت انتخاب به کمیته راهبری پروژه ارائه می‌گردد تا کمیته در مورد آن بررسی لازم را صورت داده و گزینه مناسب را انتخاب کند. اما دو نکته در این جا می‌تواند مورد اهمیت واقع شود.

ـ اعلام مناقصه

این مرحله موقعیت خوبی برای به مناقصه گذاردن بقیه پروژه می‌باشد زیرا مشخصات پروژه و نیازمندی‌های آن در مراحل قبلی به خوبی شناخته شده و مستندات مربوطه ایجاد گردیده است و به راحتی به شرکتی می‌توان ارائه داد تا آن شرکت سیستم را مطابق مستندات مربوطه ایجاد نماید. بنابراین کمیته راهبری پروژه نیز علاوه بر انتخاب گزینه راجع به چنین موضوعاتی نیز می‌تواند تصمیم لازم را اتخاذ کند.

۲. استفاده از برنامه‌های نرم‌افزاری آماده یا Package

در این مرحله مزایا و معایب استفاده از یک Package یا «نرم‌افزار آماده» را نیز می‌توان مورد نظر قرار داد این نوع نرم‌افزارها معمولاً از قابلیت‌های توسعه سریع ، ارزان و اطمینان برخوردار هستند و می‌توان زمان و هزینه تغییر یک نرم‌افزار آماده را با زمان و هزینه سایر مراحل مقایسه کرد ولی اگر اصولاً استفاده از یک نرم‌افزار آماده اجباری باشد، این مسأله بهتر است در مراحل قبل مورد نظر قرار گیرد ولی به عنوان یک راه‌حل

می‌توان آن را با سایر گزینه‌ها مقایسه کرد.

گام ۴۲۰ انتخاب گزینه فنی

در این گام کمیته راهبری پروژه با توجه به گزینه‌های موجود یکی را انتخاب می‌کند شاید بهتر باشد گزینه فنی انتخابی را به اطلاع گروه بزرگ‌تری از کاربران رساند و از نظرات آنها استفاده کرد.

پس از تعیین گزینه نهایی اگر در مورد مراحل مختلف پیاده‌سازی سیستم نیز بررسی‌هایی صورت گیرد بسیار خوب است این بررسی می‌تواند در موارد زیر صورت گیرد:

۱. استراتژی پیاده‌سازی چطور باشد. آیا سیستم به یکباره عملیاتی گردد و یا به صورت تدریجی و مرحله‌ای و پس از تست و آزمایش هر مرحله، مرحله بعدی پیاده‌سازی گردد.
۲. نحوه تبادل اطلاعات موجود بررسی گردد و پاسخ سؤالات زیر مشخص شود.
   - تعیین شود که چه میزان از اطلاعات موجود مورد نیاز سیستم جدید است.
   - منابع این اطلاعات در کجا قرار دارد.
   - نحوه تبادل اطلاعات به چه صورت باشد (دستی، ماشینی)
۳. محدودهٔ تست سیتم و نحوه تست به چه صورت باشد و کاربران چگونه سیستم را می‌توانند تست کنند.
۴. نحوه آموزش سیستم و مراحل آن تدوین گردد.

چنین بررسی‌هایی می‌تواند پیاده‌سازی سیستم را سهل‌تر سازد.

فصل هجدهم

طراحی منطقی (Logical Design)

اهداف فصل

این فصل به تشریح جزئیات گام‌های مرحله پنجم از مراحل SSADM می‌پردازد. در این مرحله جزئیات پردازش‌های بروزرسانی و پرس‌وجو مستند می‌شود. این مستندات براساس لیست کارکردها و نمودار تأثیر رویدادها و نمودار مسیر دسترسی شکل می‌گیرد و مشخص می‌سازد که اطلاعات از کدام موجودیت استخراج گردیده و یا در هنگام بروزرسانی کدام یک را بهنگام می‌سازد. چنین مستنداتی موجب می‌گردد تا در مرحله طراحی فیزیکی مشخصات برنامه‌ها راحت‌تر تهیه گردد.

این مرحله از دو گام ۵۱۰ (تعریف پردازش‌های بروزرسانی) ۵۲۰ (تعریف پردازش‌های پرس‌وجو) تشکیل می‌گردد.

در برنامه‌ریزی مرحله پنج (طراحی منطقی) مدیریت پروژه مزایای انجام این مرحله را با زمان و هزینه‌ای که برای اتمام آن صرف می‌گردد، مقایسه کرده و در صورتی که توجیهات لازم وجود داشت این مرحله را انجام دهد.

مهم‌ترین مزیت این مرحله ایجاد قابلیت نقل و انتقال نرم‌افزار Protability می‌باشد که امکان طراحی بدون در نظر گرفتن امکانات سخت‌افزاری و نرم‌افزاری را موجب می‌گردد.

اگرچه تغییر سخت‌افزار یا مدیریت بانک اطلاعاتی یا محیط توسعه می‌تواند تا حدودی باعث تغییرات و طراحی مجدد گردد ولی این تغییرات کلی نمی‌باشد. مزیت دیگر اتمام این مرحله تهیه مستنداتی است که باعث می‌گردد تا تهیه مشخصات برنامه‌ها در مرحله بعد آسان‌تر شود.

پارامتر دیگری که باید در نظر گرفته شود هزینه است که علاوه بر هزینه مستقیم انجام اقدامات این مرحله، هزینه‌های بروزرسانی مستندات در صورت ایجاد تغییرات در کارکردها را نیز باید در نظر داشت زیرا بدون به روزرسانی مستندات این مرحله کارآیی چندانی نخواهد داشت.

نکته آخری که در تصمیم‌گیری برای اجرای اقدامات این مرحله یا عدم اجرای آن باید در نظر داشت، محیط توسعه و برنامه‌سازی می‌باشد. اگر محیط توسعه برنامه‌ها محیط غیررویه‌ای (Non-Procedural) باشد یعنی به طور مثال از مدیریت بانک اطلاعاتی استفاده کنیم که رابطه‌ای باشد و معمولاً امروزه نیز چنین است، نمودار رویدادها و نمودار مسیر دسترسی خود ابزارهای خوبی برای مشخص‌سازی پردازش‌ها

بروزرسانی و پرس‌وجو می‌باشد و صرفاً هنگامی که از یک محیط رویه‌ای (به طور مثال مدیریت بانک اطلاعاتی سلسله مراتبی) استفاده شود چون ترتیب و توالی پردازش‌ها توسط نمودارهای فوق دقیقاً مشخص نمی‌گردد بهتر است که این مرحله انجام گردد تا مرحله طراحی فیزیکی با سهولت بیشتری صورت گیرد.

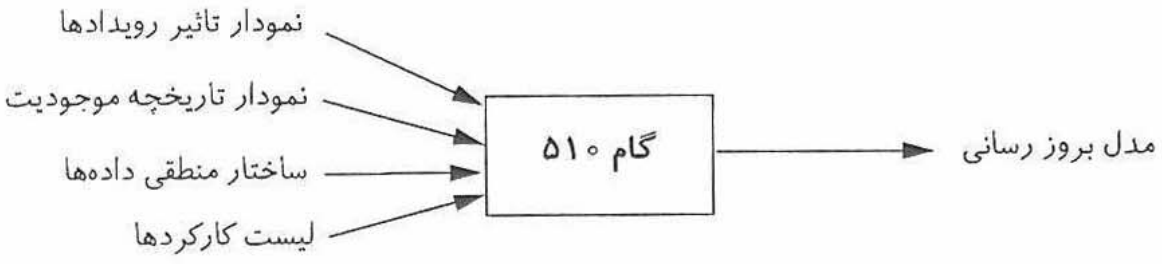
گام ۵۱۰ تعریف مشخصات پردازش‌های به روزرسانی (ساخت مدل بروزرسانی)

همان‌گونه که اشاره رفت نمودار تطابق اثر رویدادها ابزار مناسبی برای نشان دادن ترتیب و توالی مراحل یک پردازش بروزرسانی نمی‌باشد به همین دلیل از نمودارها و علایم روش جکسون استفاده می‌گردد تا بتوان توالی این مراحل را نشان داد.

بنابراین با توجه به این علائم نمودار تطابق اثر رویدادها را تبدیل به مدل بروزرسانی پردازش می‌کنیم. در ابتدا فرض می‌شود که نمودار ساختار منطقی داده‌ها پیاده‌سازی گردیده است و چنین ساختار منطقی وجود دارد و یک بانک اطلاعاتی براساس این مدل ساخته شده است.

علایم جکسون که برای ساخت این مدل به کار می‌رود همان علایمی است که در ساختار تاریخچه زندگی موجودیت و ایجاد ساختارهای ورودی و خروجی از آنها استفاده می‌گردد.

هر مدل پردازش تأثیرات یک رویداد را نشان می‌دهد و هر مستطیل انتهایی، معرف یک مرحله از کل پردازش است.



شکل ۱۸.۱ نمودار ورودی‌ها و خروجی‌های گام ۵۱۰

برای ساخت مدل بروزرسانی کارهای زیر را در مورد هر رویداد انجام می‌دهیم.

۱. نمودار تطابق اثر رویداد را در نظر می‌گیریم

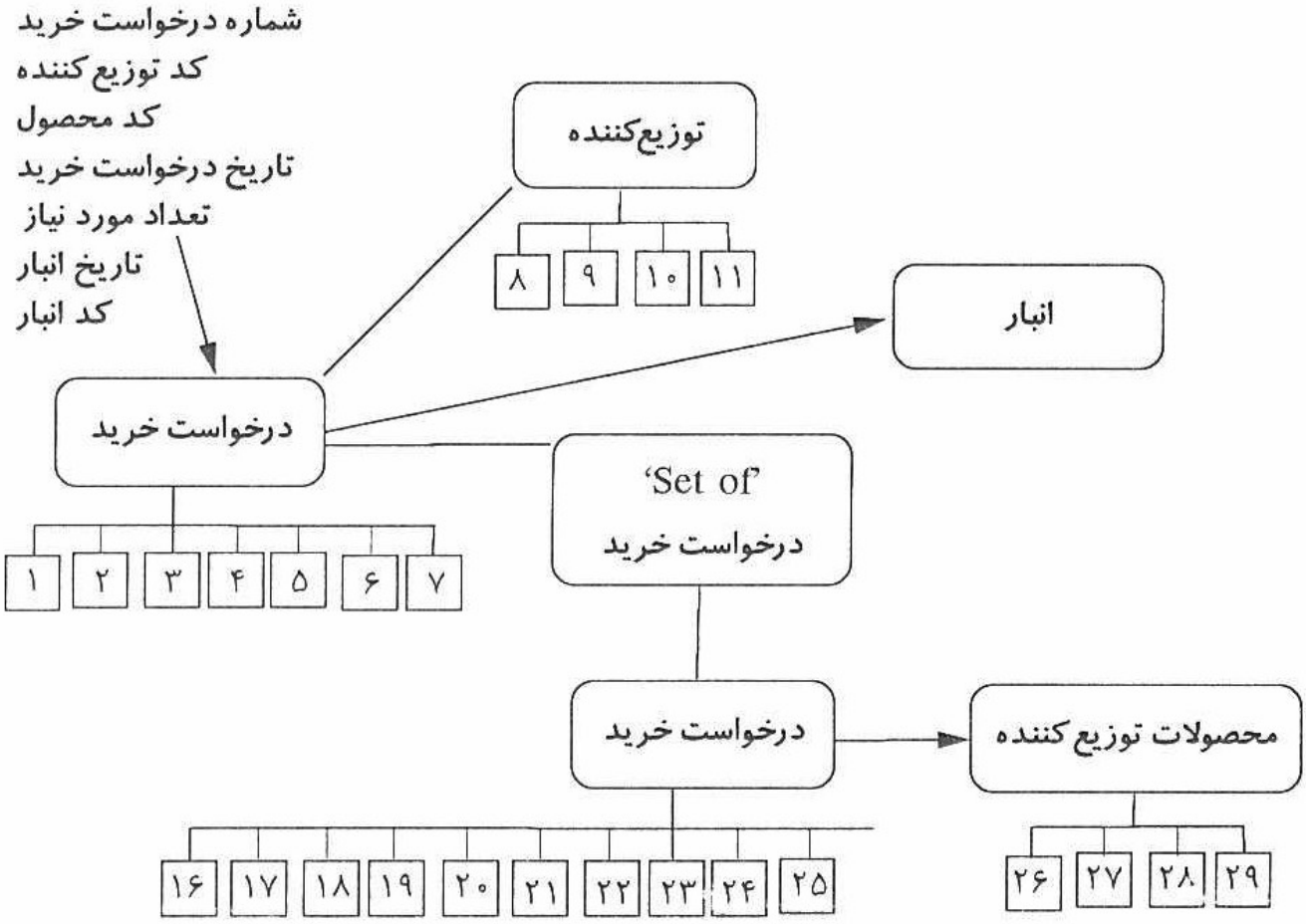
نمودار تطابق اثر رویداد، اثرات یک رویداد را بر موجودیت‌های بانک اطلاعاتی

منطقی (ساختار منطقی داده‌ها)، نشان می‌دهد. چون غرض از ایجاد مدل بروزرسانی، تعیین جزئیات پردازش‌هایی می‌باشد که بر اثر رویداد ایجاد می‌شود پس مدل با نمودار تطابق اثر رویداد باید کاملاً هماهنگ باشد.

برای مثال رویداد «ایجاد درخواست خرید» را در نظر می‌گیریم. ورودی‌های این رویداد به شرح زیر می‌باشد.

- شماره درخواست
- کد فروشنده
- کد اقلام
- تاریخ درخواست
- تعداد مورد نیاز
- تاریخ نیاز به اقلام

چگونگی رسم نمودار تطابق اثر این رویداد در فصل ۱۴ تشریح گردیده است و نمودار آن به صورت زیر می‌باشد.



شکل ۱۸.۲ نمودار تطابق اثر رویداد «ایجاد درخواست خرید»

۱. «درخواست» (ایجاد درخواست Create

۲. «شماره درخواست» Set

(تعیین شماره درخواست)

۳. «تاریخ درخواست» Set

۴. «درخواست خرید» به انبار Tie

(ایجاد رابطه بین درخواست خرید و انبار)

۵. «درخواست خرید» به توزیع‌کننده Tie

ایجاد رابطه بین درخواست خرید و توزیع‌کننده

۶. ۲= وضعیت درخواست Set

۷. «درخواست خرید» Write

ایجاد رکورد مربوطه در جدول درخواست‌ها

۸. «توزیع‌کننده» Read

خواندن اطلاعات توزیع‌کننده

۹. ۲<> وضعیت توزیع‌کننده Fail if

۱۰. ۲= وضعیت توزیع‌کننده Set

۱۱. وضعیت توزیع‌کننده Write

۱۲. «انبار درخواست خرید» Read

خواندن اطلاعات انبار مربوطه

۱۳. ۱<> وضعیت انبار Fail if SI

اگر وضعیت انبار درست نباشد پیغام خطا صادر می‌گردد)

۱۴. ۱= «وضعیت انبار» Set

۱۵. «وضعیت انبار» Write

ثبت وضعیت جدید انبار

۱۶. اقلام درخواست Create

(ایجاد اقلام درخواست)

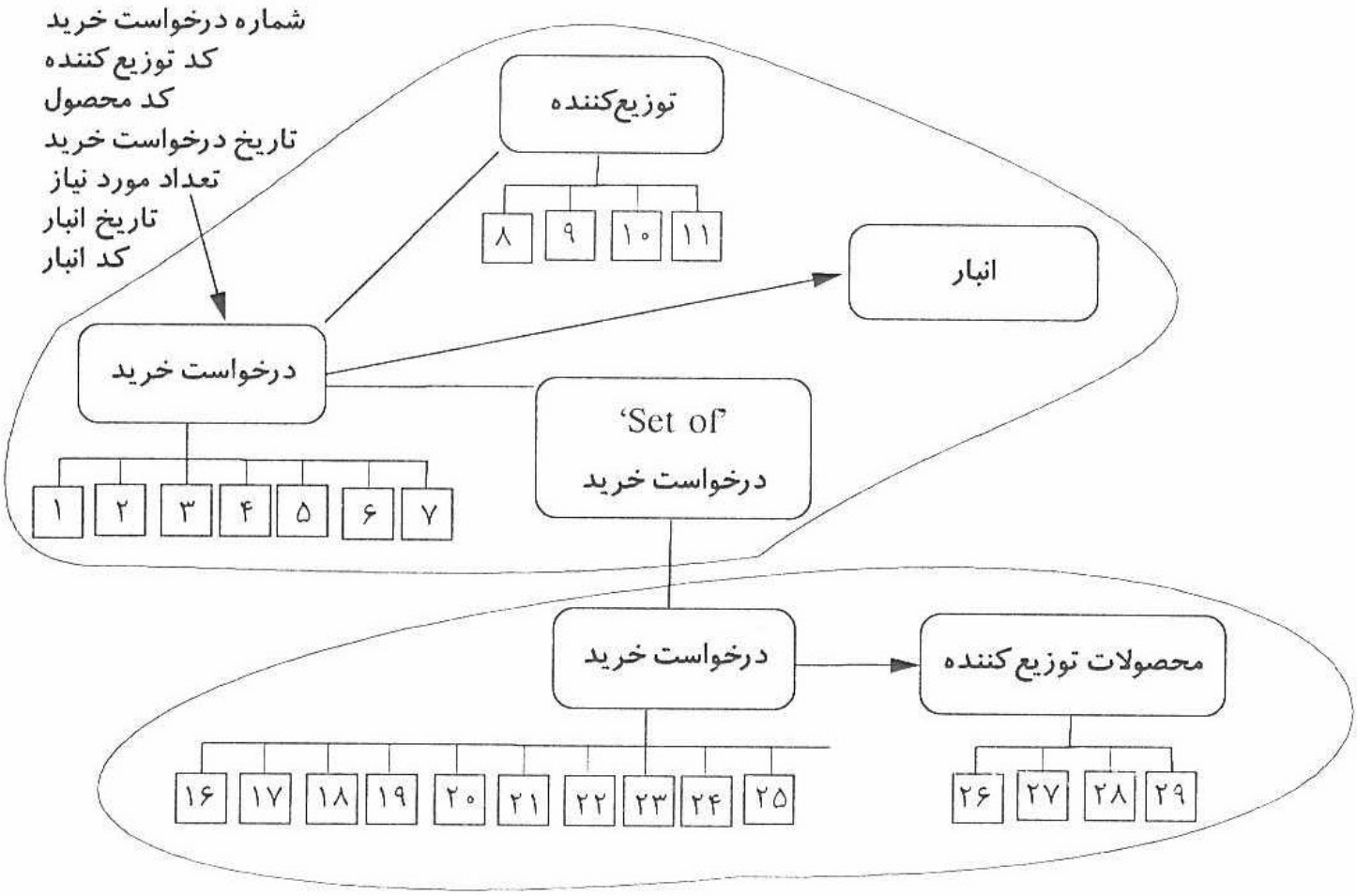
۱۷. «کد محصول» Set

۱۹. «تعداد مورد نیاز	Set
۲۰. «تاریخ نیاز»	Set
۲۱. «کد توزیع‌کننده»	Set
۲۲. «اقلام درخواست خرید» به درخواست خرید	Tie
۲۳. «اقلام درخواست خرید» به محصولات توزیع‌کننده	Tie
۲۴. ۲= وضعیت اقلام درخواست	Set
۲۵. «اقلام درخواست» ثبت اقلام	Write
۲۶. محصولات توزیع‌کننده برای این محصول	Read
۲۷. ۱<> وضعیت محصول	Fail if SI
۲۸. ۱= وضعیت محصول	Set
۲۹. محصولات توزیع‌کننده	Write

۲. تمام مستطیل‌هایی راکه تأثیر یک به یک بر یکدیگر دارند، گروه‌بندی می‌کنیم.

بدین ترتیب که از اولین مستطیل شروع کرده و تمام مستطیل‌هایی را که با آن ارتباط یک به یک دارند، در یک گروه قرار می‌دهیم. در این جا اولین مستطیل، مستطیل درخواست خرید می‌باشد که با مستطیل‌های فروشنده و انبار و مجموعه اقلام درخواست خرید ارتباط یک به یک دارد، پس اینها در یک گروه قرار می‌گیرند و گروه دوم مستطیل‌های مربوط به اقلام درخواست خرید و محصولات توزیع‌کننده می‌باشد. هر گروه را با کشیدن خطوطی در پیرامون آن مشخص می‌کنیم.

یادآوری می‌شود که در این نمودار هر مستطیل نمایانگر یک موجودیت می‌باشد، مگر آنهایی که توسط کلمه مجموعه (Set) مشخص شده‌اند. برای توضیح بیشتر می‌توان به فصل ۱۴ مراجعه کرد.



شکل ۱۸.۳ گروه‌بندی نمودار تطابق اثر

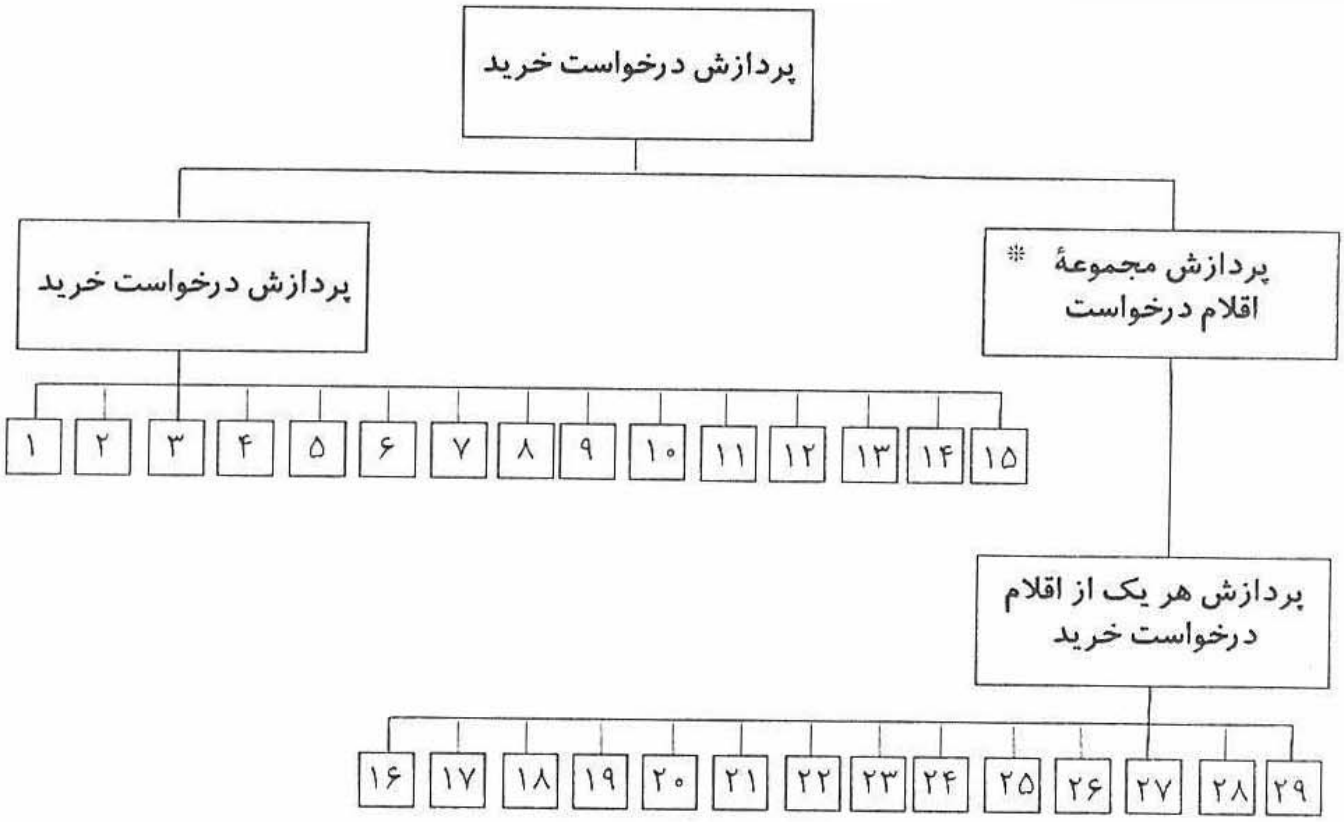
بدین ترتیب ماجول‌ها یا مراحل مستقل هر پردازش تعیین می‌گردد و برای هر گروه می‌توان نامی که نشان‌دهنده پردازش‌های آن گروه باشد، در نظر گرفت و هر مستطیلی که تنها بماند، خود به تنهایی در نظر گرفته می‌شود.

نمودار جکسون را ترسیم می‌کنیم.

در ابتدا یک مستطیل که حالت ریشه داشته و نام رویداد به اضافه کلمه پردازش دارد، ترسیم می‌گردد. گروهی که شامل نقطه ورودی کارکرد می‌باشد، ریشه این نمودار است و سپس سایر گروه‌ها ترسیم می‌گردد. ساختار به وجود آمده نشان‌دهندهٔ کلیه مراحل پردازش بروزرسانی می‌باشد. سایر گروه‌ها هر یک زیرگروه ماجولی با نام همان گروه می‌باشند.

اضافه‌کردن عملیات و شرط‌های پردازش

عملیات نمودار تطابق اثر رویدادها و شروط انجام این عملیات به مدل بروزرسانی اضافه می‌گردد. اگر اثر رویداد به روی چندین موجودیت با یکدگر تلفیق گردد، تمامی این بروزرسانی‌ها با یکدیگر و به ترتیب انجام می‌شوند.



شکل ۱۸.۴ مدل به روزرسانی رویداد

۱. «درخواست» (ایجاد درخواست	Create
۲. «شماره درخواست» (تعیین شماره درخواست)	Set
۳. «تاریخ درخواست»	Set
۴. «درخواست خرید» به انبار	Tie
۵. «درخواست خرید» به توزیع‌کننده	Tie
۶. ۲= وضعیت درخواست	Set
۷. «درخواست خرید»	Write
۸. «توزیع‌کننده»	Read
۹. ۲<> وضعیت توزیع‌کننده	Fail if
۱۰. ۲= وضعیت توزیع‌کننده	Set
۱۱. وضعیت توزیع‌کننده	Write
۱۲. «انبار درخواست خرید»	Read
۱۳. ۱<> وضعیت انبار	Fail if
۱۴. ۱= «وضعیت انبار»	Set

۱۵. «وضعیت انبار»	Write
۱۶. اقلام درخواست	Create
۱۷. «کد محصول»	Set
۱۸. «شماره درخواست»	Set
۱۹. «تعداد مورد نیاز	Set
۲۰. «تاریخ نیاز»	Set
۲۱. «کد توزیع‌کننده»	Set
۲۲. «اقلام درخواست خرید» به درخواست خرید	Tie
۲۳. «اقلام درخواست خرید» به محصولات توزیع‌کننده	Tie
۲۴. ۲= وضعیت اقلام درخواست	Set
۲۵. «اقلام درخواست»	Write
۲۶. محصولات توزیع‌کننده	Read
۲۷. ۱<> وضعیت محصول	Fail if
۲۸. ۱= وضعیت محصول	Set
۲۹. محصولات توزیع‌کننده	Write

بررسی مدل بروزرسانی کامل شده

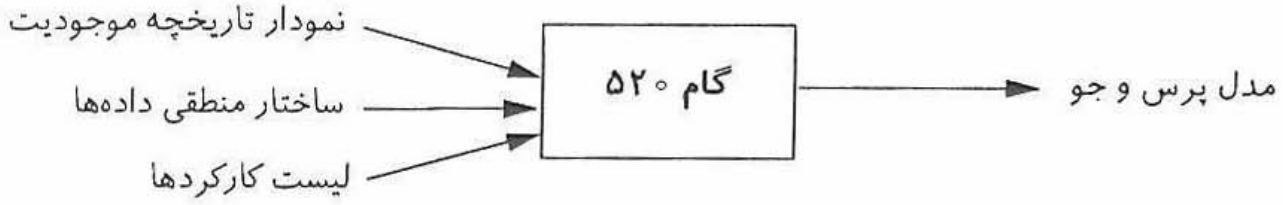
سپس مدل با کاربر در میان گذاشته تا مراحلی که احیاناً فراموش شده در نظر گرفته شود.

گام ۵۲۰ تعریف پردازش‌های (کارکرد) پرس‌وجو (ساخت مدل پرس‌وجو)

در این مرحله نمودارهایی را تهیه می‌کنیم که روش رسیدن به جواب در هر پرس‌وجو را مشخص می‌کند. یعنی می‌توان براساس نمودار به این که پرس‌وجو چه کارهایی انجام می‌دهد و اطلاعات را از چه موجودیت‌هایی بیرون می‌کشد پی برد. در این نمودارها روش کار هر پرس‌وجو مشخص می‌شود این نمودارها کاملاً شبیه نمودارهای تهیه شده در گام قبلی یعنی مدل بروزرسانی می‌باشد و تفاوت تنها در این جاست که نقطه شروع

برای تهیه این نمودارها براساس نمودار مسیر دستیابی می‌باشد و نه نمودار تطابق اثر.

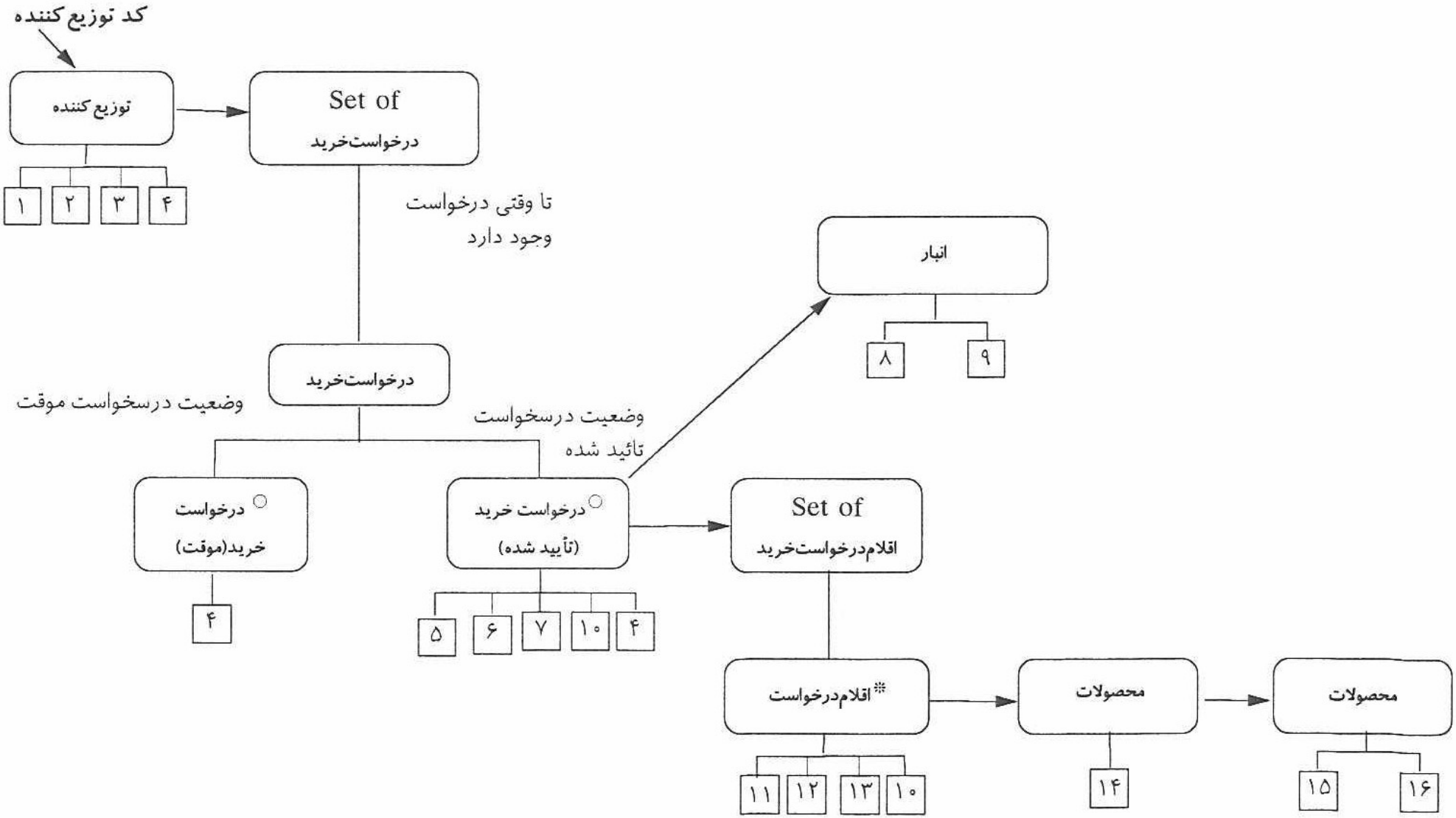
همان‌طور که در مورد مدل بروزرسانی بیان گردید با استفاده از علائم جکسون مدل پرس‌وجو را می‌سازیم تا بتوانیم ترتیب و توالی مراحل یک پردازش پرس‌وجو را نشان دهیم. با رسم این قبیل نمودارها علاوه بر کامل کردن مراحل پردازش می‌توان مراحلی را مشخص کرد که در آن مراحل خطا می‌تواند صورت گیرد.



شکل ۱۸.۵ ورودی‌ها و خروجی‌ها گام ۵۲۰

مراحل تهیه نمودار پرس‌وجو به ترتیب زیر می‌باشد:

۱. نمودار مسیر دستیابی را برای پردازش پرس‌وجو در نظر می‌گیریم و براساس آن مدل را ترسیم می‌کنیم. برای مثال پرس‌وجو مربوط به درخواست‌های خرید یک توزیع‌کننده را که در فصل ۱۴ چگونگی رسم نمودار مسیر دستیابی آن ذکر شده را در نظر می‌گیریم. نمودار مسیر دستیابی برای این پرس‌وجو به صورت زیر است:



شکل ۱۸.۶ نمودار مسیر دستیابی کارکرد «تعیین درخواست‌های خرید یک توزیع‌کننده»

عملیات این پردازش عبارت از:

۱. کد توزیع‌کننده — get

(دریافت کد توزیع‌کننده از ورودی)

۲. نام توزیع‌کننده — Read

(خواندن نام توزیع‌کننده از جدول مربوط)

۳. نام توزیع‌کننده — Output

(نمایش نام توزیع‌کننده)

۴. درخواست خرید — Read Next

(خواندن اطلاعات اولین درخواست از جدول مربوطه)

۵. تاریخ درخواست — Output

(نمایش تاریخ درخواست)

۶. ساعت درخواست — Output

(نمایش ساعت درخواست)

۷. انبار درخواست — Read

(خواندن اطلاعات انبار مربوطه)

۸. کد انبار — Output

(نمایش کد انبار درخواست)

۹. نام انبار — Output

(نمایش نام انبار درخواست)

۱۰. اقلام درخواست — Read Next

(خواندن اطلاعات اولین قلم درخواست)

۱۱. تعداد درخواستی — Output

(نمایش تعداد درخواستی)

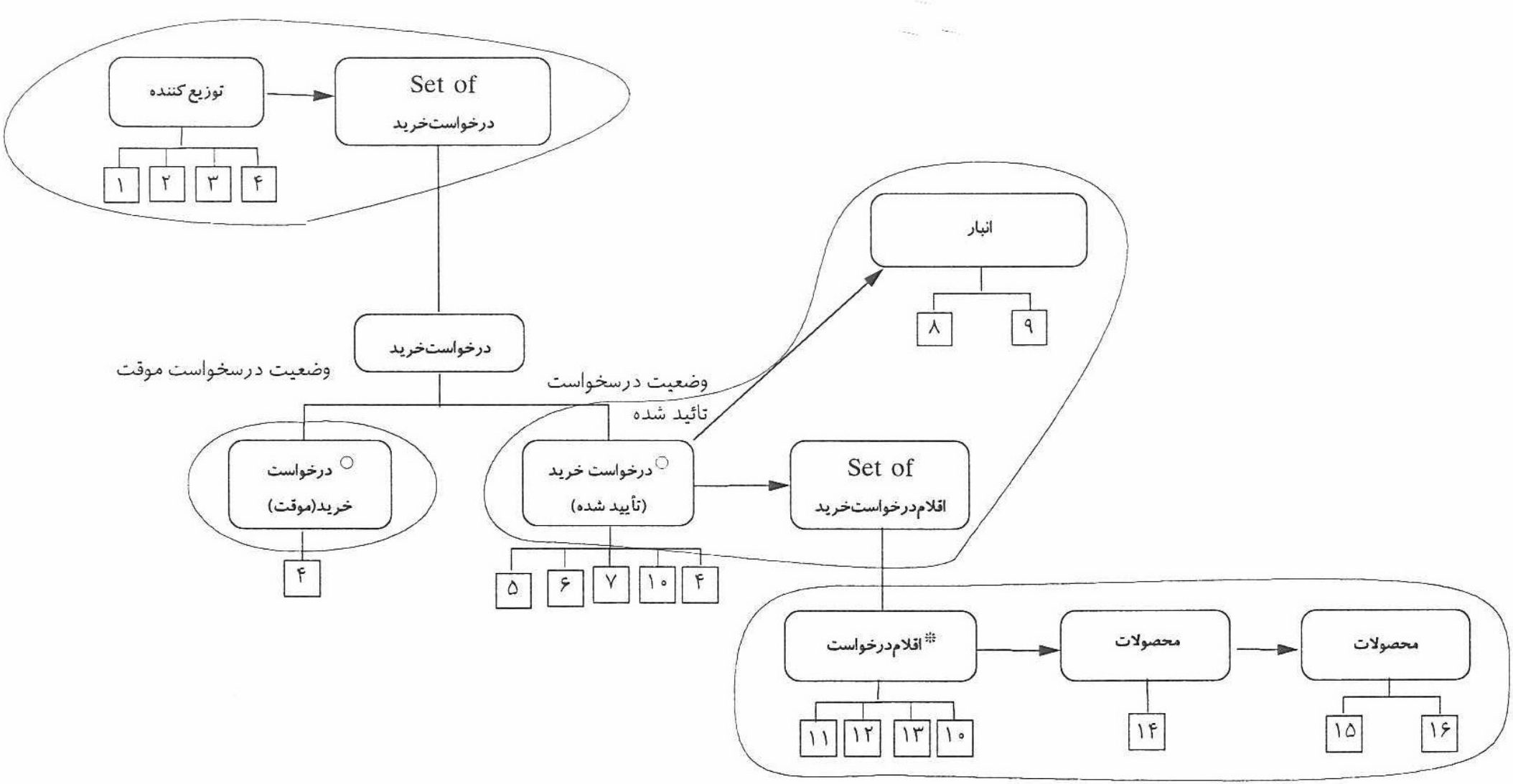
۱۲. تعداد تأیید شده — Output

(نمایش تعداد موافقت شده)

۱۳. محصولات توزیع‌کننده — Read

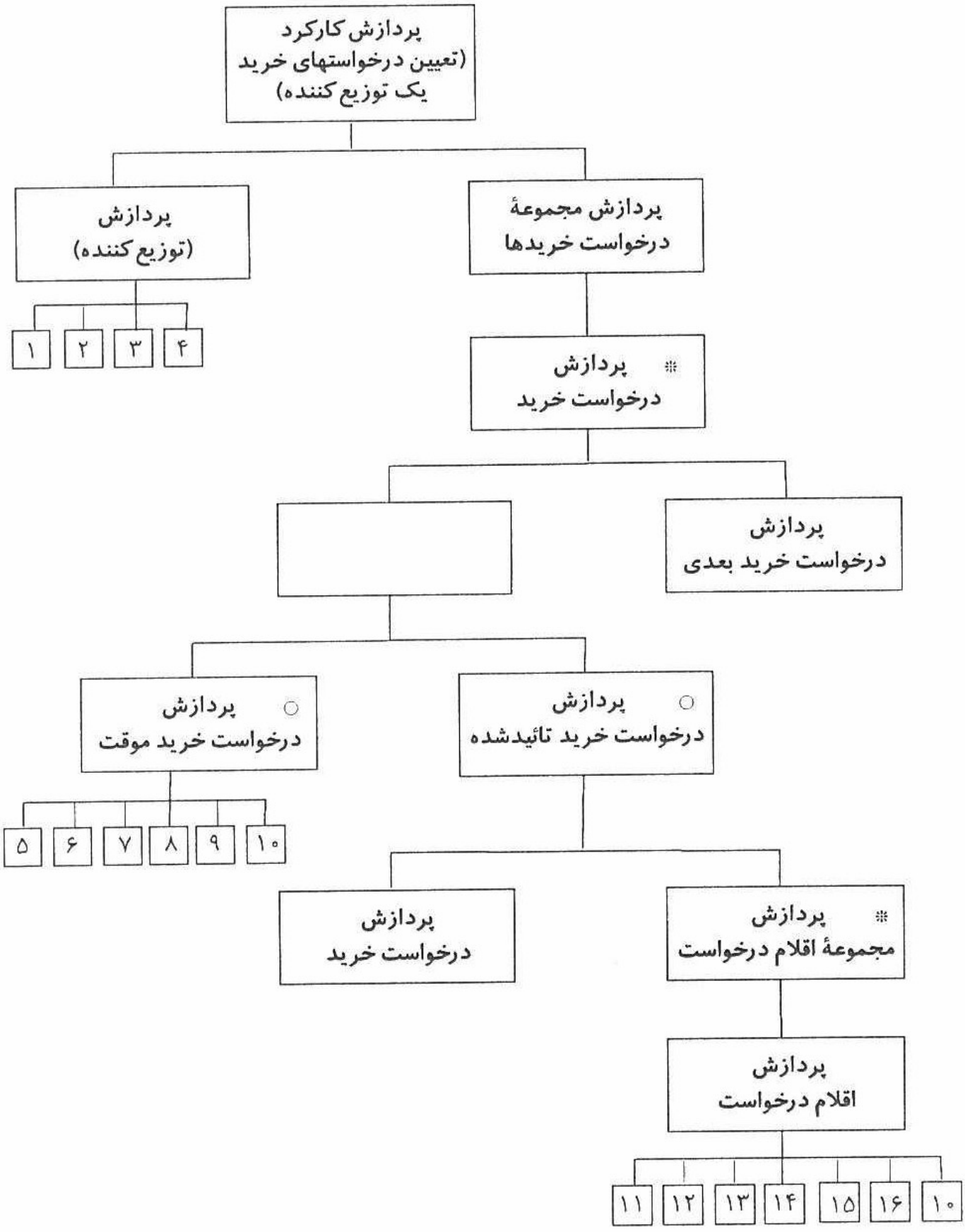
(خواندن اطلاعات محصولات توزیع‌کننده)

۱۴. محصولات — Read



شکل ۱۸.۷ گروه‌بندی نمودار مسیر دستیابی

۳. سپس نمودار را تبدیل به نمودار جکسون می‌کنیم. بدین ترتیب که برای هر گروه مستطیلی رسم می‌گردد.
۴. سپس شرط‌ها و عملیات هر پردازش به نمودار اضافه می‌شود و در مورد عملیات خواندن اطلاعات باید دقت شود تا در جای مناسب قرار گیرد یعنی یکبار قبل از شروع عملگر تکرار و یکبار در پایان هر تکرار وجود داشته باشد تا در صورت پیدا نشدن اطلاعات مناسب (رکورد مناسب) عملگر تکرار پایان یابد.



شکل ۱۸.۹ مدل پرس‌وجو

عملیات پردازش (کارکرد)

۱. کد توزیع‌کننده get

(دریافت کد توزیع‌کننده از ورودی)

۲. توزیع‌کننده Read

(خواندن نام توزیع‌کننده از جدول مربوط)

۳. نام توزیع‌کننده Output

(نمایش نام توزیع‌کننده)

۴. درخواست خرید Read Next

(خواندن اطلاعات اولین درخواست از جدول مربوطه)

۵. تاریخ درخواست Output

(نمایش تاریخ درخواست)

۶.ساعت درخواست Output

(نمایش ساعت درخواست)

۷. انبار درخواست Read

(خواندن اطلاعات انبار مربوطه)

۸. کد انبار Output

(نمایش کد انبار درخواست)

۹. نام انبار Output

(نمایش نام انبار درخواست)

۱۰. اقلام درخواست Read Next

(خواندن اطلاعات اولین قلم درخواست)

۱۱. تعداد درخواستی Output

(نمایش تعداد درخواستی)

۱۲. تعداد تأیید شده Output

(نمایش تعداد موافقت شده)

۱۳. محصولات توزیع‌کننده Read

(خواندن اطلاعات محصولات توزیع‌کننده)

۱۴. محصولات Read

(خواندن اطلاعات محصولات)

۱۵. کد محصول Output

Output

۱۶. نام محصول

(نمایش نام محصول)

۵. پس از ترسیم مدل آن را با کاربر در میان گذاشته تا مراحلی که احیاناً فراموش شده در نظر گرفته شود.

اهداف فصل

مرحله طراحی فیزیکی آخرین مرحله از مراحل SSADM می‌باشد. تمام مراحل تجزیه و تحلیل و طراحی به این مرحله که در آن مشخصات بانک اطلاعاتی مورد نیاز و برنامه‌ها ایجاد می‌گردد، ختم می‌شود. مستندات این مرحله با توجه به محیط پیاده‌سازی به گونه‌ای طراحی می‌شود که بتواند جوابگوی تیم برنامه‌نویسی باشد.

در طراحی فیزیکی دو اقدام اساسی انجام می‌شود:

اول آن که یک بانک اطلاعاتی براساس مدل منطقی داده‌ها ایجاد می‌شود که شامل کلیه موجودیت‌ها و ارتباطات بین آنها و اقلام اطلاعاتی مورد نظر باشد و دیگر آن که می‌خواهیم براساس لیست کارکردها که شامل ورودی‌ها و خروجی‌های هر کارکرد و ریزپردازش آن می‌باشد، مشخصات برنامه‌ها را تهیه کنیم.

سایر مستنداتی که در این مرحله و تهیه مستندات آن نقش کلیدی می‌تواند داشته باشد عبارت است از:

- فهرست کاربران
- لیست خطاها
- مشخصات فنی نرم‌افزارها و سخت‌افزارها
- استانداردهای سازمان مربوطه جهت ایجاد نرم‌افزارهای کاربردی

مقصود از استانداردهای سازمان عبارت است از قواعدی که سازمان مربوطه جهت ایجاد بانک اطلاعاتی و برنامه‌های مربوط به آن در نظر می‌گیرد که اگر چنین قواعدی وجود نداشته باشد بایستی ایجاد گردد.

این قواعد می‌تواند شامل استانداردهای نامگذاری و فرمت تهیه مستندات باشد.

به طور کلی می‌توان طراحی فیزیکی را شامل گام‌های زیر دانست:

گام ۶۱۰ ایجاد آمادگی لازم برای ایجاد بانک اطلاعاتی سیستم

گام ۶۲۰ طراحی بانک اطلاعاتی

گام ۶۳۰ تعریف گفتگوهای سیستم با کاربران و چگونگی این گفتگوها

گام ۶۴۰ تهیه لیست برنامه‌های هر کارکرد

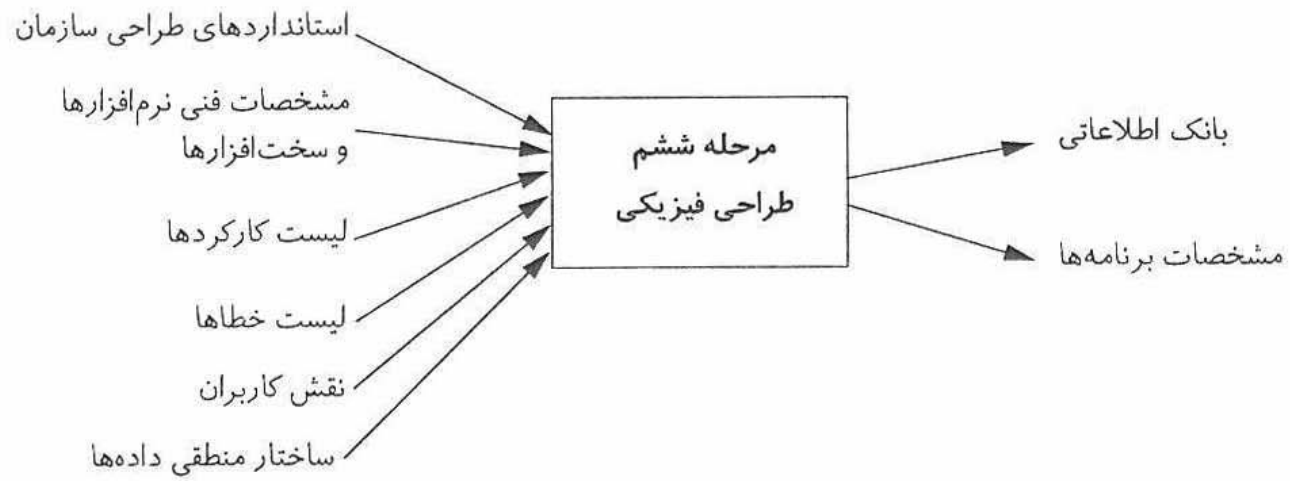
گام ۶۵۰ تهیه مشخصات برنامه‌ها

گام ۶۶۰ بهینه‌سازی طراحی بانک اطلاعاتی

گام ۶۷۰ ایجاد واسط بین داده‌ها و پردازش‌ها

گام آخر جمع‌آوری نتایج

گام‌های ۶۳۰ و ۶۴۰ می‌توانند همزمان با هم انجام شوند.



شکل ۱۹.۱ ورودی‌ها و خروجی‌های این مرحله

این مرحله براساس نیازهای نرم‌افزار مدیریت بانک اطلاعاتی انجام می‌شود. بنابراین کاملاً به نرم‌افزار مربوطه وابسته است. راه‌کارهایی که SSADM در این مرحله ارائه می‌دهد، شامل توصیه‌هایی جهت ایجاد بانک اطلاعاتی و سایر نکاتی است که نباید فراموش گردد و البته حالت عام دارد ـ آنچه می‌خواهد بگوید آن است که چگونه براساس نیازهای خودمان بانک اطلاعاتی مورد نظر را پیاده‌سازی کنیم.

بعضی از تولیدکنندگان نرم‌افزار، خود مستنداتی را در این زمینه ایجاد کرده‌اند که در این مرحله می‌توان از آن استفاده کرد. البته اگر از مدیریت بانک اطلاعاتی رابطه‌ای استفاده کنیم به دلیل آن که چنین نرم‌افزارهایی مشابهت زیادی به یکدیگر دارند. آنچه را در مورد یکی از آنها کاربرد دارد در مورد دیگران نیز می‌توان استفاده کرد.

گام ۶۱۰ ایجاد آمادگی جهت طراحی بانک اطلاعاتی

در این گام ابتدا سعی می‌شود که مستندات و استانداردهایی که پروژه‌های قبلی برای خود ایجاد کرده‌اند شناسایی گردد. چنین مستنداتی معمولاً مستندات توسعه برنامه‌های کاربردی نامیده می‌شود و آنچه برای ساخت و توسعه یک نرم‌افزار کاربردی در نظر گرفته شده می‌تواند برای ایجاد نرم‌افزار دیگر نیز مورد استفاده قرار گیرد. وجود چنین مستنداتی بسیار راه‌گشا می‌باشد. ولی آنها می‌بایستی به روز باشند چه اگر به روز نباشند از خصوصیات جدید مدیریت بانک اطلاعاتی استفاده نمی‌کنند به هر حال اگر چنین

مستنداتی در اختیار نباشد ما نسبت به ایجاد آن باید اقدام کنیم و در تهیه آن نکات زیر را در نظر بگیریم:

۱. چطور می‌توان ساختار منطقی داده‌ها را به طراحی فیزیکی تبدیل کرد و یا به عبارتی ارتباط بین موجودیت‌ها و جداول چگونه می‌باشد و روابط بین آنها چطور پیاده‌سازی گردد؟ چه پردازش‌هایی درون بانک اطلاعاتی انجام شود و چه قسمت‌هایی از پردازش به عهده برنامه‌ها گذاشته شود؟ استانداردهای نامگذاری در بانک اطلاعاتی چگونه باشد و اطلاعاتی از این قبیل که برای گام ۶۲۰ مورد نیاز است.

۲. مشخصات برنامه‌ها چگونه و چطور تهیه می‌شود؟ آیا از فرمت خاصی برای تهیه آنها استفاده گردد؟ آیا فلوچارت برنامه رسم گردد و یا به سادگی کلیه پردازش‌ها شرح داده شود.

زبان‌های محیط توسعه چیست؟ آیا از زبان‌های رویه‌ای (زبان‌های نسل سوم) یا غیر رویه‌ای (SQL) استفاده می‌گردد؟

۳. پارامترهای مؤثر در سرعت و کارآیی سیستم چیست؟ چه نکاتی در طراحی بانک اطلاعاتی باعث تسریع در دسترسی به اطلاعات می‌گردد. چطور backup بگیریم تا پردازش‌های روزانه دچار کندی نگردند.

۴. در این مرحله (طراحی فیزیکی) چه کارهایی انجام دهیم و فعالیت‌های این مرحله با توجه به محیط نرم‌افزاری و سخت‌افزاری چه باشند؟

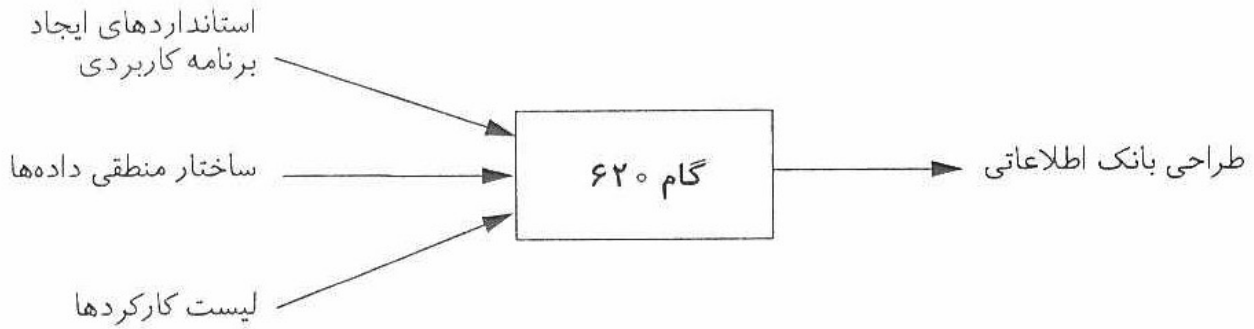
جا دارد متذکر شویم که مطالب عنوان شده در این مرحله کلی می‌باشد و برای کسب اطلاعات بیشتر و پی بردن به جزئیات، می‌توان به مستندات مدیریت بانک اطلاعاتی مورد نظر رجوع کرد.

گام ۶۲۰ طراحی اولیه بانک اطلاعاتی

در این گام براساس نمودار منطقی داده‌ها بانک اطلاعاتی مورد نظر را طراحی می‌کنیم و استانداردهای ما در این راستا، استانداردهای ایجاد برنامه کاربردی سازمان مربوطه می‌باشد.

اقداماتی که در این گام صورت می‌گیرد وابسته به مدیریت بانک اطلاعاتی می‌باشد و برحسب این که چه نرم‌افزاری را به کار ببریم این اقدامات متفاوت است. اما اگر از

مدیریت بانک اطلاعاتی رابطه‌ای استفاده کنیم این تفاوت بسیار اندک می‌باشد به ویژه آن که امروزه این نرم‌افزارها تفاوت چندانی با یکدیگر ندارند.

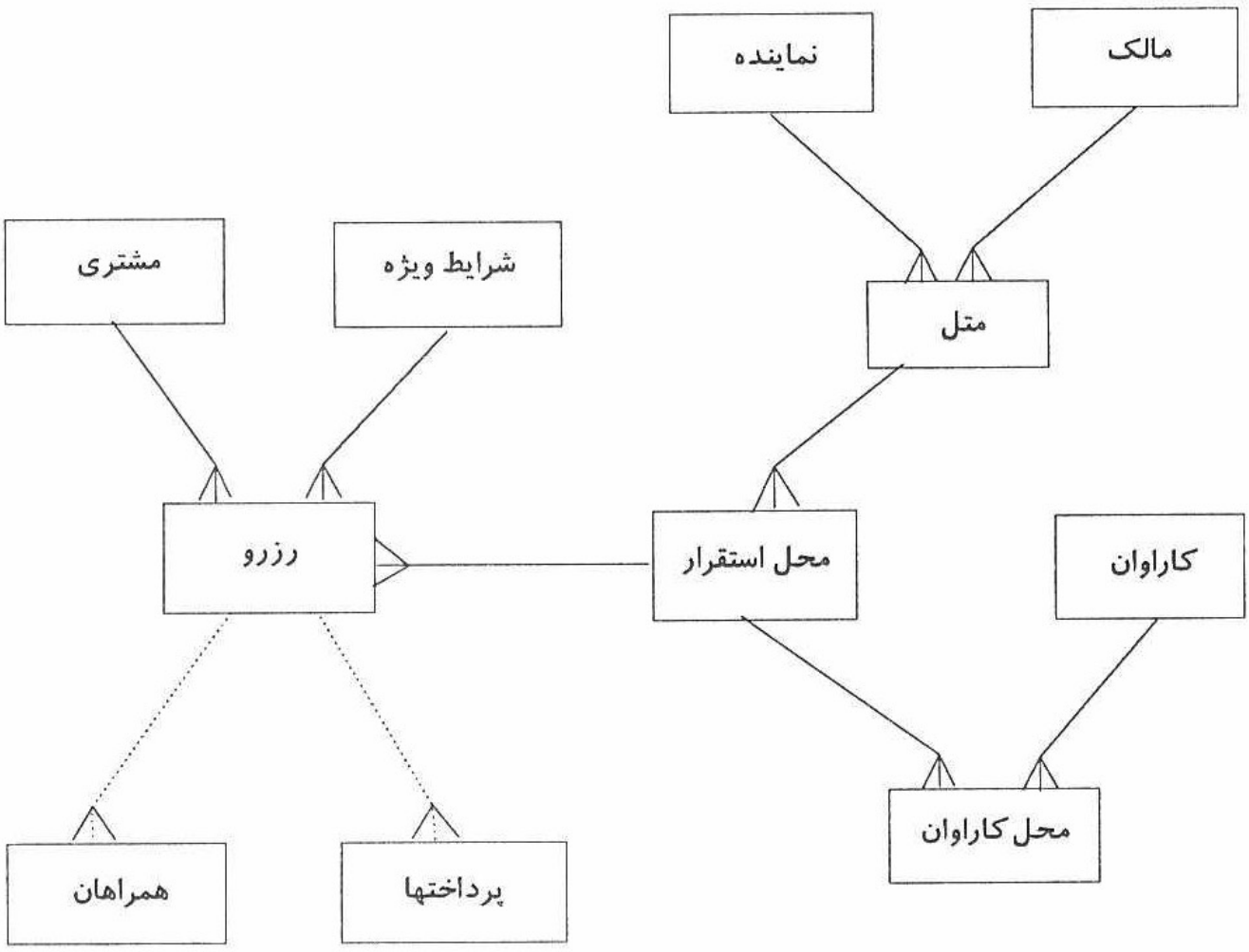


شکل ۱۹.۲ نمودار ورودی‌ها و خروجی‌ها این گام

برای ساده‌تر کردن مطلب مثال سیستم رزرواسیون متل را در نظر می‌گیریم و مطلب را براساس این مثال بیان می‌کنیم.

در شکل ۱۹.۳ نمودار منطقی داده‌ها برای سیستم مزبور نشان داده شده است.

اقدامات لازم جهت ایجاد بانک اطلاعاتی به شرح زیر می‌باشد:



شکل ۱۹.۳ نمودار ساختار منطقی داده‌ها سیستم رزرواسیون متل

۱. هر موجودیت به یک جدول تبدیل می‌گردد.
۲. هر جدول زیر دارای کلید اولیه به یکی از شکلهای زیر می‌باشد.

ـ کلید ساده

یکی از اقلام اطلاعاتی جدول به عنوان کلید اولیه برگزیده شده به قسمی که سطرهای جدول را یکتا می‌سازد.

ـ کلید ترکیبی

اگر کلید موجودیت ترکیبی است یعنی براساس ترکیب دو یا چندین قلم اطلاعاتی که یکی از آن‌ها اصلی و کلید موجودیت بالاسری است و دیگری به همراه اولی سطرهای جدول را یکتا می‌سازد قلم اطلاعاتی اصلی در ابتدا و دیگری بعد از آن می‌آید بنابراین اگر موجودیت همراهان را در نظر بگیریم قلم اطلاعاتی شماره رزرو چون وابسته به موجودیت بالاسری «رزرو» می‌باشد در ابتدا و سپس «شماره فرد همراه» که با قلم اطلاعاتی «رزرو» کلیه سطرهای جدول را یکتا می‌سازد در پی آن می‌آید.

ـ کلید مرکب

اگر موجودیت کلید مرکب داشته باشد یعنی کلید موجودیت از همراهی چندین قلم اطلاعاتی که هر یک وابسته به یک موجودیت بالاسری می‌باشد ایجاد می‌گردد در این صورت ترتیب قرار گرفتن اقلام اطلاعاتی در کنار هم مهم است اگر موجودیت محل کاراوان را در نظر بگیریم کلید آن از ترکیب اقلام زیر به دست می‌آید:

شماره کاراوان

شماره متل

کد محل استقرار

تاریخ استقرار

۳. اگر قرار باشد پرس‌وجوهایی براساس هر یک از اقلام اطلاعاتی که کلید اصلی را می‌سازند انجام دهیم می‌توانیم براساس هر یک از اقلام اطلاعاتی کلید اصلی کلیه ثانویه ایجاد کنیم و بدین ترتیب سیستم با در نظر گرفتن کلیدهای ثانویه سریع‌تر پاسخگو خواهد بود.
۴. تمام ارتباطات ساختار منطقی داده‌ها که در تعریف کلید اصلی در نظر گرفته نشده است مانند ارتباط موجودیت بالاسری با موجودیت وابسته در حالی که برای ساخت

کلید موجودیت وابسته از آن استفاده نشده باشد به صورت کلید خارجی تعریف می‌گردد.

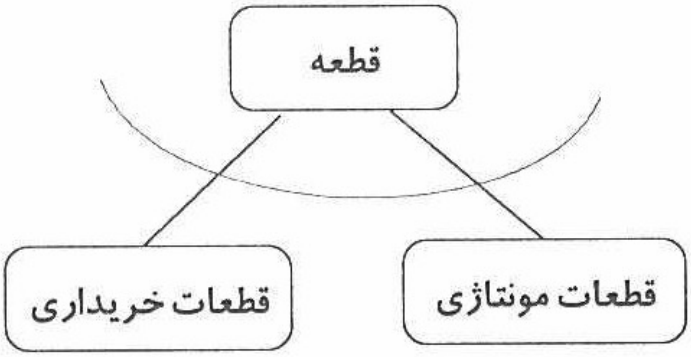
به عنوان مثال در موجودیت رزرو کلید موجودیت شماره رزرو است اما قلم اطلاعاتی شماره مشتری به موجودیت مشتری وابسته است و نمی‌تواند کد مشتری را نشان دهد که در آن موجودیت وجود نداشته باشد پس به صورت کلید خارجی تعریف می‌گردد.

۵. به لیست کارکردها رجوع کرده و مسیرهای دستیابی را در نظر می‌گیریم. هر گاه ورودی‌های مسیر دستیابی یک کارکرد کلید اصلی موجودیت نباشد مانند نام مشتری در مورد موجودیت مشتری در این صورت آن اقلام اطلاعاتی به صورت کلید ثانویه تعریف می‌گردند.

در بانک اطلاعاتی روابط موجودیت‌ها با استفاده از شاخص‌ها پیاده‌سازی می‌گردد و برای رسیدن به موجودیت بالاسری یک موجودیت وابسته از شاخص اولیه موجودیت بالاسری استفاده می‌گردد. و برای رسیدن به موجودیت وابسته از یک موجودیت بالاسری اگر کلید موجودیت وابسته ترکیبی باشد در کلید این موجودیت از کلید موجودیت بالاسری استفاده شده و اگر مرکب باشد با استفاده از شاخص‌های ثانویه می‌توان به آن دسترسی پیدا کرد. نکته دیگری که در طراحی بانک اطلاعاتی مورد توجه قرار می‌گیرد آن است که بعضی از موجودیت‌ها بهتر است در کنار یکدیگر واقع شوند (به شکل فیزیکی) اگر چنین موجودیت‌هایی شناسایی گردند، باعث تسریع در امر دسترسی به اطلاعات در بانک اطلاعاتی می‌گردد زیرا در دسترسی به یک صفحه (Page) از حافظه به تمام موجودیت‌های مرتبط با یکدیگر بازیابی می‌شوند. بعضی از نرم‌افزارهای مدیریت بانک اطلاعاتی امکان تقسیم‌بندی یک جدول به قسمت‌های مختلف را می‌دهد و امکان دسترسی موازی به یک جدول داده می‌شود که باعث کارآیی بانک اطلاعاتی در هنگامی است که پرس‌وجو و بروزرسانی جدول توسط کاربران زیاد می‌باشد پس بهتر است که موجودیت‌هایی که کاربران زیادی به صورت همزمان به آنها دسترسی دارند و تعداد رکوردهای آن زیاد است به صورت چنین جداولی در بانک اطلاعاتی نگهداری شوند.

آخرین نکته‌ای که در این زمینه می‌توان گفت: طرز نمایش زیرموجودیت‌ها در بانک اطلاعاتی می‌باشد. فرض کنید موجودیت قطعه دارای دو زیرموجودیت قطعات

خریداری و قطعات مونتاژی می‌باشد می‌توان سه وضعیت را برای نشان دادن این موجودیت و دو زیرموجودیت آن در نظر گرفت.

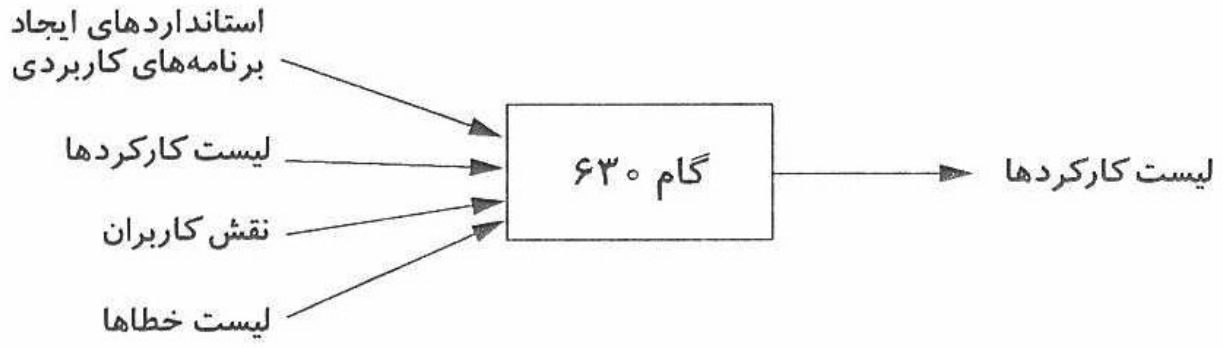


شکل ۱۹.۴ موجودیت قطعه به انضمام دو زیرموجودیت آن

۱. یک جدول که شامل تمامی اقلام اطلاعاتی هر سه موجودیت باشد، در نظر گرفته شود و به ازای هر قطعه اقلام اطلاعاتی مربوطه پر شود اما این وضعیت باعث ایجاد تعداد زیادی اقلام اطلاعاتی خالی خواهد شد که فضای زیادی از بانک اطلاعاتی را به خود اختصاص می‌دهد.
۲. دو جدول یکی برای قطعات خریداری و دیگری برای قطعات مونتاژی در نظر گرفته شود که این دو جدول بدین لحاظ که ارتباط خاصی با یکدیگر ندارند باعث ایجاد مشکلاتی در طراحی پرس‌وجوها خواهد شد و باعث ایجاد دوباره کاری نیز می‌گردد.
۳. برای موجودیت و زیرموجودیت‌ها هر کدام یک جدول ایجاد گردد که هر سه دارای یک کلید می‌باشند (شماره قطعه) که این راه مزایای نسبی هر دو روش بالا را دارد.

گام ۶۳۰ طراحی گفتگوهای سیستم

همان‌طور که قبلاً گفته شد SSADM این گام را در مرحله تعریف و مشخص‌سازی نیازها یعنی در مرحله سوم در نظر گرفته است اما این گام می‌تواند عملیات مربوط به این مرحله را بیشتر کند و در نتیجه زمان آن را طولانی‌تر سازد بنابراین بهتر است این گام در مرحله پنجم یا ششم صورت گیرد.



شکل ۱۹.۵ نمودار ورودی‌ها و خروجی‌های گام ۶۳۰

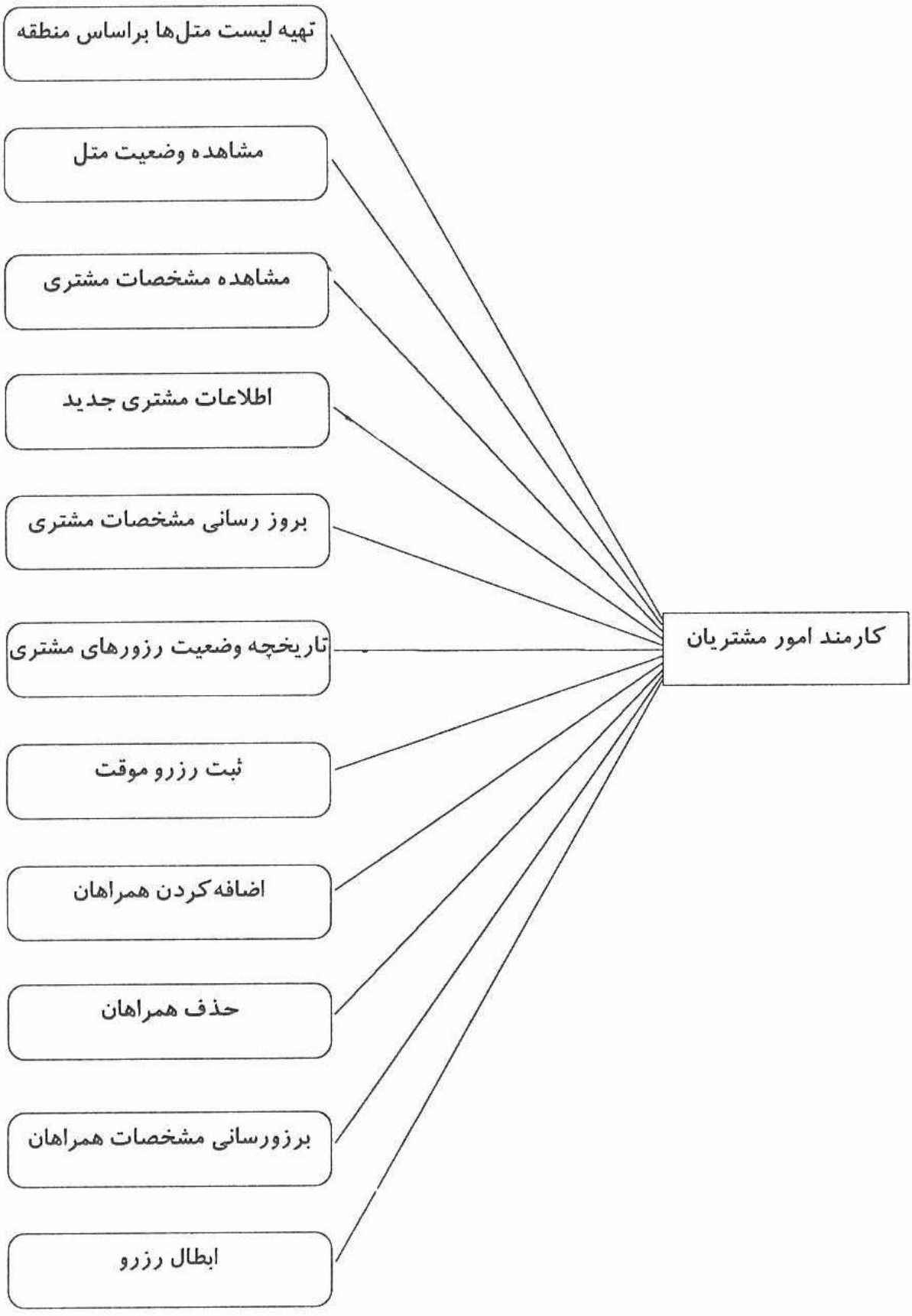
ساختار SSADM کاملاً سیستم مدار است یعنی آن که محاورات و گفتگوهای سیستم با کاربران پس از شناخت دقیق کارکردها صورت می‌گیرد و همان‌طور که گفته شد شناخت تمامی کارکردها پس از شناخت رویدادها صورت می‌گیرد پس این گام در زودترین مرحله پس از شناخت رویدادها می‌تواند انجام شود هر دیالوگ یا آنچه گفتگو نامیده می‌شود عبارت است از کارهایی که کاربر انجام می‌دهد و پیغام‌هایی که در جواب سیستم ارائه می‌دهد. هرگاه کاربر بخواهد یکی از کارکردها را فعال سازد با یک سری اقلام اطلاعاتی ورودی روبرو می‌شود پس در این گام می‌خواهیم نحوه ارتباط بین کاربر خاص و کارهایی که او می‌تواند انجام دهد و اقلام ورودی مربوطه را طراحی کنیم. در مرحله تجزیه و تحلیل، کاربران احتمالی سیستم شناخته می‌شوند و در مرحله شناخت نیازمندی‌ها، نقش هر کاربر و کارکردهایی که او می‌تواند انجام دهد شناسایی می‌گردد. این اطلاعات در یک ماتریس به نام نقش کاربر / کارکرد قرار می‌گیرد که این ماتریس می‌تواند برای طراحی منوها مورد استفاده باشد.

ساختار منوها

پس از آن که کاربر خود را به سیستم معرفی کرد، سیستم کارکردهای مخصوص هر کاربر را برای او نمایش می‌دهد و کاربر براساس نیاز خود یکی از آنها را انتخاب می‌کند. اگر سیستم رزرواسیون کاراوان‌ها را در نظر بگیریم کارمند امور مشتریان وظایف زیر را می‌تواند انجام دهد.

کارمند امور مشتریان	ـ تهیه لیست متل‌ها براساس منطقه
	ـ مشاهده وضعیت متل
	ـ مشاهده مشخصات مشتری
	ـ ثبت اطلاعات مشتری جدید
	ـ بروزرسانی مشخصات مشتری
	ـ ثبت رزرو موقت
	ـ اضافه کردن همراهان
	ـ حذف همراهان
	ـ بروزرسانی مشخصات همراهان
	ـ ابطال رزرو

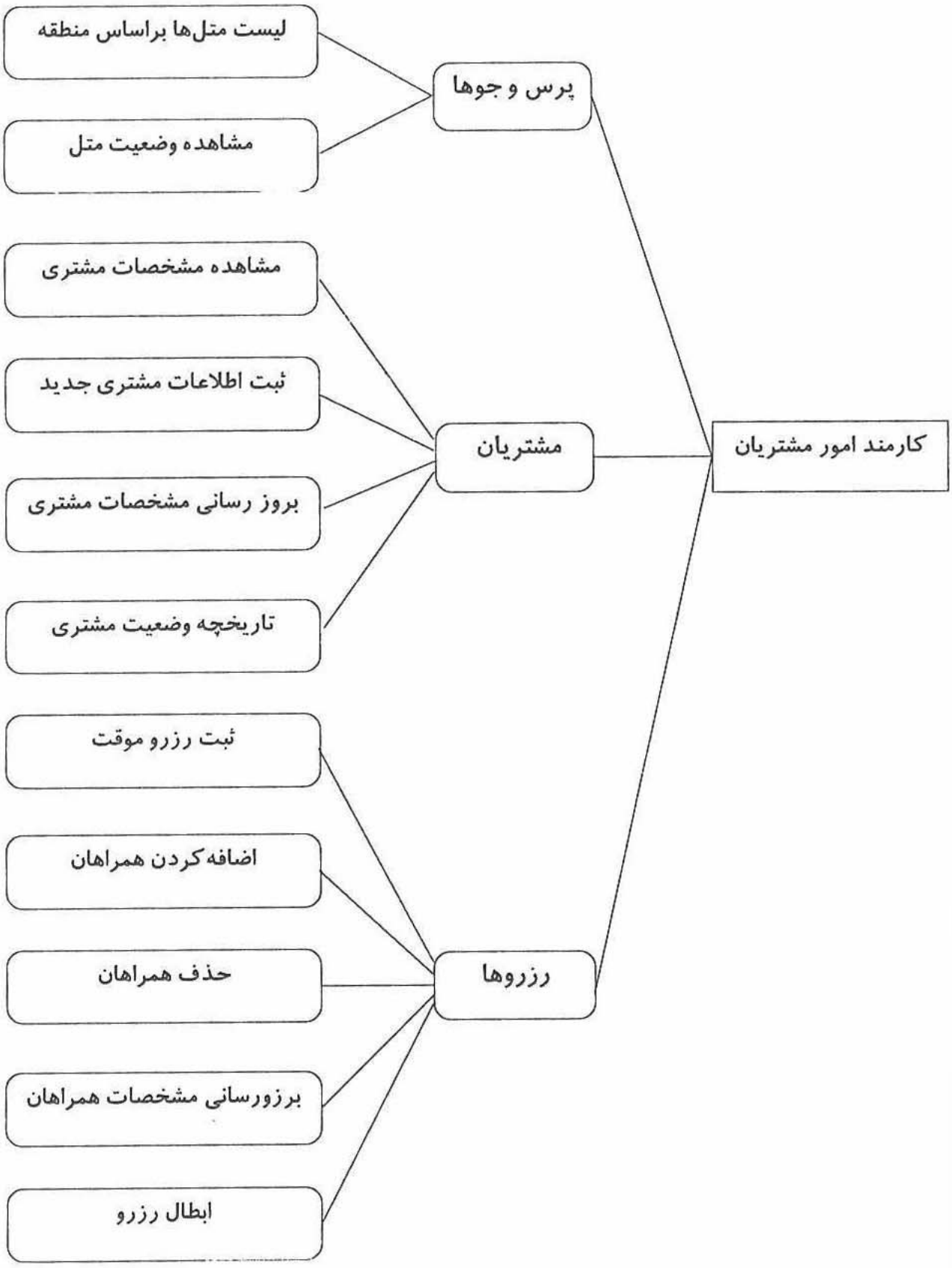
با استفاده از نمادهای SSADM منوی مربوطه جهت کارمند امور مشتریان، به صورت زیر نمایش داده می‌شود.



شکل ۱۹.۶ منوی سیستم برای کارمند امور مشتریان

برای نشان دادن منو از یک مستطیل استفاده می‌شود و اقلام مربوطه توسط مستطیل گردگوشه مشخص می‌گردد.

می‌توان برای پرهیز از شلوغ شدن زیاد منو آن را در قالب زیرمنو (Submenu) دسته‌بندی کرد و به صورت زیر نمایش داد:



شکل ۱۹.۷ منوی سیستم برای کارمند امور مشتریان

ساختار فرمان

پس از انتخاب هر یک از اقلام آخرین سطح منو که SSADM آن را دیالوگ می‌نامد اطلاعات مورد نظر بر روی صفحه نمایش ظاهر می‌گردد که این صفحه می‌تواند شامل دیالوگ‌های دیگری نیز باشد. دیالوگ‌ها می‌توانند به صورت منو یا آنچه در محیط Windows مرسوم است به صورت یک Button نمایش داده شوند. چگونگی نمایش دیالوگ‌ها بستگی به ابزارهای موجود و محیط سیستم دارد که برحسب امکانات می‌تواند به صورت‌های مختلفی نمایش داده شود اما در این جا می‌خواهیم فرمی به نام ساختار فرمان پیشنهاد کنیم. پس از انتخاب هر دیالوگ کارها و اطلاعاتی که می‌بایستی روی صفحه ظاهر گردد در این فرم ثبت و مستند می‌شود. فرض کنید که کاربر ما (کارمند امور مشتریان) می‌خواهد اطلاعات مشتری را به روزآوری کند آنچه او نیاز دارد جستجوی لیست مشتریان برای پیدا کردن شماره مشتری مورد نظر و سپس به روزآوری اطلاعات او می‌باشد سپس می‌تواند به منوی اصلی باز گردد یا همین کار را برای مشتری دیگری انجام دهد.

ساختار فرمان		
نام سیستم: سیستم رزرواسیون کاراوانها		
نقش کاربر: کارمند امور مشتریان		
نام دیالوگ: بروزآوری اطلاعات مشتریان		کد دیالوگ: Dialoog 005
انتخابها	**دیالوگ یا منو**	**نام دیالوگ یا منو**
لیست مشتریان	دیالوگ	جستجوی لیست مشتریان Dialoo6
ثبت اطلاعات	دیالوگ	ثبت اطلاعات بروزآوری شده Dailoo7
خروج	منو	برگشت به منوی اصلی

جدول ۱۹.۱

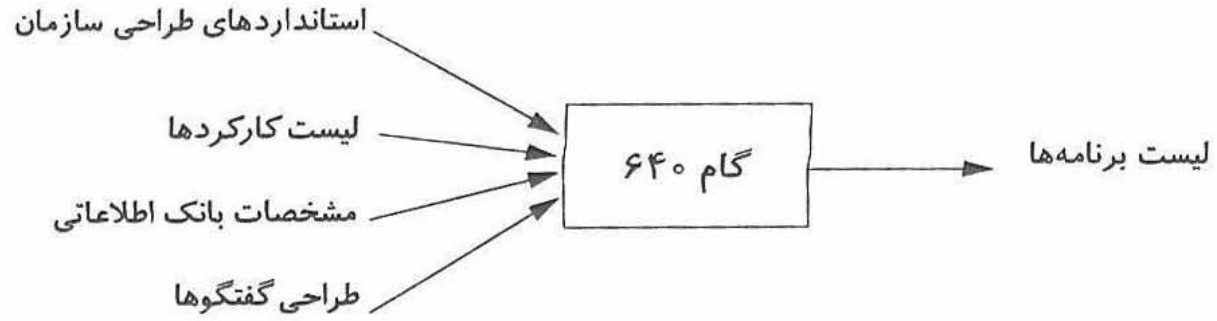
مدیریت خطا

آخرین کاری که در این گام انجام می‌شود تعیین نحوهٔ عملکرد سیستم در هنگام بروز خطا در کارکردها می‌باشد. هر کارکرد در هنگام بروز خطا پیغام‌های خطا را از لیست خطاها تهیه کرده و به کاربر نشان می‌دهد همین‌طور در هنگام اجرای عملیات طولانی که

تعداد زیادی رکورد را بروزآوری می‌کنیم پیغام مناسب، وضعیت سیستم را به کاربر نمایش می‌دهد طرز نمایش این‌گونه پیغام‌ها براساس طراحی فیزیکی و استانداردهای موجود در سازمان می‌باشد.

گام ۶۴۰ پیاده‌سازی کارکردها

هدف اصلی در این گام تجزیه هر کارکرد به تعدادی برنامه می‌باشد که این برنامه‌ها در مجموع می‌توانند وظایف کارکرد را انجام دهند. واژه برنامه از دیدگاه افراد متفاوت می‌تواند تعاریف متعدد و متفاوت داشته باشد و در بعضی از محیط‌های توسعه ممکن است آن را پردازش، ماجول روتین و غیره نامیده شود ولی آنچه در این جا مورد نظر است قسمتی از پردازش است که یکباره انجام می‌شود در گام ۳۳۰ «کارکرد» به صورت زیر تعریف شد: «کاری است که از نظر کاربر یکباره انجام می‌شود» اگر این دو تعریف را در کنار یکدیگر قرار دهیم می‌توان گفت که برای هر کارکرد یک برنامه در نظر گرفته می‌شود اما این مطلب در کلیه موارد صادق نیست و به همین دلیل باید روشی برای تجزیه کارکردها انتخاب شود.



شکل ۱۹.۸ ورودی‌ها و خروجی‌های گام ۶۴۰

می‌توان مانند جدول ۱۹.۲ یک ماتریس را در نظر گرفت که یک ضلع ماتریس نام کارکردها و ضلع دیگر آن نام برنامه‌ها می‌باشد و علامت x نشان می‌دهد که برای اجرای کارکرد چه برنامه‌هایی مورد نیاز می‌باشد. اگر تعداد کارکردها و برنامه‌ها زیاد باشد ایجاد چنین ماتریس‌هایی مشکل خواهد بود اما این ماتریس مفهوم کلی مورد نظر را نشان می‌دهد چنین ماتریسی نقشه پیاده‌سازی اجزا کارکرد

(Function Component Implementation Map)

نامیده می‌شود.

نام برنامه / نام کارکرد		Prog 1	Prog 2	Prog 3	Prog 4	Prog 5	Prog 6
f1	کارکرد	*					
f2	کارکرد		*	*			
f3	کارکرد			*	*		
f4	کارکرد				*	*	*

جدول ۱۹.۲

همان‌طور که در شکل نمایش داده می‌شود، کارکرد F۱ تنها توسط یک برنامه اجرا می‌گردد. کارکرد F۴ توسه سه برنامه Prog۴ و Prog۵ و Prog۶ اجرا می‌شود که می‌توان تصور کرد که Prog۴ اطلاعات را بازیابی می‌کند و Prog۵ اطلاعات را پردازش می‌نماید. و Prog۶ اطلاعات را در قالب یک گزارش عرضه می‌کند. کارکردهای F۲و F۳ نیز هر یک توسط چندین برنامه انجام می‌شوند که یک برنامه Prog۳ در بین آنها مشترک است اگر محیط توسعهٔ نرم‌افزار، محیط Client / Server باشد تعیین این که کدام برنامه در محیط کاربر و کدام برنامه در محیط Server انجام شود، بسیار مهم است. یک دستگاه عابر بانک را در نظر بگیرید، اگر مشتری از دستگاه درخواست ارسال صورت‌حساب را بنماید در این صورت احتمالاً سه برنامه برای این کارکرد در نظر گرفته می‌شود یکی در محیط خود دستگاه اجرا می‌شود که اطلاعات مشتری را برای کامپیوتر مرکزی می‌فرستد دومی در محیط کامپیوتر مرکزی اجرا شده و اطلاعات مشتری را درون جدولی ثبت می‌کند و سومی برنامه‌ای شبانه است که اطلاعات صورت‌حساب مشتری را از درون جداول سیستم استخراج و گزارش آن را تهیه می‌کند تا برای او ارسال گردد.

در تهیه این ماتریس نکات زیر را در نظر می‌گیریم:

۱. تا آنجا که ممکن است کارکردها را به صورتی تجزیه می‌کنیم که قطعات برنامه‌ها به صورت مجدد توسط سایر کارکردها استفاده گردد و برنامه حالت reuseable داشته باشد، راه خاصی برای شناسایی چنین برنامه‌هایی وجود ندارد و فقط باید به تجربه، تحلیل‌گر تکیه کرد.

۲. در مورد نحوه پیاده‌سازی برنامه‌ها تصمیم گرفته شود که آیا برنامه‌ها توسط

زبان‌های غیررویه‌ای (مانند SQL) پیاده‌سازی گردد و یا توسط زبان‌های رویه‌ای پیاده‌سازی شود.

۳. هر برنامه در ضمن اجرا ممکن است چندین جدول را به روزآوری کند اگر در ضمن اجرای برنامه به هر دلیل یکی از جداول نتواند به روز گردد در بانک اطلاعاتی مشکل پیش می‌آید. در چنین حالتی اطلاعات باید به صورت اول باز گردد بنابراین در تجزیه کارکردها به این نکته توجه می‌کنیم که چه برنامه‌هایی می‌بایستی همگی و به درستی اجرا گردند تا بتوانیم در صورت بروز خطا اطلاعات را به حالت اول برگردانیم.

به برنامه‌هایی که همگی باید با هم اجرا شوند یا با هم برگردانده شوند واحد موفقیت (Success Unit) می‌گویند تعریف SSADM از واحد موفقیت به صورت زیر است: «مجموعه‌ای از پردازش‌ها که همگی و به یکباره انجام می‌شوند و یا برگشت می‌خورند» در حالتی که پردازش‌ها برگشت داده می‌شود وضعیت سیستم به حالت قبل از شروع پردازش‌ها باز می‌گردد.

حالت‌هایی نیز می‌تواند وجود داشته باشد که برگشت اطلاعات برای کاربران مشکل‌ساز باشد خصوصاً در مورد برنامه‌های لحظه‌ای چنین مشکلاتی بیشتر نمایان می‌شود. فرض کنید که می‌خواهیم اطلاعات یک فرم را که دارای تعداد زیادی اقلام اطلاعاتی است وارد کامپیوتر نماییم. هرگاه به دلیل مشکلاتی نتوانیم اطلاعات آخرین اقلام را وارد کنیم تمام اطلاعات وارد شده توسط کاربر از بین می‌رود. بنابراین می‌توانیم مکانیزم‌هایی را در نظر بگیریم که در مورد چنین فرم‌هایی اجرا شوند (مثلاً ثبت موقت) تا پس از رفع مشکلات از بقیه اطلاعات وارد شده بتوان استفاده کرد.

گام ۶۵۰ تهیه مشخصات برنامه‌ها

در این گام مشخصات برنامه‌ها تهیه می‌شود. باید هدف ایجاد مستندات در حدی باشد که برنامه‌نویس بتواند آن را کد کند. هیچ استاندارد معین و مشخصی در SSADM در این زمینه وجود ندارد. اما مشخصات برنامه‌ها می‌تواند شامل عناوین زیر باشد:

۱. شرح برنامه‌ها
۲. شرح جزئیات اطلاعات ورودی
۳. شکل و فرمت گزارشات

۴. اطلاعات جداول و روابط آنها

۵. پیغام‌های خطا

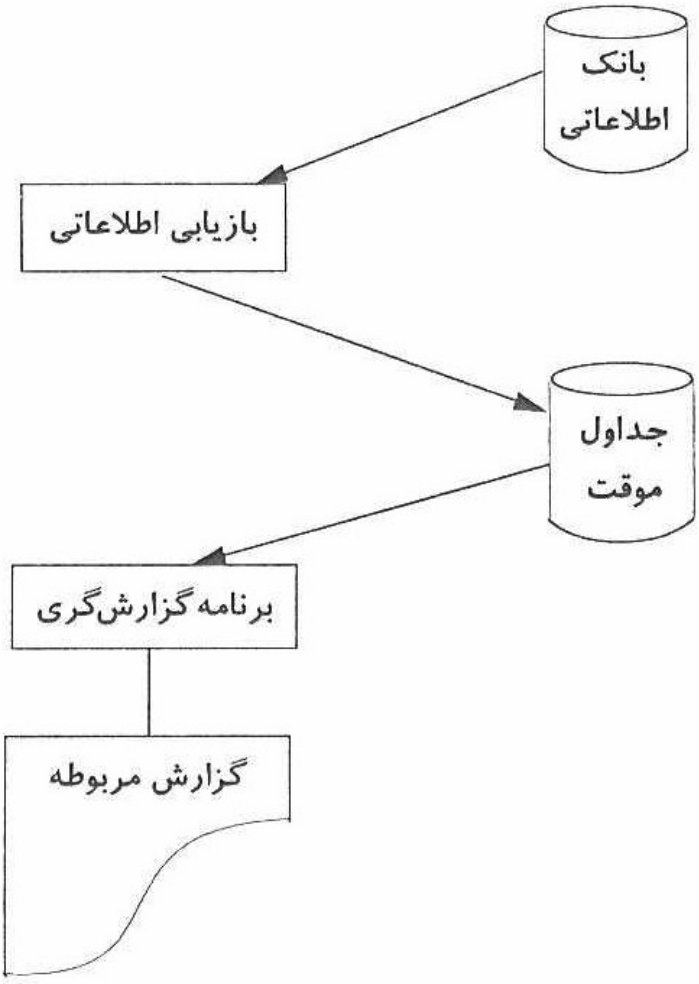
۶. نام و مشخصات زیر روال‌ها (روتین‌ها) که در برنامه از آنها استفاده می‌شود.

۷. تعیین واحد پیشروی

۸. چگونگی و نحوهٔ استفاده کاربر از اطلاعات کمکی

این اطلاعات براساس اطلاعات لیست کارکردها و ساختار منطقی داده‌ها و سایر مستندات تهیه شده ایجاد می‌گردد. جزئیات هر یک از موارد فوق تا حد زیادی بسته به محیط فیزیکی سیستم و زبان برنامه‌نویسی مورد نظر می‌تواند متفاوت باشد.

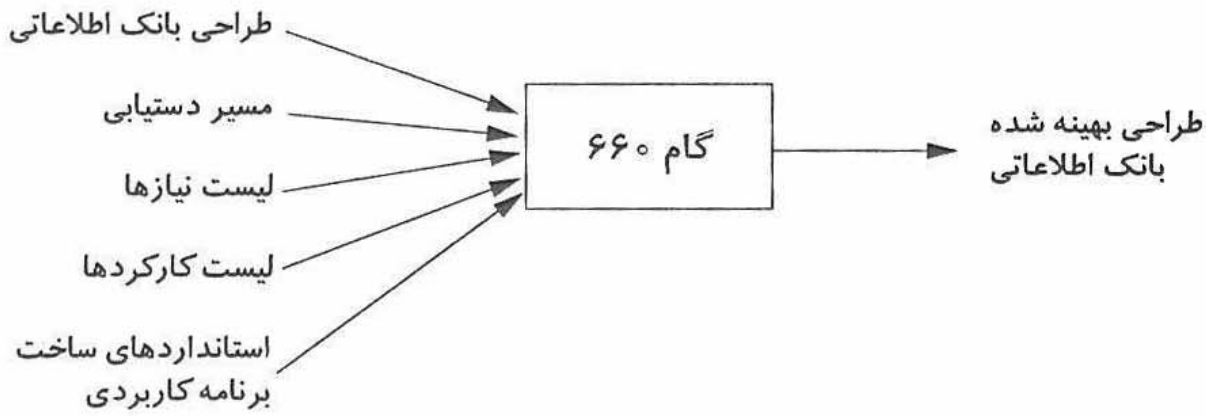
می‌توان با رسم نمودارهایی ارتباط بین برنامه‌ها و بانک اطلاعاتی را نشان داد. چنین نمودارهایی به برنامه‌نویس کمک می‌کند تا شکل و نوع ارتباط بین برنامه‌ها را بهتر درک کند. به طور مثال نمودار زیر نحوهٔ عملکرد برنامه‌های مختلف برای تهیه یک گزارش را نشان می‌دهد.



شکل ۱۹.۹

گام ۶۶۰ بهینه‌سازی طراحی بانک اطلاعاتی

در این گام، طراحی بانک اطلاعاتی مورد بازبینی قرار می‌گیرد تا مطابقت آن با لیست کارکردها و نیازها بررسی شود. در این گام بیشترین تأکید بر روی زمان پاسخ‌دهی سیستم می‌باشد و بانک اطلاعاتی از زاویه داشتن کارآیی لازم و پاسخ به کاربران در زمان تعیین شده مورد بررسی قرار می‌گیرد تا اگر زمان پاسخ‌دهی سیستم به کاربران و فضای مورد نیاز آن با لیست نیازها مطابقت نداشته باشد، ساختار بانک اطلاعاتی مورد تجدید نظر قرار گیرد.



شکل ۱۹.۱۰ ورودی‌ها و خروجی‌های گام ۶۶۰

در این گام اقدامات زیر صورت می‌گیرد:

ـ تخمین فضای مورد نیاز سیستم

فضای مورد نیاز سیستم براساس تعداد رکوردهای پیش‌بینی شده تخمین زده می‌شود و با فضایی که در لیست نیازها تعیین شده مقایسه می‌گردد، در صورت عدم تطابق آن، ساختار بانک اطلاعاتی به گونه‌ای تغییر می‌یابد که جوابگوی انتظارات کاربران باشد.

ـ تخمین زمان پاسخ‌دهی سیستم

زمان کارکردهای مهم تخمین زده شده و با زمان مورد نیاز کارکرد در لیست کارکردها مطابقت داده می‌شود در صورت عدم تطابق، تکنیک‌هایی را برای کم کردن زمان پاسخ‌دهی سیستم در نظر می‌گیریم. بعضی مواقع واسط‌های گرافیکی خصوصاً در هنگامی که می‌خواهند اقلام ورودی را با اطلاعات بانک مقایسه کنند باعث کندی کار برنامه‌ها می‌شوند در چنین مواردی اصلاحاتی در ساختار بانک صورت می‌گیرد تا زمان پاسخ‌دهی مورد نظر به دست آید.

اگر زمان پاسخ‌دهی کارکرد را هنگامی که کارکرد به صورت مجزا اجرا می‌گردد در نظر بگیریم مسلماً نتایج درستی به دست نمی‌آوریم.

فرض کنید می‌خواهیم زمان پاسخ‌دهی یک کارکرد لحظه‌ای را در نظر بگیریم که همزمان با یک کارکرد دسته‌ای (batch) اجرا می‌گردد. کارکرد دوم تعداد زیادی از رکوردها را قفل (lock) می‌کند. در این صورت زمان پاسخ‌دهی کارکرد لحظه‌ای متفاوت با زمان پاسخ‌دهی آن در حالت مجزا می‌باشد، پس در پیاده‌سازی کارکردها، چنین نکاتی باید در نظر گرفته شود و پیاده‌سازی براساس آن صورت گیرد. بدین ترتیب می‌توان تغییرات بانک اطلاعاتی را در زمان توسعه و تهیه برنامه به حداقل کاهش داد.

گام ۶۷۰ ایجاد واسط بین داده‌ها و پردازش‌ها

هر گاه بین ساختار منطقی داده‌ها و طراحی فیزیکی داده‌ها در بانک اطلاعاتی تفاوت اساسی وجود داشته باشد SSADM ساخت چنین واسطی را پیشنهاد می‌کند. لزوم ایجاد چنین واسطی هنگامی احساس می‌گردد که برای تهیه بانک اطلاعاتی از یک نرم‌افزار مدیریت بانک اطلاعاتی که اطلاعات را به صورت سلسله مراتبی ذخیره‌سازی می‌کند، استفاده کنیم و در این صورت مدیریت بانک اطلاعاتی داده ها را توسط شاخص‌های خاص خود نگهداری می‌کند و موجودیت‌ها به صورت جدول در نظر گرفته نمی‌شوند. ایجاد چنین واسطی کمک می‌کند که برنامه‌های سیستم داده‌های بانک اطلاعاتی را به صورت ساختار منطقی داده‌های سیستم ببینند و به طراحی فیزیکی داده‌ها و این که بانک اطلاعاتی برای دستیابی به آنها از چه روشی استفاده می‌کند و یا چگونه آنها را به روزآوری می‌کند کاری نداشته باشند در این صورت بازیابی و به روزرسانی داده‌ها به عهده این واسط گذاشته می‌شود. دلایل استفاده از چنین واسطی می‌تواند به شرح زیر باشد:

- به جای آن که تغییرات در سطح فیزیکی صورت گیرد، در سطح ساختار منطقی داده‌ها و با کمک گرفتن از این واسط اعمال گردد تا در نتیجه بسیار سریع‌تر انجام شود.
- این واسط اجازه می‌دهد که در صورت عوض شدن مدیریت بانک اطلاعاتی برنامه‌های سیستم بدون تغییر باقی بمانند و تغییراتی در این لایه نداشته باشیم اگر چه ایجاد این لایه امکانات بالا را به ارمغان می‌آورد، به دلیل زیر همیشه

احتیاجی به ایجاد آن نیست.

۱. اکثر نرم‌افزارهای مدیریت بانک اطلاعاتی امروزه به صورت رابطه‌ای طراحی می‌شوند و امکان ایجاد بانک اطلاعاتی را به صورت مستقیم براساس ساختار منطقی داده‌ها را می‌دهند. بنابراین اساساً احتیاجی به چنین واسطی نداریم.

۲. اکثر نرم‌افزارها برای شرکت خاص ایجاد می‌شود و معمولاً احتیاجی به تعویض مدیریت بانک اطلاعاتی در بین نمی‌باشد. مگر نرم‌افزارهایی که به صورت عام (Package) از طرف یک کمپانی نرم‌افزاری برای فروش نوشته می‌شوند.

گام آخر گردآوری نتایج

در این گام مستندات این مرحله از جهت کامل بودن و مطابقت داشتن با یکدیگر بررسی می‌گردند و امضا می‌شوند. مستندات این مرحله عبارتند از:

- مشخصات برنامه‌ها
- طراحی بانک اطلاعاتی