



دانشکده مهندسی

سمینار دوره کارشناسی ارشد مهندسی فناوری اطلاعات

مفاهیم، راهبردها و فناوری های یکپارچه سازی برنامه های کاربردی سازمانی

نگارش

حسین مرادی

hsmoradi@gmail.com

استاد راهنما

دکتر اردشیر بحرینی نژاد

بهار 85

چکیده

یکی از جهت گیری های اصلی فناوری اطلاعات، یکپارچه سازی و اتصال سیستم ها و فناوری های اطلاعاتی جهت دست یابی به سیستم های یکپارچه و بدون درز است تا بدین وسیله، بستر مورد نیاز جهت تحقق جامعه اطلاعاتی فراهم گردد. این جهت گیری، یکپارچه سازی را به نیازی غیرقابل تردید بدل کرده است. برای پوشش این نیاز، راهبردهای مختلفی مطرح است که یکی از مهمترین آنها، بکارگیری فناوری ها و راه حل های یکپارچه سازی برنامه های کاربردی سازمانی (EAI) می باشد. فناوری های نرم افزاری EAI با فراهم ساختن امکان اشتراک اطلاعات و فرایندهای کسب و کار سیستم های اطلاعاتی مرتبط با یکدیگر، دست یابی به سیستم های یکپارچه را ممکن می کنند.

این تحقیق را می توان به دو بخش کلی تقسیم کرد. بخش اول آن که نگاهی توصیفی به مقوله EAI دارد، با بررسی مفهوم کلی یکپارچه سازی سازمانی آغاز می شود، سپس یکپارچه سازی سیستم های اطلاعاتی مورد بررسی قرار می گیرد. در ادامه انواع رویکردهای EAI مورد بررسی قرار می گیرد و پس از آن نوبت به توصیف معماری و توپولوژی توانمندسازهای EAI می رسد. سپس به تشریح، مقایسه و ارزیابی انواع راه حل ها و فناوری های مورد استفاده در EAI می پردازد. بخش دوم این تحقیق که همان نتیجه گیری است، به بررسی مباحث تحقیقاتی مطرح در زمینه EAI می پردازد، سپس محدودیت اصلی تحقیق در زمینه EAI را مورد بررسی قرار می دهد و در نهایت به بررسی جهت گیری آینده EAI و چشم انداز تحقیقات مطرح در آن می پردازد.

کلمات کلیدی: یکپارچه سازی برنامه های کاربردی سازمانی، فناوری های EAI، رویکردهای EAI، فناوری های یکپارچه سازی، برنامه کاربردی و سیستم اطلاعاتی

فهرست مطالب

صفحه

عنوان

فصل اول: یکپارچه سازی در سازمان و سیستم های اطلاعاتی

2	1-1- مقدمه
3	2-1- یکپارچه سازی در سازمان
5	3-1- یکپارچه سازی در سیستم های اطلاعاتی
8	1-3-1- یکپارچه سازی نوع BIG I
9	2-3-1- ترکیب یکپارچه سازی BIG I با یکپارچه سازی LITTLE I
11	3-3-1- یکپارچه سازی نوع LITTLE I
12	4-1- تعاریف و قابلیت های EAI
13	5-1- محرک های EAI
13	1-5-1- بسته های ERP
14	2-5-1- تجارت الکترونیک، بازارهای الکترونیکی و زنجیره تامین
15	3-5-1- ادغام و اتحاد دو یا چند شرکت
15	4-5-1- دولت الکترونیک
17	6-1- نتیجه گیری

فصل دوم: رویکردها و دسته بندی های مختلف EAI

19	1-2- مقدمه
20	2-2- انواع یکپارچه سازی بر اساس ضعیف یا قوی بودن آن
21	3-2- دسته بندی EAI بر اساس طرف های درگیر در آن
21	1-3-2- مولفه اول: یکپارچه سازی برنامه کاربردی درون سازمانی
22	2-3-2- مولفه دوم: یکپارچه سازی برنامه کاربردی بین سازمانی
23	1-2-3-2- سازمان توسعه یافته
23	2-2-3-2- سازمان مجازی
23	3-3-2- مولفه سوم: یکپارچه سازی برنامه کاربردی پیوندی
26	4-2- دسته بندی بر اساس انواع سیستم های نرم افزاری که ممکن است با هم یکپارچه شوند
28	5-2- انواع رویکردهای یکپارچه سازی بر اساس سطح مطلوب برای یکپارچه سازی
28	1-5-2- خراش صفحه
28	2-5-2- طراحی مجدد رابط
29	3-5-2- یکپارچه سازی شیئی
30	4-5-2- مهاجرت موروثی
30	6-2- انواع رویکردهای EAI بر اساس عناصر سیستم های درگیر در یکپارچه سازی
33	1-6-2- نقاط قوت و ضعف این رویکردها
36	7-2- انواع رویکردهای EAI بر اساس لایه های معماری مورد نیاز در یکپارچه سازی

- 38.....8-2- دستة بندی بر مبنای یک معماری EAI مدل محور
- 39.....1-8-2- مدل برنامه کاربردی بینی
- 39.....2-8-2- مدل برنامه کاربردی درونی
- 40.....3-8-2- مدل سرویس های برنامه کاربردی
- 40.....4-8-2- مدل سرویس های تراکنشی
- 40.....5-8-2- مدل مختص فناوری
- 41.....9-2- دستة بندی بر مبنای یکپارچه سازی افقی معماری مدل محور
- 41.....1-9-2- یکپارچه سازی برنامه کاربردی بینی
- 42.....2-9-2- یکپارچه سازی برنامه کاربردی درونی
- 43.....1-2-9-2- یکپارچه سازی سرویس های برنامه کاربردی
- 43.....1-1-2-9-2- یکپارچه سازی داده ای
- 44.....2-1-2-9-2- یکپارچه سازی شیئی
- 45.....3-1-2-9-2- یکپارچه سازی فرایندی
- 46.....2-2-9-2- یکپارچه سازی میان افزار
- 46.....1-2-2-9-2- یکپارچه سازی پلتفرم
- 46.....2-2-2-9-2- یکپارچه سازی شبکه
- 47.....10-2- انواع رویکردهای یکپارچه سازی بر اساس درجه پیچیدگی آن
- 47.....1-10-2- انتقال داده های مشترک
- 47.....2-10-2- یکپارچه سازی در سطح پایگاه داده
- 48.....3-10-2- پردازش رابط
- 48.....4-10-2- یکپارچه سازی در سطح فرایند
- 48.....5-10-2- یکپارچه سازی سرویس گرا
- 49.....6-10-2- یکپارچه سازی درون یک سازمان خدماتی
- 50.....11-2- نتیجه گیری

فصل سوم: معماری و توپولوژی توانمندسازهای EAI

- 52.....1-3- مقدمه
- 53.....2-3- معماری و لایه بندی EAI
- 54.....1-2-3- لایه خدمات اتصال
- 54.....2-2-3- لایه خدمات ارتباطی
- 55.....3-2-3- لایه خدمات توزیعی
- 55.....4-2-3- لایه خدمات تبدیلی
- 56.....5-2-3- لایه مدیریت فرایند
- 56.....3-3- چارچوب یکپارچه سازی B2B
- 58.....4-3- توپولوژی های پیاده سازی EAI

58.....	1-4-3- مدل هاب و اسپیک
60.....	1-1-4-3- مدل ایستای فدرالی
61.....	2-1-4-3- مدل پویای فدرالی
61.....	2-4-3- مدل باس شبکه ای
63.....	5-3- نتیجه گیری

فصل چهارم: تشریح، مقایسه و ارزیابی راه حل ها و فناوری های EAI

65.....	1-4- مقدمه
66.....	2-4- فناوری های یکپارچه سازی نقطه به نقطه
68.....	3-4- محصولات یکپارچه سازی برنامه کاربردی سازمانی (EAI)
70.....	4-4- آداپترها
71.....	5-4- فناوری های یکپارچه سازی پایگاه داده با پایگاه داده
73.....	6-4- فناوری های یکپارچه سازی با استفاده از انبار داده
75.....	7-4- یکپارچه سازی با استفاده از سرویس دهنده برنامه کاربردی
77.....	8-4- مقایسه XML و EDI
78.....	9-4- سرویس های وب و EAI
80.....	10-4- مقایسه EAI، EDI و سرویس های وب
82.....	11-4- مقایسه معماری سرویس گرا و EAI
84.....	12-4- چارچوبی جهت ارزیابی فناوری های یکپارچه سازی
85.....	1-12-4- ارزیابی چارچوب پیشنهادی ارائه شده
88.....	13-4- نتیجه گیری

فصل پنجم: نتیجه گیری

90.....	1-5- مقدمه
91.....	2-5- مباحث تحقیقاتی مطرح در زمینه EAI
92.....	3-5- محدودیت اصلی تحقیق در زمینه EAI
93.....	4-5- جهت گیری آینده EAI و چشم انداز تحقیقات آینده
97.....	5-5- نتیجه گیری
98.....	فهرست منابع و مأخذ

فهرست جداول

- جدول 2-1- یکپارچه سازی نوع قوی و ضعیف 20
- جدول 2-2- مشخصات زیر رده های دسته بندی بر مبنای طرف های درگیر در EAI 25
- جدول 2-3- دسته بندی انواع سیستم های نرم افزاری که ممکن است با هم یکپارچه شوند 26
- جدول 2-4- انواع رویکردهای EAI بر اساس عناصر سیستم های درگیر در یکپارچه سازی 32
- جدول 2-5- مروری بر نقاط قوت و ضعف رویکردهای مختلف EAI 34
- جدول 2-6- انواع رویکردهای EAI بر اساس لایه های معماری مورد نیاز در یکپارچه سازی 37
- جدول 4-1- مقایسه EDI و XML 77
- جدول 4-2- مزایا و معایب فناوری های یکپارچه سازی EDI، سرویس های وب و EAI 80
- جدول 4-3- مقایسه فناوری های یکپارچه سازی EDI، سرویس های وب و EAI 81
- جدول 4-4- معیارهای ارزیابی پیشنهادی 84
- جدول 4-5- ارزیابی فناوری های یکپارچه سازی 86

فهرست شکل ها

- شکل 1-1- آگهی اراکل برای Big I..... 6
- شکل 1-2- ارتباط مولفه های سیستم برای تعریف یک راه حل سازمانی 7
- شکل 1-3- مثالی از ترکیب Big I و Little i در یک سازمان 9
- شکل 1-4- یکپارچه سازی زنجیره تامین از طریق EAI..... 15
- شکل 1-2- دسته بندی EAI، بر اساس طرف های درگیر در یکپارچه سازی 21
- شکل 2-2- رویکردهای یکپارچه سازی سیستم ها بر مبنای سطح مطلوب برای یکپارچه سازی 29
- شکل 2-3- یک معماری EAI مدل محور 39
- شکل 2-4- یکپارچه سازی افقی معماری EAI..... 42
- شکل 1-3- معماری محصولات EAI..... 53
- شکل 2-3- چارچوب یکپارچه سازی B2B..... 57
- شکل 3-3- مدل هاب و اسپک برای EAI..... 59
- شکل 3-4- پیکرندی ایستای فدرالی، شامل چند سرویس دهنده های یکپارچه سازی (IS) متمایز..... 61
- شکل 1-4- مثالی از یکپارچه سازی نقطه به نقطه 67
- شکل 2-4- یک محصول EAI با معماری هاب و اسپک 69
- شکل 3-4- آداپترهای اطلاعات گرا 70
- شکل 4-4- آداپترهای سرویس گرا 71
- شکل 4-5- طرح کلی مفهومی یکپارچه سازی پایگاه داده با پایگاه داده 72
- شکل 4-6- نمای مفهومی یکپارچه سازی انبار داده ای 73
- شکل 4-7- نمای مفهومی LIDB 74
- شکل 4-8- یکپارچه سازی با سرویس دهنده برنامه کاربردی 75
- شکل 4-9- معنای علامت های استفاده شده در رتبه بندی فناوری های یکپارچه سازی 85

فصل اول

یکپارچه سازی در سازمان و سیستم های اطلاعاتی

1-1- مقدمه

مرور تاریخچه سیستم های رایانه ای، روندهای جالب توجهی را نمایان می سازد. استفاده اولیه از رایانه ها به شدت دپارتمانی بود و برنامه نویسان برای خودکارسازی وظایف هر دپارتمان به ساخت برنامه های اختصاصی می پرداختند. مشکل اصلی این برنامه ها آن بود که چون فرایندهای کسب و کار سازمانی چندین دپارتمان را در بر می گرفت، سازمان ها مجبور می شدند، جهت اشتراک اطلاعات بین دپارتمان ها، این برنامه ها را با استفاده از رابط های نقطه به نقطه به یکدیگر پیوند دهند که این منجر به ایجاد یک معماری به صورت اسپاگتی و در نتیجه افزایش پیچیدگی و بالا رفتن هزینه توسعه و نگهداری می شد. در نتیجه، سیستم های یکپارچه برنامه ریزی منابع سازمانی (ERP) به عنوان راه حلی برای این مشکل و به صورت جایگزین برنامه های کاربردی دپارتمانی مطرح شد. هدف آنها ایجاد یک راه حل جامع برای حل تمام مسائل سازمان بود. ERPها راه حل های یکپارچه ای بودند که نیاز به رابط ها را کاهش داده و سبب کاهش هزینه نگهداری کل سیستم می شدند. البته آنها تمام کارکردهای لازم برای پشتیبانی از فرایندهای کسب و کار سازمان ها را ارائه نمی دادند یا در پاره ای از موارد، جایگزینی همه برنامه های موجود سازمان با یک سیستم یا پرهزینه بود و یا اصلاً امکان پذیر نبود. بنابراین برای یکپارچه سازی سیستم های ERP با سیستم های موروثی یا سایر سیستم های ERP به چندین رابط نیاز بود. این جا بود که فروشندگان نرم افزار، راه حل هایی را با عنوان EAI توسعه دادند تا زیرساخت لازم برای اتصال و ارتباط اطلاعاتی سریع جهت یکپارچه سازی ERPها با سیستم های موروثی داخلی سازمان فراهم کند. بعدها امکان اتصال B2B نیز به

EAI اضافه شد و سازمان ها برای ارتباط با همکاران و مشتریان خود، برنامه های کاربردی و فرایندهای کسب و کارشان را کمک EAI با یکدیگر یکپارچه کردند. در این فصل، روی این تاریخچه و انواع یکپارچه سازی مرتبط با آن متمرکز می شویم.

در ابتدای این فصل به یکپارچه سازی در سازمان پرداخته شده که یکپارچه سازی سیستم های اطلاعاتی، بخشی از آن است. در ادامه، تنها روی مباحث مربوط به یکپارچه سازی سیستم های اطلاعاتی تمرکز شده و دو نوع عمده آن یعنی سیستم های یکپارچه (مانند ERPها) و یکپارچه سازی برنامه های کاربردی مورد بررسی قرار می گیرد. با توجه به این که تمرکز اصلی این کار، روی یکپارچه سازی برنامه های کاربردی سازمانی (EAI) است، در ادامه با تفصیل بیشتری روی این نوع از یکپارچه سازی تمرکز شده و در نهایت، تعاریف و قابلیت های EAI و همچنین محرک هایی که سبب استفاده سازمان ها از آن می شود، مورد بررسی قرار می گیرد.

2-1- یکپارچه سازی در سازمان

مفهوم یکپارچه سازی، در طول سالها بین رشته های¹ مختلف به طور متنوعی، درک و تصور شده و در یک روش نسبتاً غیر وایسته و غیر یکپارچه تکامل یافته است، به طوری که هر رشته ای، از منظر خود به موضوع نگاه می کند (Barki and Pinsonneault, 2005).

یکپارچه سازی یک جایگاه مرکزی در حوزه های متعددی چون مدیریت، استراتژی، تئوری سازمانی، مدیریت تولید/عملیات و سیستم های اطلاعاتی اشغال می کند (Barki and Pinsonneault 2005).

در استراتژی، ریشه های مفهومی یکپارچه سازی را می توان در مفاهیم تعاون و هماهنگی (Fayol 1949) و همچنین در تعریف Lawrence and Lorsch (1969) از یکپارچه سازی ردیابی کرد.

"... فرایند دست یافتن به پیوستگی و وحدت در تعامل بین زیر سیستمهای متعدد برای انجام وظایف سازمانی ..." (ص.34)

بنابراین از این دیدگاه، منظور از یکپارچه سازی این است که دپارتمانهای مختلف چطور با همدیگر به صورت موزونی تعامل دارند و فعالیتهايشان به صورت تنگاتنگی هماهنگ شده است. در راستای

¹ Discipline

این نگاه، ادبیات استراتژی از مفهوم یکپارچه سازی، برای شرح هماهنگی فعالیتها یا مدیریت وابستگیها بین آنها استفاده می کند (Barki and Pinsonneault 2005).

مفهوم یکپارچه سازی در سایر ادبیات، هم راستا با دیدگاه بالا است. برای مثال در تولید/ عملیات/ لجستیک، یکپارچه سازی به صورت مدیریت هماهنگ اطلاعات، گردش مواد، عملیات ماشین آلات کارخانه و لجستیک از طریق مجموعه مشترکی از اصول، استراتژی ها، سیاست ها، و معیارهای کارایی¹ دیده می شود (Barki and Pinsonneault 2005). در ادبیات نوآوری، از یکپارچه سازی برای ارائه اینکه فعالیتهای فرایند نوآوری (مثلاً تحقیق و توسعه و ساخت و تولید) چطور به همدیگر متصل شده و به طور تنگاتنگی هماهنگ شده اند، استفاده شده است (Barki and Pinsonneault 2005). در سیستم های اطلاعاتی، به مفهوم یکپارچه سازی از حداقل دو منظر نگاه می شود.

از نظر فنی، از یکپارچه سازی برای شرح به هم پیوستگی فناوریهای اطلاعاتی سازمان و این که عناصر داده ای تا چه درجه ای از یک شمای مفهومی عمومی² مشترک استفاده می کنند، استفاده می شود (Barki and Pinsonneault 2005). بر اساس این دیدگاه، یکپارچه سازی نشان می دهد که سیستمهای مختلف تا چه اندازه ای، به هم پیوسته اند و می توانند با همدیگر صحبت کنند. دیدگاه دیگر، یکپارچه سازی را در این می بیند که فرایندهای کسب و کار دو یا چند سازمان مستقل، تا چه اندازه استاندارد شده و از طریق فناوریهای مخابراتی و رایانه ای پیوند مستحکمی با همدیگر دارند³ (Barki and Pinsonneault 2005).

همانطور که مشاهده می شود، چون رشته های مختلف روی مولفه ها یا فعالیتهای سازمانی مختلفی تمرکز کرده اند، مفهوم یکپارچه سازی در حوزه های مختلف به اشکال مختلفی درک و تعریف شده است. البته با وجود تفاوت های واضح آنها، مفهوم یکپارچه سازی حول یک مفهوم بنیادی سیر می کند که Braki and Pinsonneault آنرا یکپارچه سازی سازمانی⁴ (OI) می نامند و آنرا به این شکل تعریف می کنند که مولفه های سازمانی متمایز و وابسته به یکدیگر، تا چه اندازه تشکیل یک کل

¹ Performance metric

² Common conceptual schema

³ Tightly couple

⁴ Organization Integration

متحد¹ می دهند. در این تعریف، واژه مولفه به واحدها، دپارتمانها، یا همکاران سازمانی اشاره دارد و شامل فرایندهای کسب و کار، افراد و فناوری درگیر در آن است (Barki and Pinsonneault 2005).

1-3- یکپارچه سازی در سیستم های اطلاعاتی

در این کار، تمرکز اصلی روی یکپارچه سازی در سیستم های اطلاعاتی است، اما با توجه به اینکه، یکپارچه سازی سیستم های اطلاعاتی در خلا صورت نمی گیرد و روی مولفه های سازمانی تاثیرات مستقیم یا غیر مستقیمی دارد، برای پیاده شدن کامل یکپارچه سازی سیستم های اطلاعاتی باید در بستر یکپارچه سازی سازمانی پیاده شود.

آگهی زیر (شکل 1-1) را از شرکت اراکل و نقل قول مربوطه را از CEO آن، لاری الیسون² در نظر بگیرید. این شکل، به طور واضح و مشخص، تبلیغ نوعی از یکپارچه سازی است که Gulledge (2006) آن را "Big I" می نامد که در آن تمام داده های مرتبط با هم، بر اساس یک مدل داده ای معین تراز شده و تنها یک بار ذخیره شده اند. معنای آن، این است که شما می توانید تمام داده های خود را برای مجموعه ای از فرایندهای کسب و کار که در ستون میانی شکل 1-1 لیست شده اند، درون سویت کسب و کار الکترونیک اراکل³ قرار دهید و به طور قابل توجهی، هزینه کلی مالکیت⁴ (TCO) سیستم را کاهش دهید. در حقیقت، آگهی ادعا می کند که اراکل با پیاده کردن Big I، سالانه یک میلیارد دلار صرفه جویی کرده است. معنای دیگری که از نقل قول الیسون بر می آید، این است که اگر شما با استفاده از رابط⁵، ترکیبی از بهترین ماژول های محصولات مختلف⁶ را پیاده کنید، با مشکلات پیاده سازی روبرو خواهید شد؛ یعنی "هیچ اهمیتی ندارد، شما چقدر خوب آن را انجام دهید، در هر صورت یکپارچه سازی سیستم ها هرگز نمی تواند تمام داده های شما را در یک مکان قرار دهد" (Ferguson and Callaghan 2002, Gulledge 2006).

Unified whole¹

Larry Ellison 2

Oracle E-Business Suite 3

Total Cost of Ownership 4

Inteface⁵

Best of Breed 6

Using our own E-Business Suite,
Oracle saved
\$1 billion in 1 year.

E-Business Kit	E-Business Suite	
Microsoft	Database	Oracle
Epiphany	Marketing	Oracle
Sibel	Sales	Oracle
Clarify	Support	Oracle
IBM	Webstore	Oracle
Commerce One	Procurement	Oracle
SAP	Manufacturing	Oracle
i2	Supply Chain Mgmt	Oracle
SAP	Financials	Oracle
PeopleSoft	Human Resources	Oracle

**All Relevant Data in
the Same domain**



**"No matter how well you do it,
systems integration can never
get all of your data together in
one place," Larry Ellison***

How much will you save?



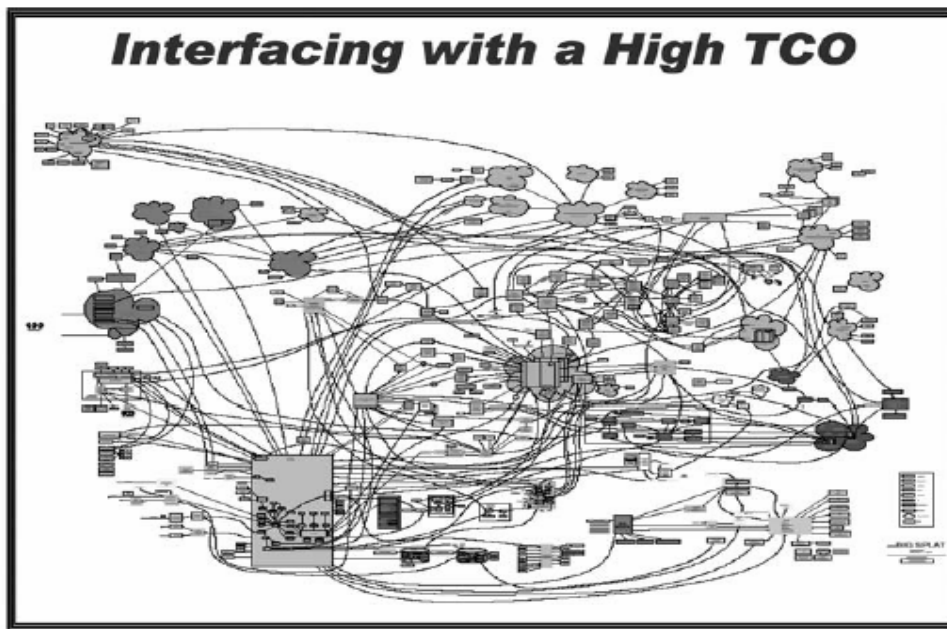
www.oracle.com

*In R.B. Ferguson and D. Callaghan, Tools to Ease Integration, eWeek, Vol. 19 #15 (April 15, 2002), p.12.

شکل 1-1- آگهی اراکل برای Big I

نقل قول الیسون یک نکته دارد - رابط ها گران هستند و بخش مهمی از هزینه هر پروژه پیاده سازی سیستم های سازمانی را تشکیل می دهند. در ضمن مشکلات پیچیدگی کلی و مدیریت جامعیت¹ چند منبع داده ای نیز وجود دارد (Gulledge 2006). شکل 1-2 را از یک شرکت بدون نام در نظر بگیرید. شکل 1-2 یک موقعیت را نشان می دهد که در ادبیات، "یکپارچه سازی سیستم"² نامیده می شود که این به معنای ارتباط سیستم ها با همدیگر به شکلی است که بتوانند اطلاعات را از بین یک تصویر تکنولوژیک پیچیده، به همدیگر ارسال کنند. Gulledge این نوع از یکپارچه سازی را، یک شکل از Little i می نامد و بیان می کند که این شکل از Little i (نقطه به نقطه³)، یک راه حل گران است. داده ها بایستی به طور ثابت بین چندین منبع داده ای هماهنگ⁴ و تمیز⁵ شوند و هر تغییری روی یک سیستم می تواند منجر به آزمایش مجدد یا حتی طراحی مجدد و کد نویسی رابط ها شود که کاری پیچیده و پرهزینه است (Gulledge 2006).

-
- Integrity 1
 - System integration 2
 - point-to-point 3
 - Harmonize 4
 - Cleanse 5



شکل 1-2- ارتباط مولفه های سیستم برای تعریف یک راه حل سازمانی

تا اینجا دو هدف نهایی بیان شد که که البته بسیاری از سازمانهای بزرگ با وسعت جهانی، هیچ کدام از آنها را به صورت مطلق نپذیرفته اند. بسیاری از سازمان ها، به دلایل متعددی نمی خواهند تمام داده هایشان را درون یک برنامه کاربردی (به طور مثال اراکل، SAP، مایکروسافت و غیره) قرار دهند، اما در عین حال، هیچ کدام دوست ندارند با مشکلات پیاده سازی Little i، مشابه آنچه که در شکل 1-2 نشان داده شد، مواجه شوند (Gulledge 2006).

انتخاب های دیگری نیز وجود دارد. در حقیقت انتخابهای بسیاری وجود دارد، که تمام انتخاب ها (شامل دوتایی که در بالا ذکر شد) یکپارچه سازی خوانده می شوند. Big I، ممکن است دست یافتنی نباشد، و یا حتی مناسب هم نباشد. سوالی که در اینجا مطرح می شود اینست که اگر Little i مناسب است، بر اساس موقعیت و حالت فناوریهای پدیدار شده، چه نوعی از Little i مناسب است (Gulledge 2006). در ادامه این سوالات با تفصیل بیشتری مورد بررسی قرار می گیرند.

1-3-1- یکپارچه سازی نوع Big I

منظور از این یکپارچه سازی این است که تمام داده های مرتبط با یک مجموعه دقیق و مشخص از فرایند های کسب و کار توسط برنامه کاربردی نرم افزاری یکسانی پردازش شوند. هر به روز رسانی در یک ماژول یا مولفه از برنامه کاربردی، بدون نیاز به رابط های خارجی پیچیده، در کل منطق فرایند کسب و کار انعکاس می یابد. داده ها یکبار ذخیره شده و بطور آنی با تمام فرایندهای کسب و کار که توسط برنامه کاربردی نرم افزاری، ارائه شده اند، به اشتراک گذاشته می شوند.

این یک تعریف نسبتاً جامع و محدود کننده است که خاطره سیستم های برنامه ریزی منابع سازمانی¹ (ERP) نسل اول را زنده می کند (Gulledge 2006). البته سیستم هایی با عناوین سیستم های یکپارچه نیز، عمدتاً بر پایه این تعریف طراحی می شوند.

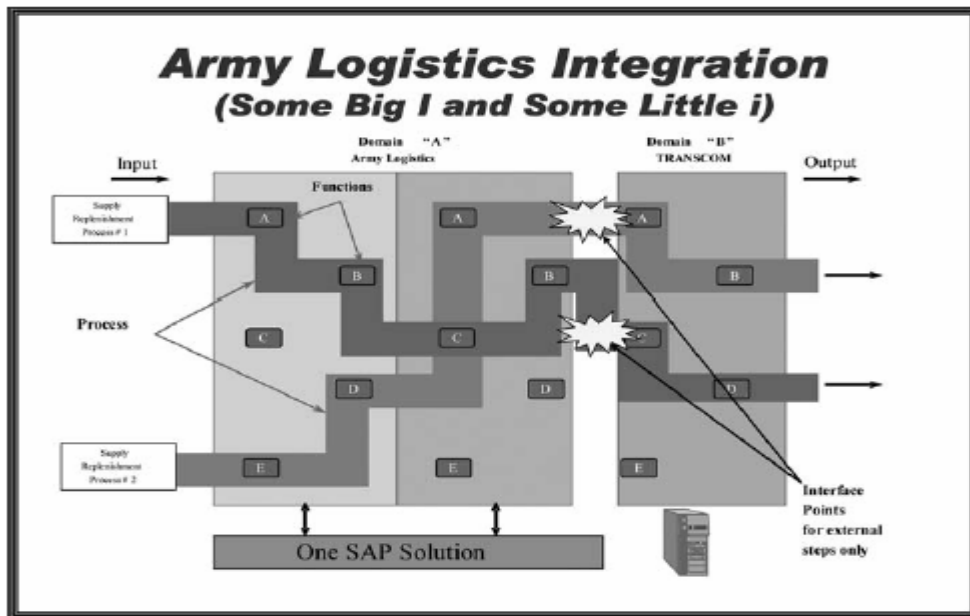
Osterle (1996)، این نوع از سیستم ها را، سیستم های متجانس با یک نمونه² می نامد و بیان می کند که این مدل از مشکلاتی که به علت افزونگی در ذخیره داده ها و تبادل غیر همزمان داده ها بین برنامه های کاربردی مجزا، ممکن است پدیدار شود، جلوگیری می کند (Puschmann and Alt 2004).

به طور قطع و یقین، هدف مدیریت، خصوصاً برای فرایندهای کسب و کار Big I است. دلیل این امر نیز ساده است، تنها "یک منبع درستی"³ وجود دارد که کل فرایندهای کسب و کار سازمان که توسط راه حل های ERP ارائه می شود، از آن استفاده می کنند. مفهوم ساده است: اگر تمام داده ها، یک بار ذخیره شده و به اشتراک گذاشته شوند، آنگاه احتمال وقوع مسایل جامعیتی⁴ نیز کمتر خواهد بود. باتوجه به این که به رابط های بین مولفه های نرم افزاری نیازی ندارید، TCO به طور قابل توجهی کمتر است، بنابراین پیچیدگی به مقدار زیادی کاهش می یابد (Gulledge 2006).

¹¹ Enterprise Resource Planning
² Homogeneous with one instance
³ One source of truth
⁴ Integrity issues

1-3-2- ترکیب یکپارچه سازی Big I با یکپارچه سازی Little i

شکل 1-3 نشان می دهد که چطور برای یک مثال ساده که مربوط به لجستیک ارتش آمریکا است، Big I با Little i مرتبط شده است.



شکل 1-3-1- مثالی از ترکیب Big I و Little i در یک سازمان

در این مثال، فرایندهای لجستیک ارتش، توسط راه حل SAP به صورت Big I تحت پوشش قرار گرفته، یعنی اینکه، هیچ رابطی بین مولفه های SAP وجود ندارد. اما تعدادی از فرایندهای کسب و کار لجستیک، در خارج از ارتش جریان دارند. در این نمونه، فرایندهای حمل و نقل که بخشی از فرایندهای کسب و کار لجستیک آنها-تا-انتها¹ هستند، مشخص شده که آنها در خارج از ارتش قرار می گیرند و به وسیله فرماندهی حمل و نقل آمریکا² (TRANSCOM) مدیریت می شوند. سیستم هایی که این بخش از فرایند آنها تا انتها را پوشش می دهند، SAP نبوده و حتی در مالکیت ارتش نیز قرار ندارند. این نمونه ای از برنامه کاربردی ترکیبی³ است که باید به منظور حفظ جامعیت منطق فرایند کسب و کار از Little i استفاده شود.

¹ End-to-end
² Transportation Command
³ Composite application

شکل 1-3، حتی اگر یک تصویر ساده باشد، اما مطالب زیادی را درباره یکپارچه سازی نشان می دهد. نخست آن که به این مطلب اشاره می کند که سازمانهای بزرگ و پیچیده، تمایل ندارند، تمام فرایندهای کسب و کارشان را در یک برنامه کاربردی منفرد قرار دهند. اگرچه ادعای شکل 1-1 درست است¹، اما حداقل سه دلیل وجود دارد که اکثر شرکتهای، نتوانند تنها از یک نمونه منفرد ERP استفاده کنند:

1- مشکلات فرهنگی و کنترل اجتماعات یکپارچه سازی داخلی و خارجی، اجازه چنین به هم پیوستگی را نخواهد داد (Gulledge 2006).

2- ناتوانی یک تولید کننده نرم افزارهای استاندارد برای تحویل 100 درصد نیازهای نرم افزاری سازمان های امروزی، شرکت ها را مجبور به استفاده از چند معماری IS نامتجانس کرده (Puschman and Alt 2004) و آنها را مجبور کرده تا به سمت یکپارچه سازی Little i حرکت کنند.

3- یکپارچه سازی با همکاران کسب و کار، مانند مشتریان و تامین کنندگان که دو شرکت لزوماً از زیر ساخت سیستم اطلاعاتی همگنی برخوردار نیستند (Puschman and Alt 2004).
خوشایند باشد یا نه، با توجه به وضعیت فعلی فناوری، حرکت به سمتی است که مجبوریم با ترکیبی از Big I و Little i زندگی کنیم، لااقل تا زمانی که روند جاری ادامه دارد. با توجه به این حقیقت که ارائه دهندگان نرم افزاری بزرگتر، محصولاتشان را باز می کنند و انعطاف آنها را برای ترکیب بیشتر می کنند و به این شکل فرصت هایی برای Little i فراهم می کنند، واقعیت داشتن این وضعیت تقویت می شود. این حقیقت در محصولات تکنولوژیکی مانند Oracle Data Hubs و SAP NetWeaver مشهود است. درست است که با حرکت به سمت Big I، TCO می تواند کاهش یابد، اما بسیاری از سازمان ها، انعطاف یا علاقه ای برای انجام آن ندارند.

البته معنای این سخن، این نیست که Big I مرده است. همیشه بسته هایی از Big I وجود دارند که باید توسط Little i به بسته های دیگری از Big I متصل شوند. این فقط یک ادعای فنی نیست، بلکه مستقیماً به عقل سلیم وابسته است. برای مثال، هیچ کس نمی خواهد یک محصول مانند هسته ERP

1 منظور آگهی شرکت اراکل است

SAP را بشکافد و سپس مجدداً اجزای آن را با همدیگر ارتباط دهد. باید تلاش شود با پیاده کردن Big I، تا جای ممکن از جامعیت محصول نگهداری شود و از Little i صرفاً برای قرار دادن مولفه هایی که نتوانسته اند در قلمرو یکپارچه سازی قرار بگیرند، استفاده شود. هیچ کس در رویای جداکردن مولفه مالی از مواد و مجدداً ارتباط دادن آنها در هنگام پیاده سازی SAP نیست. یا حتی بدتر از آن، از عقل به دور است که راه حل SAP را به صورت خانواده ای از سیستم های مستقل، در بخشهای مختلف شرکت، طوری پیاده کنید که بدون حضور یک جهت گیری سازمانی فعالیت کنند (Gulledge 2006).

3-3-1- یکپارچه سازی نوع Little i

همانطور که قبلاً ذکر شد، تمام روش های Little i، چند نوع از برقراری ارتباط، با استفاده از رابط است، حتی اگر به طور بی ربطی به آن یکپارچه سازی سیستم¹ گویند (Gulledge 2006).

Osterle (1996) دو نوع از Little i را شناسایی کرده که عبارتند از:

- **سیستم متجانس با چندین نمونه²**: در این نوع، چند برنامه کاربردی مشابه که روی رایانه های متفاوتی قرار داشته و از نظر منطقی از پایگاه داده ی مجزایی استفاده می کنند، از چند فرایند مشابه در چند واحد کسب و کار متفاوت پشتیبانی می کنند.
- **سیستم نامتجانس³**: در این نوع، چند فرایند متفاوت، که در واحدهای کسب و کار متمایزی اجرا می شوند، توسط برنامه های کاربردی متفاوتی پوشش داده می شوند. مشکل این نوع از یکپارچه سازی، در مقایسه با محیط متجانس، این است که برنامه های کاربردی مربوطه، بر اساس مدل های داده ای مختلفی ساخته شده اند، به این معنا که آنها معنای⁴ متفاوتی از داده هایی که قرار است مبادله شوند، ارائه می دهند (Puschmann and Alt 2004).

یکپارچه سازی نوع Little i که در این تحقیق، قرار است روی آن تمرکز شود، در ادبیات با اسامی یکپارچه سازی برنامه های کاربردی سازمانی⁵ (EAI) یکپارچه سازی سیستم های اطلاعاتی،

¹ System integration

² Homogeneous with several instances

³ Heterogeneous

⁴ Semantic

⁵ Enterprise Application Integration

یکپارچه‌سازی برنامه های کاربردی¹ و غیره شناخته می شود. در ادامه کار، به معرفی و مرور ادبیات این نوع از یکپارچه سازی می پردازیم. جهت جلوگیری از ابهام، به یاد داشته باشید که هر جا از واژه یکپارچه سازی برنامه کاربردی سازمانی (EAI) یا یکپارچه سازی، استفاده می شود، منظور ما little i است، مگر این که صریحاً به نوع دیگر یکپارچه سازی، یعنی Big I اشاره شود.

1-4- تعاریف و قابلیت های EAI

تعاریف متعددی از EAI در ادبیات وجود دارد و اگرچه نویسندگان بسیاری، آن را به صورت متمایز نسبت به دیگران تعریف کرده اند، اما تمام این تعاریف ریشه مشترکی دارند. EAI در ابتدا، تنها روی یکپارچه سازی ERP با سایر برنامه های کاربردی داخل سازمان تمرکز داشت، اما امروزه، واژه ای است که تمام جوانب یکپارچه سازی در کسب و کار را پوشش می دهد (Qureshi 2005, Green, EAI). EAI را به صورت یکپارچه سازی سیستم های کاربردی غیر مشابه جهت به اشتراک گذاشتن اطلاعات توسط یک رابط کاربری مشترک تعریف می کند (Green 2000, Erasala et al. 2003). Linthicum از منظر فرایندی به آن نگاه می کند و EAI را به صورت اشتراک نامحدود اطلاعات و فرایندهای کسب و کار، بین تمام سیستم های اطلاعاتی متصل به هم در سازمان تعریف می کند (Linthicum 2000a, Linthicum 2000b, Erasala et al. 2003). Taylor احساس می کند که EAI، سیستم های اطلاعاتی را به سمت یک پلتفرم، برای پشتیبانی از تجارت الکترونیک سوق می دهد (Taylor 2000, Erasala et al. 2003). Erasala و دیگران (2003)، EAI را به این صورت تعریف می کنند:

یکپارچه سازی برنامه های کاربردی جهت اشتراک اطلاعات و فرایندهای کسب و کار که منجر به انجام موثر عملیات و تحویل انعطاف پذیر خدمات کسب و کار به مشتری می شود. با پیاده سازی EAI، از برنامه های کاربردی جاری یا موروثی به صورت خشک و غیر قابل تغییری، دست برداشته نمی شود، بلکه برعکس، ارزش این برنامه های کاربردی آزاد شده و به صورت مقیاس پذیر و مستحکمی از عملکرد آنها بهره برداری می شود.

EAI فعالیتی است که برنامه های کاربردی، فرایندها و وظایف² کسب و کار مجزای داخل سازمان را یکپارچه و هماهنگ می کند تا برنامه های کاربردی، عملکردها و سرویس های کسب و کار، به

Application integration¹
Functions²

صورت عمومی و قابل اشتراکی در معرض نمایش قرار بگیرد. مزایای پیاده سازی پروژه های EAI واضح است، اول آنکه وظایف و سرویس های بهتری نسبت به سیستم های انفرادی عرضه می شود. دوم آنکه افزونگی داده ای¹ و همپوشانی وظیفه ای² کاهش می یابد، بنابراین، جامعیت و سازگاری داده ای با درجه بالاتری تضمین می شود (Themistocleous and Irani 2001, Mosawi et al. 2006).

EAI فرایند هماهنگ کردن عملیات برنامه های کاربردی مختلف سرتاسر سازمان است. این واژه، اغلب به مجموعه ای از برنامه های کاربردی اشاره دارد که برای تسهیل این فرایند طراحی می شوند (Harikumar et al. 2005). EAI یک فناوری است که در پس زمینه اجرا می شود و در حالی که از فرایندهای کسب و کار سازمان، پشتیبانی می کند، مستقیماً برای عموم کارمندان، کاربران و یا ذینفعان سازمان قابل مشاهده نیست. EAI یک فناوری معین نیست و اغلب به سببی از فناوری های مختلف نیاز دارد. می توان گفت EAI چارچوبی برای ترکیب چندین فناوری یکپارچه سازی مانند دلال پیغام³، آداپتر⁴ و سرویس دهنده های برنامه کاربردی⁵ است (Janssen and Cresswell 2005).

5-1- محرک های EAI

چهار عامل اصلی زیر، سبب تحریک سازمان ها برای استفاده از EAI می شود:

1-5-1- بسته های ERP

رشد استفاده از بسته های ERP به وسیله شرکتها، اغلب نیاز به EAI را بوجود می آورد. با توجه به اینکه، ممکن است به دلایل متعددی نتوان برنامه های کاربردی موروثی موجود را کنار گذاشت، به اجبار باید از EAI برای یکپارچه سازی سیستم های ERP با برنامه های موروثی استفاده کرد. (Erasala et al. 2003). علاوه بر این، با توجه به اینکه، بسیاری از پیاده سازی های ERP، برای نائل شدن به اهداف اعلان شده، نیاز به سفارشی سازی سنگینی دارند، می توان نیاز به این سفارشی سازی

¹ Data redundancy
² Function overlapping
³ Message Broker
⁴ Adapter
⁵ Application Servers

را با انتخاب بهترین مازول ها از هر فروشنده و یکپارچه سازی آنها، با استفاده از EAI، برای تشکیل یک سیستم یکپارچه، به حداقل کاهش داد (Alshawi et al. 2004).

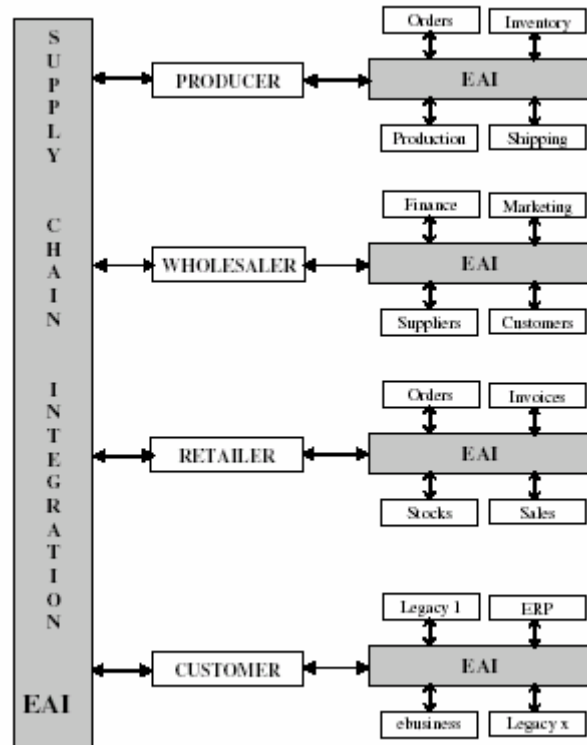
1-5-2- تجارت الکترونیک، بازارهای الکترونیکی و زنجیره تامین

یک مسئله که کسب و کارهای امروز، با آن روبرو هستند، به دست آوردن فرصت های ارائه شده به وسیله تجارت الکترونیک و بازارهای الکترونیکی، با حداقل خطر و حداکثر استفاده از سرمایه گذاری های موجود در زمینه سیستم های اطلاعاتی می باشد که EAI به این مسئله پاسخ می دهد. تجارت الکترونیک، شرکت ها را مجبور کرده تا بعضی از فرایندهای کسب و کارشان را (مانند تدارکات، پردازش سفارش فروش، سرویس دهی به مشتری، مدیریت زنجیره تامین (SCM)، پشتیبانی از مشتری) با فرایندهای شرکت های همکار و مشتریان، یکپارچه کنند. این مسئله، سبب نیاز به یکپارچه سازی در سطح فرایندهای کسب و کار داخلی شرکت ها و همچنین فرایندهای بین سازمانی شرکت هایی می شود که تشکیل یک سازمان توسعه یافته¹ می دهند. تحقق این اهداف، بدون یکپارچه سازی عملکرد سیستم های داخلی با برنامه های کاربردی مبتنی بر وب، امکان پذیر نیست. در نهایت، به یکپارچه سازی تراکنش ها از طریق اینترنت، اینترنت یا اکسترانت می رسیم و "درگیر یکپارچه سازی برنامه های کاربردی موروثی و در معرض نمایش گذاشتن آنها به صورت پیش خوانهایی² می شویم که استفاده از آنها آسان بوده و مبتنی بر وب باشند" (Sealey 2000, Erasala et al. 2003). با ظهور زنجیره های تامین، نیاز به یکپارچه سازی در سطح درون سازمانی و بین سازمانی، جهت یکپارچه سازی همکاران زنجیره تامین، مانند تولید کنندگان، عمده فروشان، خرده فروشان و مشتریان با استفاده از فناوری های EAI افزایش می یابد. در صورت پیاده شدن یکپارچه سازی با پیوند قوی³ بین سازمانها، آنها به صورت یک سازمان مجازی⁴ عمل خواهند کرد. در شکل 1-4، یکپارچه سازی زنجیره تامین را از طریق EAI مشاهده می کنید (Themistocleous et al. 2004).

¹ Extended enterprise
² Front-end
³ Tightly Integration
⁴ Virtual organization

1-5-3- ادغام و اتحاد دو یا چند شرکت

در سالهای اخیر، شاهد فعالیتهای عمده ای در زمینه ادغام شرکت ها و بهره برداری از آنها بوده ایم.



شکل 1-4- یکپارچه سازی زنجیره تامین از طریق EAI

محققان دریافته اند که برنامه های کاربردی شرکت های ادغام شده، معمولاً روی پلتفرم های متفاوتی اجرا می شوند و سازمان ها برای به هم پیوستن این برنامه های کاربردی و پلتفرم های ناهمگن به یک روش ارزان، مشکل دارند. EAI به حل مشکلات این سازمان ها کمک می کند (Charles 1999, Eshel 2000, Erasala et al. 2003).

1-5-4- دولت الکترونیک

برای نائل شدن به اهداف ارائه بهتر خدمات دولتی یکپارچه و مسئولانه به شهروندان، نیاز است از برنامه های کاربردی مستقل به سمت معماری های فراگیرتر و یکپارچه تر حرکت شود. در نتیجه، یک فشار پایه ای برای یکپارچه سازی سیستم های مختلف موسسات و بخش های عمومی وجود دارد، تا از تبادل اطلاعات و فرایندهای کسب و کار بین سازمانی پشتیبانی شود. سیستم ها و پلتفرم های ناسازگار و هزینه های بالای نگهداری، سبب شده، نتوان از مزایای اشتراک و تبادل اطلاعات

برای بهره بردن از ارزش و قدرت واقعی سیستم های اطلاعاتی یکپارچه استفاده کرد. برای رسیدن به این اهداف، بایستی از EAI سود برد (Janssen and Cresswell 2005).

شرکت ها در چند سال گذشته با ایده یکپارچه سازی دست و پنجه نرم کرده اند. تحقیقات Forrester تخمین می زند که 35% از زمان توسعه راه حل های یکپارچه سازی، به ایجاد رابط و نقاط یکپارچه سازی، برای برنامه های کاربردی و منابع داده ای اختصاص می یابد. این مشکل، برای سیستم های مشتری/ سرویس دهنده سنتی¹، که ساخت آنها ارزان و یکپارچه سازی و نگهداری آنها مشکل است، به طور دردآوری تشریح شده است (Aberdeen Group 1998, Erasala et al. 2003).

همان طور که شرکت ها به جای برنامه های کاربردی دپارتمانی، به سمت فرایندهای کسب و کار حرکت می کنند، نیاز به یکپارچه سازی تشدید شده و شرکت ها، مقادیر بیشتری از درآمد خود را برای برآورد این نیازها هزینه کرده اند. گزارش Aberdeen Group نشان می دهد، که فرایندهای سازمانی گسترش یافته²، از سال 1998 تا 2000، منجر به 100 درصد افزایش درآمد شده است. در همان دوره، گردش فرایند، سبب رشد 150 درصدی در درآمدها شده است. بنابراین، نیاز به یکپارچه سازی مبرم است و EAI فقط یک واژه مسطوح، که توسط یک گروه به کار برده شود، نیست، بلکه راه حلی برای مشکلات واقعی و جدی است (Aberdeen Group 1998, Erasala et al. 2003).

بر اساس بررسی³ گروه Conspectus، بسیاری از سازمانها، با وجود سرمایه گذاری وسیع در جدیدترین فناوری های نرم افزارهای کاربردی، برای کاربردهایی که برای رسیدن به ماموریتشان حیاتی است، به سیستم های موروثی خود تکیه دارند. بیش از یک سوم کسانی که به این بررسی پاسخ داده اند، این را ضروری دانسته اند که بجای سرمایه گذاری روی سیستم های جدید، باید سیستم های موروثی را نگهداری کرد. مشکل یکپارچه سازی با سیستم های موروثی، به عنوان یکی از موانع اصلی توسعه IT ظاهر شده است. اما تمرکز روی کسب و کار الکترونیک و یکپارچه سازی سازمانی، به سازمان ها فرصتی تمام عیار می دهد تا با پیوند سیستم های موروثی و جدیدترین فناوری های کاربردی، به سمت یک سازمان مبتنی بر وب، حرکت کنند (Rowell 2000, Erasala et al. 2003).

Traditional client/server¹
Extended enterprise processes²
Survey³

1-6- نتیجه گیری

یکپارچه سازی در سیستم های اطلاعاتی را می توان به دو دسته کلی سیستم های یکپارچه (Big I) و یکپارچه سازی برنامه های کاربردی (Little i) تقسیم کرد. در سیستم های یکپارچه، یک پایگاه داده مرکزی یا به عبارت دیگر، یک منبع درستی وجود دارد که تمام فرایندهای کسب و کار برای اشتراک اطلاعات از آن استفاده می کنند، بنابراین تغییر در یکی از فرایندهای کسب و کار سازمان، به راحتی در سایر فرایندها انعکاس می یابد. اما در یکپارچه سازی برنامه های کاربردی (EAI)، چند سیستم اطلاعاتی سازمانی، برنامه کاربردی یا پایگاه داده وجود دارد که چون آنها را نمی توان به آسانی کنار گذاشت، باید از قدرت آنها استفاده کرده و از به هم پیوستن آنها، یک سیستم یکپارچه به دست آورد. در این حالت، چون اطلاعات در منابع داده ای مختلفی ذخیره شده اند، کل فرایندهای کسب و کار سازمان، تنها از "یک منبع درستی" استفاده نمی کنند، برای تغییر سایر فرایندها، باید همگام سازی داده ای انجام بگیرد. تفاوت دیگر این نوع از یکپارچه سازی با سیستم های یکپارچه، این است که چون ارتباطات در آن با استفاده از رابط ها انجام می شود، هزینه توسعه و نگهداری سیستم، بالاتر است.

در سیستم های اطلاعاتی، هدف این است که با پیاده سازی یک سیستم یکپارچه مانند ERP، تا جای ممکن، جامعیت محصول حفظ شده و از EAI صرفاً برای پیوند مولفه ها یا سیستم هایی استفاده شود که نتوانسته اند در قلمرو سیستم یکپارچه قرار بگیرند. بنابراین، EAI بیش از آنکه یک اختیار باشد، یک اجبار است، چون بایستی برای بهره بردن از قدرت سایر سیستم ها و رسیدن به یک جامعه اطلاعاتی یکپارچه از آن استفاده کرد.

فصل دوم

رویکرد ها و

دسته بندی های

مختلف EAI

2-1- مقدمه

پیاده سازی EAI، کاری پیچیده است که با چالش های تکنولوژیک و سازمانی درگیر است. از لحاظ تکنولوژیک، EAI درگیر یکپارچه سازی فناوری های ناسازگار و نامتجانس است. از لحاظ سازمانی، EAI درگیر یکپارچه سازی فرایندها و وظایف ناسازگار کسب و کار است. از طرف دیگر، تغییر سریع کسب و کار و نیاز هر سازمان به ارتباط و اشتراک اطلاعاتی با سایر سازمان ها، به این معنا است که سازمان ها به طور مستمر با چالشی به نام EAI روبرو خواهند بود. بر این اساس، در دهه گذشته، در پاسخ به این نیازها، راه حل های EAI برای پشتیبانی از یکپارچه سازی برنامه های کاربردی سازمانی، توسعه داده شدند. این راه حل ها، مکانیزم ها و زیرساخت لازم برای گسترش و یکپارچه سازی سیستم های قدیمی و جدید را ارائه می دهند. نکته قابل توجه، این است که راه حل های EAI در لایه های زیرین خود از رویکردهای متنوعی استفاده کرده و مفاهیم و اصطلاحات متفاوتی را به کار می گیرند. تنوع در رویکردهای یکپارچه سازی برنامه های کاربردی، سبب تنوع در راه حل های EAI و سخت تر شدن انتخاب مناسب ترین راه حل شده است. برای حل این مشکل، در این فصل بعد از مطالعه ادبیات رویکردهای یکپارچه سازی موجود، دسته بندی ها و رویکردهای اصلی آن شناسایی و به صورت خلاصه گردآوری شده است. تعدادی از این رویکردها، EAI را صرفاً از جنبه تکنولوژیک، تعدادی دیگر از جنبه سازمانی و سایر رویکردها هم، از هر دو جنبه بررسی کرده اند. در این فصل، مروری کوتاه بر هر رویکرد و قابلیت ها، شباهت ها، نقاط قوت و ضعف این رویکردها نسبت به هم خواهیم داشت.

2-2- انواع یکپارچه سازی بر اساس ضعیف¹ یا قوی² بودن آن

نوع یکپارچه سازی (ضعیف یا قوی بودن)، یک مسئله حیاتی است که هنگام تصمیم گیری برای یکپارچه سازی برنامه های کاربردی، بایستی مورد توجه سازمان ها قرار گیرد. همانطور که در جدول 2-1، مشاهده می کنید، تفاوت های این دو نوع از یکپارچه سازی، در ادبیات به وسیله Themistocleous and Irani (1999) تشریح شده است (Themistocleous et al. 2004). در مهندسی نرم افزار، یکپارچه سازی قوی، معادل مفهوم پیوند قوی³ و یکپارچه سازی ضعیف معادل پیوند ضعیف⁴ است.

جدول 2-1- یکپارچه سازی نوع قوی و ضعیف

یکپارچه سازی ضعیف	مرجع
تمرکز روی تبادل/ اشتراک داده ها بین همکاران	Kalakota and Robinson 1999
درجه پایین از وابستگی بین فرایندها	Loinsky 1995
درجه پایین از یکپارچه سازی	Brown 2000
توسعه یک زیرساخت یکپارچه متجانس بین سازمانی، مهم نیست	Helm 1999
ارتباطات غیرهمزمان ⁵	Puschmann and Alt 2001
رسیدن به سازمان توسعه یافته ⁶ در یکپارچه سازی بین سازمانی	Irani et al. 2003
یکپارچه سازی قوی	مرجع
تمرکز روی یکپارچه سازی فرایندها و سیستم های کسب و کار بین سازمانی	Kalakota and Robinson 1999
بالاترین درجه از وابستگی بین فرایندها	Loinsky 1995
درجه بالایی از یکپارچه سازی	Brown 2000
توسعه یک زیرساخت یکپارچه متجانس بین سازمانی، مهم است	Helm 1999
ارتباطات همزمان ⁷	Puschmann and Alt 2001
رسیدن به سازمان مجازی ⁸ در یکپارچه سازی بین سازمانی	Irani et al, 2003

¹ Loose

² Tight

³ Tightly coupling

⁴ Loosely coupling

⁵ Asynchronous communication

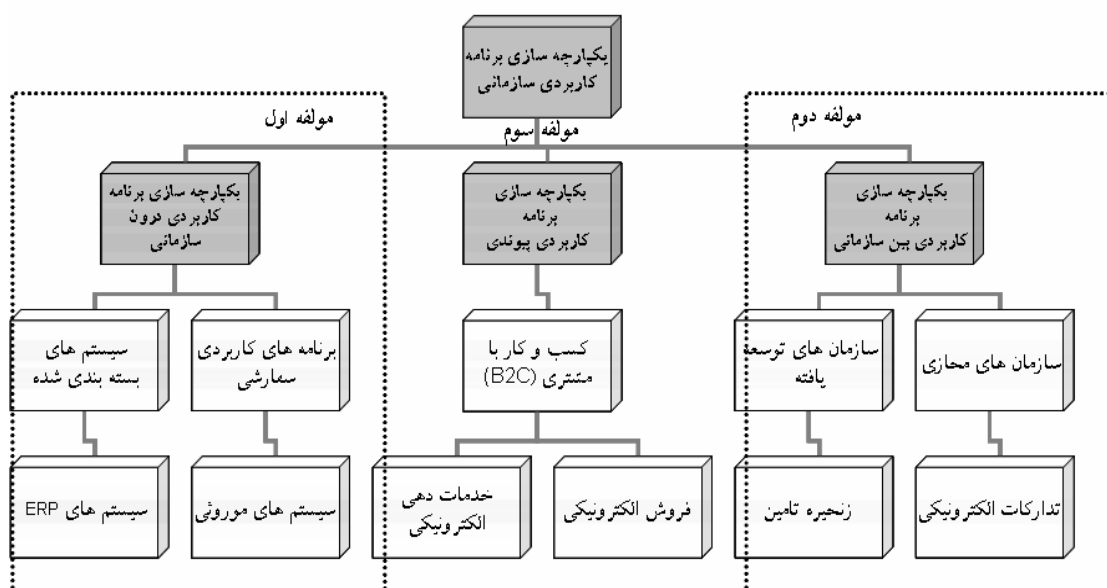
⁶ Extended enterprise

⁷ Synchronous communication

⁸ Virtual enterprise

3-2- دسته بندی EAI بر اساس طرف های درگیر در آن

Irani و دیگران (2003)، بر اساس طرف های درگیر در یکپارچه سازی (سازمان، سازمان های همکار و مشتریان)، یک دسته بندی از EAI ارائه داده است که آن را در شکل 1-2 مشاهده می کنید. این دسته بندی به مدیران این امکان را خواهد داد تا فناوری هایی را که می توان برای برنامه های کاربردی سازمانی و بین سازمانی استفاده کرد، شناسایی کنند، که این در نهایت، می تواند منجر به توسعه یک زیر ساخت یکپارچه شود که از برنامه های کاربردی درون و بین سازمانی پشتیبانی کند (Irani et al. 2003).



شکل 1-2- دسته بندی EAI، بر اساس طرف های درگیر در یکپارچه سازی

3-2-1- مؤلفه اول: یکپارچه سازی برنامه کاربردی درون سازمانی

سیستم های سفارشی و بسته بندی شده، به عنوان زیر رده برنامه های کاربردی درون سازمانی، دسته بندی شده اند (Handfield, Nichols 1999, Irani et al. 2003). Brodie and Stonebraker گزارش کرده اند که سیستم های سفارشی یا موروثی، در مقابل تغییرات و حرکات انقلابی برای برآورد نیازمندی های کسب و کار مقاومت می کنند (Brodie and Stonebraker 1995, Irani et al. 2003). بسیاری از سیستم های موروثی، از یک مدل واحد¹ پیروی می کنند که داده ها، منطق و رابط ها¹، از

¹ Monolithic

هم جدا نیستند، بلکه با همدیگر ساخته شده اند (Zahavi 1999, Irani et al. 2003). بر خلاف سیستم های سفارشی، راه حل های بسته بندی شده^۲ یک مدل معماری سه لایه^۳ دارند که در آن، داده ها از منطق کسب و کار و رابط ها جدا شده اند، بنابراین به آسانی می توانند به روز شده یا تغییر یابند (Wijegunaratne Fernandez 1998, Irani et al. 2003). علاوه بر این، سیستم های بسته بندی شده مانند ERP بر اساس نیازمندی ها و فرایندهای عمومی کسب و کار بنا شده اند، نه بر مبنای نیازمندی های یک سازمان بخصوص (Holland Light Gibson 1999, Irani et al. 2003). علاوه بر این، Davenport گزارش می کند که سیستم های بسته بندی شده، اجازه سفارشی سازی زیاد را نمی دهند، بنابراین در اکثر اوقات، سازمان ها مجبورند فرایندهای کسب و کار و استراتژی خود را تغییر دهند تا خود را با سیستم بسته بندی شده انطباق دهند (Davenport 1998, Irani et al. 2003).

2-3-2- مولفه دوم: یکپارچه سازی برنامه کاربردی بین سازمانی

یکپارچه سازی بین سازمانی، به دنبال به هم پیوستن سیستم ها و فرایندهای کسب و کار بین سازمانی از طریق یک زنجیره تامین است (Irani et al. 2003). Kalkota and Robinson، پیشنهاد می کنند که راه حل های کسب و کار الکترونیک، بخشی از این زیر رده باشند (Kalkota and Robinson 1999, Irani et al. 2003). Linthicum توضیح می دهد که EAI، کسب و کار الکترونیک را از طریق همان دسته بندی از فناوری هایی (مانند دلال های پیغام، آداپتورها و XML) ممکن می سازد که از یکپارچه سازی بین سازمانی پشتیبانی می کنند (Irani et al. 2003). البته هنگام تعامل با سازمان های خارجی، خصیصه های دیگری نیز مورد نیاز است؛ به طور مثال مدیریت اجتماع^۴، مدیریت پروفایل همکار تجاری، ارسال اطلاعات از طریق نقاط خروجی استاندارد، مکانیزم های امنیتی پیچیده مانند دیواره آتش و پشتیبانی از استانداردهای صنعتی مورد توافق طرفین، مانند OBI^۵، XML^۶ و EDI^۷

¹ Interface
² Packaged solutions
³ Three-tier
⁴ Community management
⁵ Open Buying on the Internet
⁶ Extensible Markup Language
⁷ Electronic Data Interchange

(Gulledge 2006). برای حل این نیازها، چارچوب های B2B¹ تکامل یافته اند که در فصل سوم با

جزئیات بیشتری مورد بررسی قرار می گیرند.

بر اساس نوع یکپارچه سازی (ضعیف یا قوی بودن)، می توان یکپارچه سازی بین سازمانی را به دو دسته سازمان های توسعه یافته و سازمان های مجازی تقسیم کرد.

2-3-2-1- سازمان توسعه یافته

این دسته، نماینده برنامه های کاربردی کسب و کار الکترونیکی یکپارچه (مانند SCM) است، و در جاهایی کاربرد دارد که توسعه زیرساخت یکپارچه بین سازمانی متجانس، خیلی مد نظر نیست. در این حالت، سازمان ها فعالیت های کسب و کار خود را از طریق راه حل های کسب و کار الکترونیکی گسترش می دهند و تلاش دارند با همکاران خارجی خود، متحد و به هم پیوسته شوند.

2-3-2-2- سازمان مجازی

این زیر رده، به برنامه های کاربردی کسب و کار الکترونیک با یکپارچه سازی قوی² اشاره دارد که در آن، یکپارچه سازی تعدادی از سازمان ها که داده ها و فرایندهای عمومی خود را به اشتراک گذاشته اند، بسیار مهم است. در این حالت، تلاش می شود تا سازمان ها، به صورت یک سازمان (مجازی) کار کنند. در بسیاری از موارد، توجیه این رویکرد، پشتیبانی از فرایندهای مشترک به صورت کارا³ است، چون سازمان ها به اطلاعات بلادرنگ نیاز دارند (Irani et al. 2003).

2-3-3-3- مولفه سوم: یکپارچه سازی برنامه کاربردی پیوندی⁴

Helm اظهار کرده که راه حل های B2C⁵، هیچ چالشی برای یکپارچه سازی بین همکاران کسب و کار نیستند. اما چندین نویسنده اشاره کرده اند که در بعضی از حالات (مانند فروش الکترونیکی)، یکپارچه سازی برنامه های کاربردی B2C با سایر راه حل های بین سازمانی مورد نیاز است (مثلاً با

¹ B2B framework
² Tightly integrated
³ Efficiently
⁴ Hybrid application integration
⁵ Business to Customer

تامین کنندگان، توزیع کنندگان، بانک ها و غیره). سیستم های بین سازمانی، نقش مهمی در پشتیبانی از عملکرد یک برنامه کاربردی تجارت الکترونیک دارند، بنابراین برنامه های کاربردی تجارت الکترونیک، به یکپارچه سازی مناسب با سایر برنامه های کاربردی، که تعدادی از آنها ممکن است، راه حل های موروثی یا بسته بندی شده باشند، نیاز دارند.

کاربران اصلی برنامه های کاربردی B2C، شرکت های مالک این برنامه های کاربردی (ارائه دهندگان خدمات برنامه کاربردی¹ (ASP) و ارائه دهندگان فروشگاه²) و کاربران اینترنتی (مشتریانی) هستند که با این برنامه های کاربردی ارتباط دارند (Doukidis et al. 1998, Irani et al. 2003). در بعضی از برنامه های کاربردی (مانند خدمات دهی الکترونیکی³)، مشتریان یک بار به اشتراک سیستم در می آیند (با پرداخت الکترونیکی یا غیر الکترونیکی یک مقدار پول ثابت به یک بانک) و آنگاه از سیستم برای دوره ای مشخص (به طور مثال 1 سال) استفاده می کنند. در طول این دوره، مالک برنامه کاربردی B2C، بدون نیاز به یک موجودیت خارجی (مانند تامین کننده)، به مشتری خدمات ارائه می دهد. در نتیجه، شاید نیازی به یکپارچه کردن این نوع از سیستم ها با شرکت های همکاران خارجی نباشد، چون در واقع هیچ شرکت خارجی وجود ندارد. اما انواع دیگر برنامه های کاربردی B2C، مشابه سازمان های توسعه یافته یا مجازی کار می کنند. برای نمونه، بسیاری از برنامه های کاربردی فروش الکترونیکی به یکپارچه سازی بین سازمانی نیاز دارند تا سیستم های بانک ها، تامین کنندگان و توزیع کنندگان را با هم پیوند دهند.

بر اساس این تفکر، یک زیر رده جدید با عنوان یکپارچه سازی برنامه کاربردی پیوندی خواهیم داشت که شامل برنامه های کاربردی B2C و هم سطح با یکپارچه سازی درون سازمانی و بین سازمانی است. در جدول 2-2، مشخصات این سه سطح، به صورت خلاصه گرد آورده شده است.

Application Service Provider¹
Shop-provider²
E-services³

جدول 2-2- مشخصات زیر رده های دسته بندی بر مبنای طرف های درگیر در EAI

طبقه بندی	مشخصات	مرجع
EAI		
درون سازمانی	برنامه های کاربردی سازمانی را یکپارچه می کند	Brown et al. 2000, Loinsky 1995
	سیستم های بسته بندی شده (مانند ERP) و سفارشی را یکپارچه می کند	Edwards Newing 2000, Ruh et al 2000
	هیچ تراکنشی با کاربران یا همکاران خارجی ندارد	Helm 1999, Themistocleous and Irani 2001
پیوندی	برنامه های کاربردی B2C را با زیر ساخت IT یکپارچه می کند	Themistocleous and Irani 2002
	کاربران اینترنتی، محصولات یا خدمات را سفارش می دهند. یکپارچه سازی برنامه های کاربردی پیوندی ¹	Doukidis et al. 1998, Kalakota and Robinson 1999
بین سازمانی	پشتیبانی از تراکنش ها ² با یکپارچه سازی سیستم های داخلی با همکاران خارجی برنامه های کاربردی بین سازمانی را با زیر ساخت IT یکپارچه می کند	Linthicum 2000, Zahavi 1999 Markus and Tanis 1999, Morgenthal and Forge 2000
	باید مسائلی مانند مدیریت اجتماع ³ ، مدیریت پروفایل همکار تجاری، ارسال اطلاعات از طریق نقاط خروجی استاندارد، مکانیزم های امنیتی پیچیده و استفاده از استانداردهای صنعتی مورد توافق طرفین نیز در نظر گرفته شود	Gulledge 2006
	بر اساس درجه یکپارچه سازی (ضعیف یا قوی بودن یکپارچه سازی)، دو نوع سازمان خواهیم داشت سازمان های توسعه یافته (یکپارچه سازی ضعیف) سازمان های مجازی (یکپارچه سازی قوی)	Helm 1999, Puschmann and Alt 2001

¹ Hybrid Application Integration
² Transactions
³ Community management

2-4- دسته بندی بر اساس انواع سیستم های نرم افزاری که ممکن است با هم یکپارچه شوند

برنامه های کاربردی منبع و مقصد، می توانند جزء یکی از سیستم های بسته بندی شده¹ (مانند ERP)، سیستم های سفارشی² (مانند سیستم های موروثی) و کسب و کار الکترونیک (مانند فروشگاه - الکترونیک³) باشند. بر اساس این سه دسته از سیستم ها، جایگشتی از انواع سیستم هایی که ممکن است با هم یکپارچه شوند، وجود دارد که دسته بندی آن ها به طور خلاصه در جدول 2-3 آمده است (Themistocleous et al. 2004).

جدول 2-3- دسته بندی انواع سیستم های نرم افزاری که ممکن است با هم یکپارچه شوند

دسته بندی انواع سیستم ها	توصیف
یکپارچه سازی سفارشی با سفارشی	چون سیستم های سفارشی، برای کار کردن با سایر سیستم ها، توسعه داده نشده اند، نقاط دسترسی محدودی برای یکپارچه سازی دارند. در اکثر برنامه های کاربردی سفارشی، پایگاه داده ها و رابط های کاربری، تنها نقاط دسترسی ممکن، برای یکپارچه سازی هستند. برای پشتیبانی از این دسته، فناوریهایی که داده ها را از پایگاه داده استخراج می کنند (مانند رونوشت از پایگاه داده ⁴) یا فناوری هایی که داده ها را از روی صفحه استخراج می کنند (مانند خراشنده صفحه ⁵)، مورد نیاز است.
یکپارچه سازی سفارشی با بسته بندی شده	چون برنامه های کاربردی بسته بندی شده، مانند سیستم های ERP، در بسیاری از مواقع به یکپارچه سازی نائل نمی شوند، برنامه های کاربردی سفارشی نیز به همراه آنها در سازمان موجود است، که بایستی این دو سیستم با هم یکپارچه شوند. از دیدگاه فنی، این رویکرد می تواند در سطوح متعددی مانند داده ها، اشیاء/مولفه ها با در سطح رابط پیاده شود
یکپارچه سازی کسب و کار الکترونیک با سفارشی	بسیاری از راه حل های کسب و کار الکترونیک (مانند خدمات دهی الکترونیکی، فروش الکترونیکی)، برای پشتیبانی از فرایندها و وظایفی که توسط کسب و کار الکترونیک امکان پذیر شده، نیاز به همکاری نزدیکی با برنامه های کاربردی موروثی (به طور مثال برنامه های سهام) دارند.

¹ Packaged

² Legacy

³ E-store

⁴ DB Replication

⁵ Screen scraper

<p>همچنین در بسیاری از مواقع، برای پشتیبانی از سیستم های سفارشی از عملکرد راه حل های کسب و کار الکترونیک استفاده می شود. برای پوشش این دسته به فناوری هایی نیاز است که از پیوستگی و تبادل داده ها، اشیاء و رابط ها پشتیبانی کند.</p>	
<p>در این موقعیت، سیستم های بسته بندی شده نامتجانس، مانند نسخه های متفاوت از یک سیستم ERP یا ماژول هایی از ERP های ارائه دهندگان متفاوت (مثلاً SAP و Oracle) که در یک سازمان وجود دارند، درون یک زیرساخت یکپارچه مشترک، با هم پیوند می خورند. سیستم های ERP، API هایی دارند که اجازه می دهد، سایر برنامه های کاربردی، به عملکرد یا داده های آن ERP دسترسی داشته باشند. داده ها، پیغام ها یا اشیاء می توانند با استفاده از این API ها به یک سیستم ERP، وارد یا از آن صادر شوند. بنابراین، این رویکرد، علاوه بر API ها، به فناوری هایی نیاز دارد که از استخراج و انتقال داده ها، پیغام ها و اشیاء پشتیبانی کند</p>	<p>یکپارچه سازی بسته بندی شده با بسته بندی شده</p>
<p>در این مورد، فرایندهایی که با فروش الکترونیک، تدارکات الکترونیک و مدیریت الکترونیکی زنجیره تامین (E-SCM) سروکار دارند، با سیستم های بسته بندی شده (مانند ERP) یکپارچه می شوند. برنامه های کاربردی کسب و کار الکترونیک، اغلب بر پایه فناوری های اشیاء توزیع شده (به طور مثال EJB، CORBA، COM/DCOM) و یا زبان ها/استانداردهای اینترنت گرا (مانند XML، HTML) بنا شده اند. بنابراین، فناوری هایی که داده ها، اشیاء، رابط ها و پیغام ها را در کنار همدیگر دارند، برای این مورد مهم است.</p>	<p>یکپارچه سازی بسته بندی شده با کسب و کار الکترونیک</p>
<p>در این رویکرد، یک برنامه کاربردی کسب و کار الکترونیک، با راه حل کسب و کار الکترونیک دیگر یکپارچه شده و از عملکرد دیگری پشتیبانی می کند. برای مثال، یک پایانه فروش (POS) الکترونیکی¹، با E-SCM پیوند می خورد، تا داده هایی را که برای برنامه دیگر مهم است، به اشتراک بگذارد (به طور مثال، سفارشات مشتری، اطلاعات جزئی مشتری و غیره).</p>	<p>یکپارچه سازی کسب و کار الکترونیک با کسب و کار الکترونیک</p>

¹ Electronic Point Of Sale

<p>فناوری های مبتنی بر پیغام (مانند XML)، فناوری های اشیاء توزیع شده (مانند CORBA) و فناوری های پایگاه داده گرا (JDC¹، JDBC²)، این نوع از یکپارچه سازی را امکان پذیر می کنند.</p>	
<p>چنین رویکردی، روی توسعه یک زیرساخت یکپارچه تاکید دارد که فرایندها و برنامه های کاربردی را در سطح دیارتمان، سازمان یا بین سازمانی، یکپارچه کند. چون سیستم های متنوعی با هم پیوند می خورند، نیاز به فناوری های یکپارچه سازی دارد که از تمام سطوح سازمانی پشتیبانی کند. بنابراین، فناوری هایی در سطح داده، شی، رابط و پیغام مورد نیاز است.</p>	<p>یکپارچه سازی سفارشی با بسته بندی شده با کسب و کار الکترونیک</p>

2-5- انواع رویکردهای یکپارچه سازی بر اساس سطح مطلوب برای یکپارچه سازی

بر مبنای سطح مطلوب برای EAI، می توان از چندین روش استفاده کرد (شکل 2-2). تعدادی از اینها، در زیر شرح داده شده است (Lutz 2000, Paulson 2000, Erasala et al. 2003).

2-5-1- خراش صفحه³

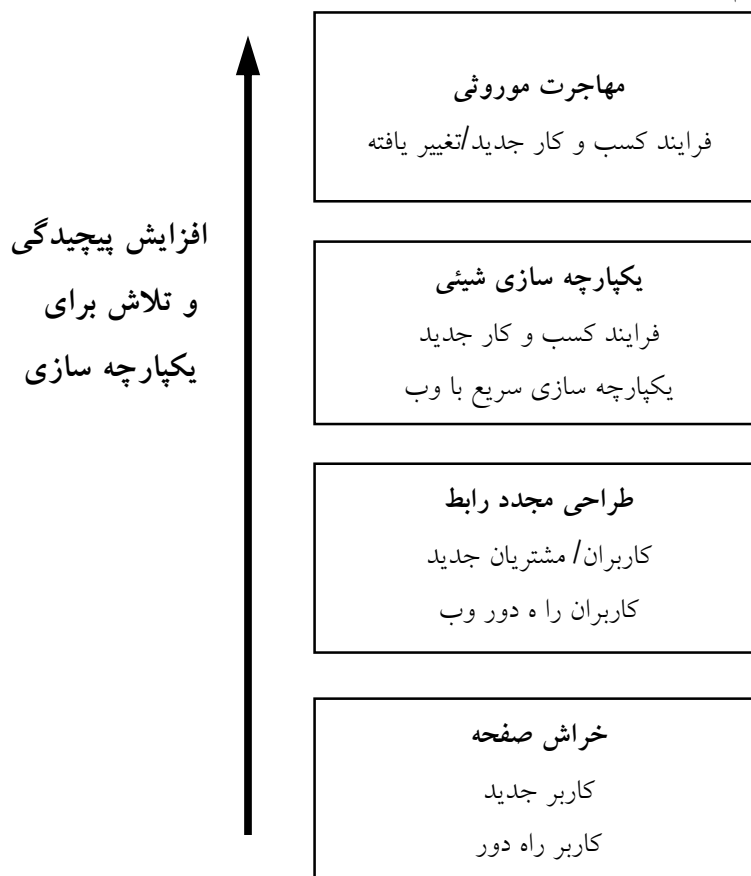
این روش، از نرم افزار تقلید⁴ برای نگاشت صفحات نشست ترمینال⁵ یک برنامه کاربردی مین فریم استفاده کرده و آن را درون یک فناوری رابط کاربری مناسب مانند HTML قرار می دهد. مزیت این رویکرد، بازنمایی صفحه سبز⁶ سیستم های موروثی در قالب یک فناوری رابط کاربری با ظاهر بهتر و همچنین فرصتی برای مهندسی مجدد رابط کاربری است.

2-5-2- طراحی مجدد رابط⁷

این راهکار، پیچیدگی و تهاجم بسیار بیشتری دارد و درگیر کپسوله کردن کد موروثی با یک API

¹ Java Developer Connection
² Java Data Base Connectivity
³ Screen scraping
⁴ Emulation
⁵ Terminal session screen
⁶ Green screen
⁷ Interface redesign

(نقطه ورود¹) جدید است که می توان به این API، وظایف دیگری نیز اضافه کرد، که این ممکن است، سبب تغییر چند فرایند کاربر و سیستم شود. اگرچه این روش می تواند گران باشد، ولی اجازه خواهد داد که از منطق اولیه کسب و کار مجدداً استفاده شده و یکپارچه سازی بهتری در سطح رویه ای (تابعی) فراهم شود.



شکل 2-2- رویکردهای یکپارچه سازی سیستم ها بر مبنای سطح مطلوب برای یکپارچه سازی

2-5-3- یکپارچه سازی شیئی²

در این روش، کد موروثی با یک رابط کاربری که عملکرد برنامه کاربردی³ را کپسوله می کند، پوشیده می شود.⁴ لفافه⁵، ورودی و خروجی های برنامه کاربردی را از طریق متدهایش ترجمه می کند. بعد از این، نرم افزار EAI، این امکان را فراهم می کند تا این متدها توسط سایر برنامه های

Entry point¹
Object integration²
Application's functionality³
Wrap⁴
Wrapper⁵

کاربردی به کار گرفته شوند و تمام جزئیات برنامه کاربردی اصلی مخفی شود. یک گزینه بهتر اما پرهزینه تر، ترجمه کد موروثی به زبان های مبتنی بر مولفه مانند COM¹، CORBA² یا EJB³ است.

2-5-4- مهاجرت موروثی⁴

این رویکرد، درگیر جذب منطق کسب و کار⁵ برنامه کاربردی موروثی و طراحی مجدد آن، اضافه کردن عملکرد به آن یا بکارگیری آن روی یک پلتفرم جدید است. این امر در مواردی انجام می شود که سیستم موروثی برای فرایند کسب و کار حیاتی بوده و همچنین یک جایگزین آماده سفارش⁶ برای آن وجود ندارد. در این راهکار، از فنون تحلیلی موروثی مانند مدیریت دارایی های نرم افزاری⁷ (SAM) یا دانایی کاوی⁸ به طور وسیع استفاده می شود، چون آنها به طور قابل توجهی، ریسک و هزینه را کاهش می دهند. در مجموع، فرایند تبدیل کدهای منبع سیستم موروثی به زبان ها و فناوری های نوین، کار را از جنبه هزینه و سرعت بادوام و ماندنی می کند (Erasala et al. 2003).
Mosawi et al. (2006)، انواع رویکردهای یکپارچه سازی برنامه های کاربردی سازمانی را به دو دسته اصلی تقسیم بندی می کند. دسته بندی اول، بر اساس عناصر سیستم های درگیر در یکپارچه سازی است و دسته بندی دوم، بر اساس لایه های معماری مورد نیاز در یکپارچه سازی است.

2-6- انواع رویکردهای EAI بر اساس عناصر سیستم های درگیر در یکپارچه سازی

دسته بندی اول Mosawi et al. (2006)، شامل رویکردهایی از EAI است که مشکل یکپارچه سازی را برای یک بخش خاص از سیستم، مانند داده یا فرایند برطرف می کنند. برای مثال، Linthicum چهار رویکرد یکپارچه سازی را پیشنهاد کرده است که عبارتند از: یکپارچه سازی داده ای، یکپارچه

Component Object Model 1
Common Object Request Broker Architecture 2
Enterprise Java Bean 3
Legacy migration⁴
Capturing business logic⁵
Off-the-shelf 6
Software Asset Management⁷
Knowledge mining⁸

سازی رابط برنامه کاربردی¹، یکپارچه سازی متد و یکپارچه سازی رابط کاربری² (Linthicum 1999, Mosawi et al. 2006)؛ Pushmann و Alt نیز، سه رویکرد یکپارچه سازی داده ای، شیئی و فرایندی پیشنهاد کرده اند (Pushmann and Alt 2001, Mosawi et al. 2006).

بسیاری از اوقات، تعدادی از این رویکردها، می توانند در یک راه حل EAI ترکیب شوند. برای مثال، یکپارچه سازی سیستم های پایگاه داده شی گرا شامل یکپارچه سازی داده ها، اشیا و فرایندها است. Mosawi et al. 9 رویکرد یکپارچه سازی را در این طبقه بندی شناسایی کرده اند که در جدول 2-4 آمده است (Mosawi et al. 2006).

یکپارچه سازی داده ای یا اطلاعاتی را می توان مهمترین رویکرد یکپارچه سازی متعارف دانست. در این روش، داده ها بین چند منبع داده ای مهاجرت می کنند و این امکان فراهم می شود تا داده ها بین چند سازمان، برنامه کاربردی و منبع³، مبادله شده و به اشتراک گذاشته شوند (Linthicum 1999, Ruh et al. 2000, Samtani et al. 2002, Mosawi et al. 2006).

یکپارچه سازی شیئی روی سیستم هایی اعمال می شود که با استفاده از فناوری شی گرا توسعه داده شده اند. این رویکرد اجازه می دهد که اشیاء کسب و کار، با ایجاد رابط های قابل استفاده مجدد⁴، با همدیگر کار کنند و عملکرد⁵ خود را بین تمام برنامه های کاربردی، به اشتراک بگذارند (Ring and Ward-Dutton 1999, Mosawi et al. 2006).

یکپارچه سازی متد یا تابع⁶ مشکل یکپارچه سازی را در سطح متد یا تابع حل می کند. مبنای آن، اشتراک منطق کسب و کاری است که درون سازمان موجود است (Linthicum 1999, Ruh et al. 2000, Samtani et al. 2002, Mosawi et al. 2006).

یکپارچه سازی رابط کاربری بدوی ترین و قدیمی ترین شکل از یکپارچه سازی است. مبنای آن، توسعه یک رابط کاربری است که عملیات کاربر نهایی را با استفاده از خراش صفحه یا تقلید ترمینال پیشرفته⁷، تقلید کند¹ (Linthicum 1999, Samtani et al. 2002, Mosawi et al. 2006).

1 Application Interface Integration
2 User interface integration
3 Resource
4 Reusable interface
5 Functionality
6 Function
7 Advanced terminal emulation

جدول 2-4- انواع رویکردهای EAI بر اساس عناصر سیستم های درگیر در یکپارچه سازی

نویسنده	رویکرد یکپارچه سازی
Linthicum 1999, Pushmann and Alt 2001, Ring and Ward-Dutton 1999, Samtani et al. 2002, Ruh et al. 2000	داده یا اطلاعات
Pushmann and Alt 2001, Ring and Ward-Dutton 1999	شی
Linthicum 1999, Ruh et al. 2000, Samtani et al. 2002	تابع/متد
Linthicum 1999, Samtani et al. 2002	رابط کاربری
Linthicum 1999	رابط برنامه کاربردی
Ruh et al. 2000	ارائه
Pushmann and Alt 2001, Samtani et al. 2002	فرایند
Ring and Ward-Dutton 1999	فرایند داخلی
Ring and Ward-Dutton 1999	فرایند بین سازمانی

یکپارچه سازی رابط های برنامه کاربردی، مشکل تعامل بین برنامه های کاربردی را از طریق اشتراک منطق کسب و کار مشترک² حل می کند که این منطق کسب و کار با استفاده از یک رابط برنامه نویسی از پیش تعریف شده، در معرض نمایش گذاشته شده است. مبنای این روش، در معرض نمایش گذاشتن³ یک رابط (API) از برنامه های کاربردی بسته بندی شده یا سفارشی است، تا به این وسیله، سرویس های کسب و کار این برنامه ها را فراخوانی کنند یا داده های آنها را بازیابی کنند (Linthicum 1999, Mosawi et al. 2006).

یکپارچه سازی ارائه⁴ ساده ترین شکل از یکپارچه سازی است. این روش، با نگاشت مجدد ارائه های قدیمی، به شما امکان ایجاد رابط کاربری جدید را می دهد (Ruh et al. 2000, Mosawi et al. 2006). این رویکرد، متناظر با یکپارچه سازی رابط کاربری است (Mosawi et al. 2006). یکپارچه سازی فرایندی با فرایندها به عنوان انتزاع های مشترک¹ از سیستم ها رفتار می کند. این روش نسبت به اشیاء و متدها، در سطحی بالاتر کار می کند. این رویکرد، توسط Samtani et al.

¹ Emulate
² Common business logic
³ Expose
⁴ Presentation integration

متدهای کسب و کار نامیده شده است. این روش، درگیر گردش اطلاعات و اتوماسیون فرایندهای کسب و کار، بر اساس قوانین کسب و کار² سیستم های کسب و کار مختلف است (Samtani et al. 2002). این رویکرد، هماهنگی تصمیمات و مدیریت وابستگی ها را بین فرایندهای سازمانی امکان پذیر می سازد (Mosawi et al. 2006).

یکپارچه سازی فرایند داخلی³ برای مدیریت فرایندهای داخلی سازمان ها به کار گرفته می شود. این روش، اجازه ردیابی وضعیت فعالیت های فرایندهای داخلی کسب و کار را می دهد (Ring and Ward-Dutton 1999, Mosawi et al. 2006).

یکپارچه سازی فرایند بین سازمانی⁴، بیشتر برای یکپارچه سازی برنامه کاربردی B2B مناسب است. این روش، اجازه می دهد فرایندهای B2B، بین چند نهاد سازمانی به اشتراک گذاشته شوند (Ring and Ward-Dutton 1999, Mosawi et al. 2006).

2-6-1- نقاط قوت و ضعف این رویکردها

نقاط قوت و ضعف رویکردهایی که در بالا مطرح شد، به صورت خلاصه در جدول 2-5 گردآوری شده است.

یکپارچه سازی داده ای رویکردی ساده و ارزان برای پیاده سازی است (Linthicum 1999, Mosawi et al. 2006). این رویکرد، از یک نمای متحدالشکل داده ای، برای تمام واحدهای سازمان و همکاران تجاری آن پشتیبانی می کند. داده ها می توانند بدون نیاز به تغییر روی سیستم مبدا، مقصد، یا پایگاه داده، بین حوزه های کسب و کار مختلف منتقل شده و همچنین بدون توجه به نوع سیستم های پایگاه داده، سیستم های عامل یا پلتفرم شبکه به اشتراک گذاشته شوند. این رویکرد، دسترسی یک بازه وسیعی از داده ها را فراهم کرده و اجازه می دهد داده ها، توسط سایر برنامه های کاربردی مجدداً استفاده شوند. اما این رویکرد، منطق کسب و کار را دور می زند، بنابراین، قابلیت های تراکنشی بلادرنگ را محدود می کند (Samtani et al. 2002, Mosawi et al. 2006). در ضمن،

¹ Common abstractions

² Business rules

³ Internal process integration

⁴ Cross-enterprise process integration

عدم فراخوانی منطق کسب و کار، به آسانی سبب بروز مشکلات جامعیتی¹ می شود، بنابراین در پیاده سازی EAI در سطح یک سازمان، باید تا جای ممکن، از این شکل از یکپارچه سازی اجتناب شود (Gulledge 2006).

جدول 2-5- مروری بر نقاط قوت و ضعف رویکردهای مختلف EAI

نقاط ضعف	نقاط قوت	رویکرد یکپارچه سازی
منطق کسب و کار را فراخوانی نمی کند شاید سبب ایجاد مشکلات جامعیت داده ای شود	پیاده سازی آن ساده است پیاده سازی آن ارزان است داده ها سازگار هستند به حداقل تغییرات روی برنامه کاربردی مبدا و مقصد نیاز دارد	داده یا اطلاعات داده ها را بین چند منبع داده ای، به اشتراک می گذارد
پیچیده است رویکردی سخت و مشکل است	منطق کسب و کار را نیز یکپارچه می کند از اشیاء می توان مجدداً استفاده کرد	شی اشیاء توزیع شده را در سراسر سازمان یکپارچه می کند
معماری آن پیچیده است پیاده سازی آن گران است	ردیابی و تحلیل بلادرنگ فرایند کسب و کار بهینه سازی و تنظیم فرایند به صورت پویا امکان پذیر است ارزیابی فرایند امکان پذیر است	فرایند مدل سازی و یکپارچه سازی فرایند کسب و کار
پیوند قوی بین برنامه های کاربردی ارتباطات آن همزمان است نگهداری آن مشکل است مقیاس پذیر نیست اتصال واقعی بین برنامه ها رخ نمی دهد	پیاده سازی آن آسان است نیاز به حداقل تغییرات روی برنامه کاربردی موجود دارد	رابط کاربری ساخت رابط های جدید با نگاشت مجدد رابط های قدیمی
پیوند قوی بین برنامه های کاربردی ارتباطات آن همزمان است هر دو برنامه کاربردی، نیاز به API یکسانی دارند	عملکرد برنامه کاربردی را فراخوانی می کند	رابط برنامه کاربردی یک رابط را برای اشتراک عملکرد کسب و کار در معرض نمایش می گذارد

¹ Integrity Issues

یکپارچه سازی شیئی منطق کسب و کار و داده ها را درون اشیاء، کپسوله می کند¹ و اجازه می دهد که این اشیاء برای تعامل با یکدیگر، به یک حالت Plug and play، با همدیگر پیوند داده شوند. این رویکرد، ارزش هنگفتی برای سازمان هایی با سیستم و معماری نامتجانس دارد. این روش، اجازه می دهد که سیستم موروثی در قالب اشیاء پوشیده شود و در یکپارچه سازی شیئی، شریک شود. البته این رویکرد پیچیده است و مشکل ترین رویکرد یکپارچه سازی است (Linthicum 1999, Mosawi et al. 2006).

یکپارچه سازی فرایندی پیشرفته تر از یکپارچه سازی داده ای و شیئی است، چون منطق و قوانین هدایت کسب و کار را در نظر می گیرد. این روش، محدودیت یکپارچه سازی داده ای را با تمرکز روی فرایند حقیقی و نه فقط روی داده، از بین می برد. با این که یکپارچه سازی فرایندی، منافع هنگفتی برای کسب و کار ارائه می کند، اما پیچیده بوده و گران ترین رویکرد در پیاده سازی است. این روش، به سرمایه گذاری زیادی نیاز دارد و معماری و توسعه آن پیچیده است (Samtani et al. 2006, Mosawi et al. 2002).

یکپارچه سازی رابط کاربری، ساده ترین و سریع ترین روش یکپارچه سازی است. اما با این وجود محدودیت های بسیاری دارد. سبک ارتباطات این رویکرد، همزمان² است و لازم است که هر دو برنامه کاربردی درگیر در یکپارچه سازی، همزمان آنلاین باشد. این روش مقیاس پذیر³ نیست، بنابراین در یک زمان مشخص، نمی تواند بیش از چند رابط صفحه⁴ را مدیریت کند (Linthicum 2006, Mosawi et al. 1999). در این رویکرد، اتصال واقعی بین برنامه های کاربردی و داده ها رخ نمی دهد، بلکه یکپارچه سازی فقط در لایه ارائه رخ می دهد (Ruh et al. 2000, Mosawi et al. 2006). یکپارچه سازی مبتنی بر پورتال را می توان جزء این دسته از یکپارچه سازی قرار داد.

یکپارچه سازی رابط های برنامه کاربردی اجازه می دهد، برنامه های کاربردی خارجی، بدون تغییر برنامه های کاربردی موجود، به سرویس های برنامه های کاربردی موجود، دسترسی داشته باشند. البته این رویکرد محدودیت های بسیاری دارد. ماهیت این رویکرد، همگام است، یعنی در هنگام

¹ Encapsulate
² Sysnchronous
³ Scalable
⁴ Screen interface

فراخوانی تابع، تا زمانی که عمل بخصوصی کامل نشود، چیزی برگشت داده نمی شود. بنابراین بایستی هر دو برنامه کاربردی همزمان موجود باشند، که این ممکن است منجر به ایجاد یک سیستم با پیوند قوی¹ شود. هیچ استاندارد صنعتی برای ایجاد API وجود ندارد و برای امکان پذیر کردن رویکرد یکپارچه سازی مبتنی بر API، هر دو برنامه کاربردی بایستی از زبان یا پروتکل یکسانی استفاده کنند. (Samtani et al. 2002, Mosawi et al. 2006).

7-2- انواع رویکردهای EAI بر اساس لایه های معماری مورد نیاز در یکپارچه سازی

طبقه بندی دومی که Mosawi و دیگران (2006) از EAI ارائه داده، شامل رویکردهایی از EAI است که نگاه معمارانه به یکپارچه سازی داشته و مسئله یکپارچه سازی را از جوانب متعددی بررسی می کند. رویکردهای EAI در این طبقه بندی، مشکل یکپارچه سازی را در لایه های معماری مختلفی سازماندهی می کنند. برای مثال Duke et al. (1999)، مسئله یکپارچه سازی را به سه لایه معماری متفاوت، شامل لایه معماری کسب و کار، لایه معماری برنامه کاربردی و لایه معماری فناوری، تقسیم می کند (Duke et al 1999, Mosawi et al. 2006). همانطور که در جدول 2-6 مشاهده می شود، Mosawi و دیگران، 8 لایه را در این طبقه بندی شناسایی کرده اند.

لایه معماری کسب و کار² ساختار سازمانی و گردش کار³ را برای مفاهیم، فرایندها و قوانین کسب و کار تعریف می کند. Brown این لایه را به دو لایه فرایند کسب و کار و معماری اطلاعات تقسیم می کند که این دو لایه در زیر تشریح شده اند (Brown 2000, Mosawi et al. 2006).

لایه فرایند کسب و کار، فرایندهای مرکزی را که هر کسب و کار، برای هدایت آن کسب و کار انجام می دهد، تعریف می کند.

لایه معماری اطلاعات مناطق موضوعی⁴ مورد توجه کسب و کار را تعریف کرده و انواع داده های لازم برای سرهم بندی اطلاعات¹ در هر منطقه موضوعی را تعریف می کند.

Tightly coupled 1
Business architecture layer 2
Workflow 3
Subject areas 4

Hasselbring این معماری را با اضافه کردن لایه بین سازمانی، برای به هم پیوستن افقی فرایندهای کسب و کار واحدهایی که با همدیگر کار می کنند، گسترش داد.

جدول 2-6- انواع رویکردهای EAI بر اساس لایه های معماری مورد نیاز در یکپارچه سازی

نویسنده	لایه یکپارچه سازی
Duke et al. 1999, Hasselbring 2000	معماری کسب و کار
Brown 2000	فرایند کسب و کار
Brown 2000	معماری اطلاعات
Hasselbring 2000	فرایندهای بین سازمانی
Duke et al. 1999, Brown 2000, Hasselbring 2000	معماری برنامه کاربردی
Hasselbring 2000	یکپارچه سازی برنامه های کاربردی سازمانی
Duke et al 1999, Brown 2000, Hasselbring 2000	معماری فناوری
Hasselbring 2000	یکپارچه سازی میان افزار

لایه فرایندهای بین سازمانی² با برش افقی فرایندهای کسب و کار از بین ساختار وظیفه ای و سنتی سازمان، به دنبال سازماندهی فرایندهای کسب و کار به یک شکل قابل رقابت است (Hasselbring 2000, Mosawi et al. 2006).

لایه معماری برنامه کاربردی³ استراتژی های پیاده سازی لایه معماری کسب و کار را تعریف می کند. Hasselbring این معماری را با اضافه کردن لایه یکپارچه سازی برنامه کاربردی سازمانی برای یکپارچه سازی افقی، گسترش می دهد.

لایه یکپارچه سازی برنامه کاربردی سازمانی⁴ به دنبال یکپارچه سازی سیستم های مستقل ERP است.

¹ Information assembling
² Inter-organizational processes layer
³ Application architecture layer
⁴ Enterprise Application

لایه معماری فناوری زیرساخت مورد نیاز برای گردش اطلاعاتی و ارتباطی کسب و کار را تعریف می کند (Mosawi et al. 2006).

Hasselbring این معماری را، با اضافه کردن لایه یکپارچه سازی میان افزار، برای یکپارچه سازی افقی بسط می دهد، تا بدین ترتیب، سیستم های اطلاعاتی با معماری مبتنی بر مولفه¹ را با استفاده از فناوری های بسیار جدیدی مانند CORBA و دروازه پایگاه داده² یکپارچه کند (Hasselbring 2000,) (Mosawi et al. 2006).

2-8- دسته بندی بر مبنای یک معماری EAI مدل محور³

Mosawi و دیگران (2006) بر اساس معماری مدل محور⁴ OMG، یک معماری مدل محور برای EAI ارائه داده اند. هدف این مدل، جداسازی وظایف و رفتار کسب و کار از قوانین مختص فناوری است تا هر دوی اینها بتوانند به صورت مستقل از هم توسعه یابند. جداسازی وظایف و رفتار کسب و کار از دستورات عملی های تکنولوژیک سبب می شود، انعطاف و قابلیت تعامل با یکدیگر⁵ که EAI به آن نیاز دارد، فراهم شود. معماری پیشنهادی، حوزه EAI را در قالب پنج نوع مدل کلی در نظر می گیرد. همانطور که در شکل 2-3 نشان داده شده، این پنج نوع مدل، شامل مدل مختص فناوری⁶، مدل سرویس های تراکنشی⁷، مدل سرویس های برنامه کاربردی⁸، مدل برنامه کاربردی درونی⁹ و مدل برنامه کاربردی بینی¹⁰ است. این معماری اجازه می دهد، حوزه های کسب و کار از طریق مجموعه ای از سرویس های مشترک تعامل و همیاری داشته باشند. هم حوزه های کسب و کار و هم سرویس های عمومی، به تسهیلات تکنولوژیک تکیه دارند که اتصال فیزیکی و تعامل را برای مدل امکان پذیر می کند (Mosawi et al. 2006).

-
- 1 Componentized information systems
 - 2 Database gateway
 - 3 Model Driven EAI architecture
 - 4 OMG's Model Driven Architecture
 - 5 Interoperability
 - 6 Technology Specific model
 - 7 Transaction Service model
 - 8 Application Services model
 - 9 Intra-Application model
 - 10 Inter-Application model



- A. Technology Specific Model
- B. Transaction Service Model
- C. Application Services Model
- D. Intra-Application Model
- E. Inter-Application Model

شکل 2-3- یک معماری EAI مدل محور

2-8-1- مدل برنامه کاربردی بینی

وظیفه این مدل، تعریف گردش اطلاعاتی و ارتباطی موجود بین سازمان و سایر همکاران تجاری اش مانند مشتری، تامین کننده و خرده فروش است. این مدل، نیازهای شرکت های همکاری را که سازمان با آنها کار خواهد کرد، تعیین کرده و روی این تمرکز دارد که همکاران تجاری، به چه نحوی فرایند کسب و کار خود را تصحیح کنند تا برنامه های کاربردی که از تعاملات تجاری بین این همکاران پشتیبانی می کنند، بتوانند با همدیگر تعامل داشته باشند.

2-8-2- مدل برنامه کاربردی درونی

این مدل، ساختار سازمانی و فرایندهایی را تعریف می کند که سازمان برای هدایت کسب و کار انجام می دهد. این مدل، وظایف و گردش کاری¹ را تعریف می کند که بایستی توسط مدل سرویس برنامه کاربردی برآورده شود.

¹ Workflow

2-8-3- مدل سرویس های برنامه کاربردی

این مدل، سرویس های لازم را برای اجرای اتوماسیون فرایندهای کسب و کار تعریف می کند. این سرویس ها از مدیریت وظایف و عملیات فرایندهای کسب و کار پشتیبانی می کنند، اما جزئیات مورد نیاز یک پلتفرم خاص را مخفی می کنند. این مدل از دو نوع سرویس تشکیل شده که در زیر مطرح شده است.

سرویس های اختصاصی سرویس کسب و کار و منطق کسب و کار را برای یک حوزه عمودی خاص مانند برنامه های کاربردی صنعتی، مالی یا دولتی فراهم می کند. **سرویس های کلی** سرویس های کلی را که برای چندین حوزه عمودی مانند برآورد سفارش، حمل و نقل یا خرید مشترک هستند، ارائه می دهد.

بنابراین این مدل، هم از کارکردهای یک حوزه کسب و کار معین و هم از تعریف و مدیریت اطلاعات و فعالیت های مربوط به چندین حوزه کسب و کار پشتیبانی می کند. سرویس های کسب و کار، بر اساس یک شیوه قابل پیش بینی با هم تعامل داشته و به هم وابسته اند تا یک فعالیت کسب و کار قابل شناسایی و با ماهیتی کلی برای یک حوزه کسب و کار بخصوص ارائه دهند.

2-8-4- مدل سرویس های تراکنشی

وظیفه این مدل، تعریف پشتیبانی تراکنشی از سرویس های برنامه کاربردی مانند انتقال وجوه، صدور فاکتور، صورت حساب و پرداخت است. این مدل، توافقات قراردادها^۱، استثناها، احتمالات^۲، پرداخت وجوه^۳ و نقض قرارداد^۴ را تعریف می کند.

2-8-5- مدل مختص فناوری

این مدل، پلتفرم تکنولوژیک بنیادی را برای اتصال فیزیکی و تعامل با سیستم برنامه کاربردی تعریف

¹ Contractual obligations

Contingencies 2

Remittances 3

Contract violations 4

می کند. این مدل، زیر ساخت ارتباطی، فناوری های ارتباطی، توپولوژی شبکه، حالت های اتصال¹ و نیازمندی های توزیعی را تعیین می کند (Mosawi et al. 2006).

2-9- دسته بندی بر مبنای یکپارچه سازی افقی معماری مدل محور

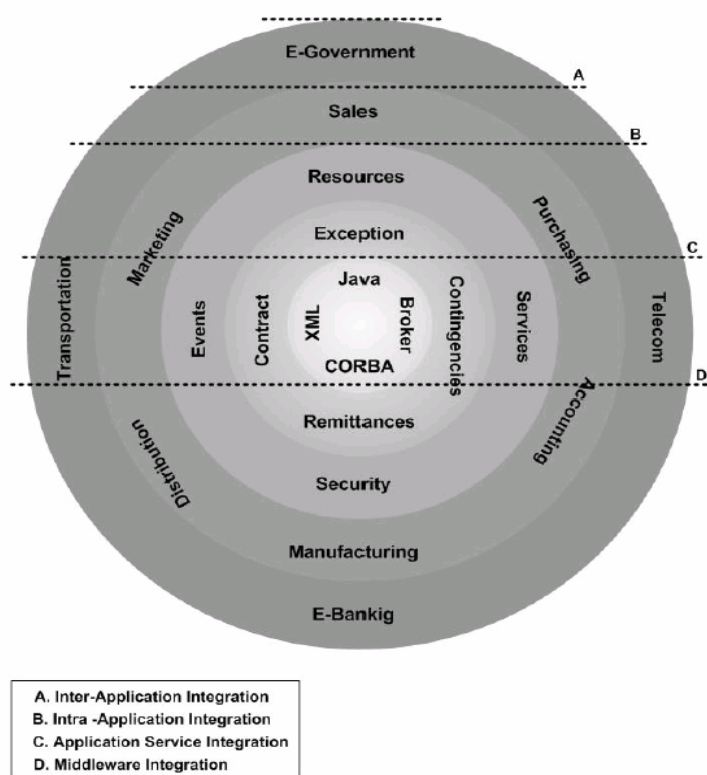
اگرچه معماری مدل محور بالا بسیار ضروری و با ارزش است، اما باید دانست که ممکن است در این معماری، فرایندهای کسب و کار واحدهایی که با هم تعامل دارند، به هم وابسته نباشند. هنگامی که برای پشتیبانی از فرایندهای کسب و کاری که در میان چند واحد کسب و کار قرار گرفته اند، لازم باشد تا فرایندی، واحدهای وظیفه ای کسب و کار را به صورت افقی قطع کند، یکپارچه سازی افقی این مدل ها مورد نیاز است. برای به هم پیوستن و اداره کردن فرایندهای کسب و کار واحدهایی که با هم کار می کنند، باید یکپارچه سازی افقی را نیز به حساب آورد. همانطور که در شکل 2-4 نشان داده شده، با یکپارچه سازی این مدل ها به صورت افقی، دو نوع یکپارچه سازی به دست می آید: یکپارچه سازی برنامه کاربردی درونی و یکپارچه سازی برنامه کاربردی بیینی (Mosawi et al. 2006).

2-9-1- یکپارچه سازی برنامه کاربردی بیینی

یکپارچه سازی برنامه کاربردی بیینی، به سازمان ها اجازه می دهد، جهت بهبود کارایی کسب و کار، فرایندهای کسب و کارشان را با فرایند کسب و کار همکار خود یکپارچه کنند. به منظور نائل شدن به یکپارچه سازی برنامه کاربردی بیینی، بایستی با توسعه یک زیر ساخت که ارتباطات بین تصمیم گیران، کارمندان، مشتریان، تامین کنندگان و همکاران را در بر می گیرد، فرایندهای کسب و کار موجود بین همکاران، هماهنگ و تعریف شود. این زیرساخت، اجازه تعامل بین موجودیت های کسب و کار مختلف را می دهد، در ضمن اجازه ردیابی تراکنش ها را در سرتاسر مرزهای سازمان های مختلف می دهد. این یکپارچه سازی، همچنین به کسب و کارها این امکان را می دهد تا به طور موثری روی اینترنت ارتباط داشته باشند. بنابراین هزینه های تراکنشی را کاهش داده، سیکل

¹ Mode of connectivity

زمانی را کوتاه کرده، خدمات دهی به مشتری را افزایش داده و همچنین زمان بازاریابی را کاهش می دهد. Biztalk، RozettaNet و EbXML تعدادی از فناوری های استاندارد هستند که مکانیزم ها و سمنتیک مشترکی برای یکپارچه سازی برنامه کاربردی بینی فراهم می کنند (Maheswari 2003, Mosawi et al. 2006).



شکل 2-4-2. یکپارچه سازی افقی معماری EAI

2-9-2- یکپارچه سازی برنامه کاربردی درونی

یکپارچه سازی برنامه کاربردی درونی، به دنبال به هم پیوستن برنامه های کاربردی سازمانی و تبادل اطلاعات در سطح سازمان است. در این جا، لازم است که سیستم های سازمانی، از سیستم های کسب و کار جدید بهره برداری کرده و با آنها یکپارچه شوند. برای دست یافتن به یکپارچه سازی برنامه کاربردی درونی، Mosawi و دیگران عقیده دارند که مدل برنامه کاربردی درونی را بایستی از طریق رویکردهای یکپارچه سازی زیر، با هم پیوند داد (Mosawi et al. 2006).

- یکپارچه سازی سرویس های برنامه کاربردی که شامل یکپارچه سازی داده ای، شیئی و فرایندی است.
- یکپارچه سازی میان افزار که شامل یکپارچه سازی پلتفرم و شبکه است.

2-9-2-1- یکپارچه سازی سرویس های برنامه کاربردی

یکپارچه سازی سرویس های برنامه کاربردی، مسئول مبادله داده ها یا توابع¹ یک برنامه کاربردی به دیگری برای انجام وظایف کسب و کار مشترک است. برای اینکه برنامه کاربردی، یکپارچه شود، شرکت بایستی در ابتدا تعیین کند کدام عنصر برنامه کاربردی (داده، شی یا فرایند)، نیاز به یکپارچه سازی دارد و نقطه دسترسی یکپارچه سازی چیست. Mosawi و دیگران پیشنهاد می کنند، برنامه های کاربردی را با یکی از رویکردهای زیر یکپارچه کند (Mosawi et al. 2006).

2-9-2-1-1- یکپارچه سازی داده ای

این یکپارچه سازی، در سطح پایگاه داده یا منبع داده ای رخ می دهد و با مهاجرت داده ها بین چند منبع داده ای حاصل می شود. داده های سازمانی، معمولاً بین چندین سیستم مدیریت پایگاه داده توزیع شده اند و می توانند در پایگاه داده های رابطه ای، پایگاه داده های شی گرا، فایل های مسطح موروثی² و بسیاری از منابع دیگر موجود باشند. هر منبع داده ای، ممکن است از یک یا چند واحد کسب و کار پشتیبانی کند و معنای³ این منابع داده ای، می تواند بر اساس واحد کسب و کاری که از آن پشتیبانی می کند، تغییر کند. بنابراین هر منبع داده ای، مدل سازی داده ای و شمای⁴ متفاوتی داشته و اغلب روی پلتفرم های متفاوتی اجرا می شود. راه حل های یکپارچه سازی، تلاش دارند تا داده ها را بین چندین منبع داده ای انتقال دهند، که این کار توسط یک رابط استاندارد منفرد که به داده های منابع داده ای متفاوت دسترسی دارد و با همگام سازی معنای⁵ برنامه های کاربردی متفاوت که در منابع داده ای متفاوتی موجودند، انجام می شود (Samtani et al. 2002, Mosawi et al. 2006). این

¹ Functions
² Legacy flat file
³ Semantic
⁴ Scema
⁵ Semantic synchronization

یکپارچه سازی، از سازگاری داده ای¹ تمام منابع داده ای پشتیبانی می کند. مکانیزم های متعددی برای انتقال داده بین چندین منبع داده ای، به صورت بلادرنگ و دسته ای²، مانند رونوشت برداری از داده³، استخراج، تبدیل و بارگذاری⁴ (ETL)، انبار داده⁵، بازار داده⁶، و پایگاه داده های فدرالی⁷ موجود است (Linthicum 2001, Samtani et al. 2002, Mosawi et al. 2006).

2-9-2-1-2- یکپارچه سازی شیئی

این یکپارچه سازی، اجازه می دهد، اشیائی که در سراسر یک سازمان توزیع شده اند، با هم تعامل داشته باشند و عملکرد خود را به اشتراک بگذارند. این روش، اجازه می دهد اشیاء از راه دور ایجاد شده و همچنین متدهای اشیاء از راه دور فراخوانی شوند. برای نائل شدن به یکپارچه سازی دو یا چند برنامه کاربردی با استفاده از یکپارچه سازی شیئی، معمار مجبور است، هر برنامه کاربردی را با استفاده از مکانیزم های یکپارچه سازی شیئی، مانند فناوری اشیاء توزیع شده (یا سرویس دهنده برنامه کاربردی⁸)، مجدداً بسازد. با این کار، منطق کسب و کار مشترک⁹ می تواند به اشتراک گذاشته شده و برای این که در آینده توسط سایر برنامه های کاربردی استفاده شود، در معرض نمایش قرار بگیرد (Zahavi 1999, Mosawi et al. 2006). فناوری های شیئی، اغلب اجازه می دهند که سیستم های موروثی، در لفافه قرار بگیرند¹⁰ و به صورت شی در معرض نمایش قرار گیرند. لفافه های برنامه کاربردی، داده های رابط کاربری مین فریم را کپسوله می کنند و آنها را به شی تبدیل می کنند. اگر سیستم موروثی، یک مرتبه در لفافه قرار گیرد، می تواند در یک محیط مبتنی بر شی سهیم شود (Samtani et al. 2002, Mosawi et al. 2006). برای دسترسی به اشیاء راه حل های بسته بندی شده، از رابط های برنامه نویسی کاربردی¹¹ (APIs)، استفاده می شود (Linthicum 2001, Mosawi et al. 2006).

-
- 1 Data consistency
 - 2 Batch
 - 3 Data replication
 - 4 Extract, Transform and Load
 - 5 Data warehouse
 - 6 Data Mart
 - 7 Federated Database
 - 8 Application server
 - 9 Common business logic
 - 10 Wrapped
 - 11 Application Programming Interfaces

2006). بسیاری از برنامه های کاربردی بسته بندی شده، مانند ERP و SCM، API های را برای تعامل با برنامه های کاربردی زیرین¹ و اشیاء کسب و کار، فراهم کرده اند. CORBA، COM و (EJB) J2EE، محبوب ترین ابزارهای فناوری اشیاء توزیع شده اند که از آنها، برای در لفافه قرار دادن منطق کسب و کاری که درون برنامه های کاربردی وجود دارد، استفاده می شود (Mosawi et al. 2006).

2-9-2-1-3- یکپارچه سازی فرایندی

این رویکرد از یکپارچه سازی، گردش اطلاعات را بین توده ای از برنامه های کاربردی موروثی، برنامه های کاربردی بسته بندی شده، برنامه های کاربردی کسب و کار الکترونیک² و سایر فرایندهای سازمانی مدیریت می کند. این روش، وابستگی هایی را تعریف می کند که جهت نائل شدن به فرایند سازمانی اثربخش، بین اشیاء و فرایندهای سازمانی متفاوت رخ می دهند. این رویکرد، فرموله سازی سیاست های عمومی کسب و کار را امکان پذیر می سازد که سبب اجرای نامتناقض فرایندهای کسب و کار سازمان می شود. این روش، سازمان ها را قادر به خودکار و یکپارچه سازی فرایندهای داخلی و خارجی کسب و کار، بر اساس قوانین کسب و کار سیستم های کسب و کار گوناگون می کند. راه حل های یکپارچه سازی فرایندی، معمولاً از چندین ابزار تشکیل شده اند که شامل ابزار مدل سازی، موتور فرایندی³ و ابزار راهبری⁴ است (Samtani et al. 2002, Mosawi et al. 2006). از ابزار مدل سازی، برای طراحی و مدل سازی فرایندهای کسب و کار به صورت گرافیکی استفاده می شود. این ابزار، با رسم نمودار گردش کار که منطق و منابع متفاوت کسب و کار را به هم پیوند می دهد، نحوه گردش اطلاعات بین برنامه های کاربردی را تعریف می کند. موتور فرایند کسب و کار، محیطی برای هماهنگ سازی اجرای فرایند و گردش آن با سایر شرکای فرایند، ارائه می دهد. ابزار راهبری، وظایف راهبری مانند مانیتورینگ فرایند کسب و کار و ایجاد گزارش هایی درباره کارایی فرایند کسب و کار را ارائه می دهد (Mosawi et al. 2006).

¹ Underlying applications
² E-business
³ Process engine
⁴ Administration

2-2-9-2- یکپارچه سازی میان افزار

یکپارچه سازی میان افزار، مسئول پایه گذاری زیرساخت ارتباطی برای اتصال فیزیکی و تعامل بین تمام مولفه های سیستم است. این یکپارچه سازی، مسائل لوله کشی¹، سیم کشی² و محیط زمان اجرا را برای انطباق با نیازمندی های کسب و کار، هدف قرار می دهد. آن اجازه اجرا، توالی و کنترل نمونه هایی از فرایندهای کسب و کار را در ارتباط با چند پروتکل می دهد. یکپارچه سازی میان افزار، می تواند از طریق یکپارچه سازی پلتفرم و شبکه حاصل شود (Mosawi et al 2006). در ادبیات، گاهی به این نوع از یکپارچه سازی، یکپارچه سازی زیرساخت³ نیز می گویند.

2-2-9-2-1- یکپارچه سازی پلتفرم

این یکپارچه سازی اجازه می دهد، انواع برنامه های کاربردی که روی پلتفرم های کاملاً متفاوتی اجرا می شوند، یکپارچه شوند. یکپارچه سازی پلتفرم، تنها مراقب در دسترس بودن ابزارها و فرایندها است تا برنامه های کاربردی که روی پلتفرم های متفاوتی قرار دارند، اجازه یابند به صورت بهینه و امن ارتباط برقرار کنند، به شکلی که داده ها و توابع بتوانند به صورت قابل اطمینانی⁴ منتقل شوند، مانند ارسال مطمئن داده ها از یک ماشین NT به یک ماشین لینوکس (Mosawi et al. 2006).

2-2-9-2-2- یکپارچه سازی شبکه

این یکپارچه سازی، با ترجمه پروتکل های غیرمشابه به همدیگر، تبادل اطلاعات بین شبکه های ناسازگار⁵ را امکان پذیر می سازد. یکپارچه سازی شبکه، به مثابه پلی⁶ برای پر کردن شکاف بین شبکه های ناسازگار است، به شکلی که داده ها بتوانند بین سیستم های متفاوت منتقل شوند (Zahavi 1999, Mosawi et al. 2006). هدف این یکپارچه سازی، برقراری اتصال بین برنامه های کاربردی و ماژول هایی است که روی شبکه های متفاوتی قرار دارند. این اتصال، تضمین می کند که داده ها یا پیغام ها می توانند از هر شبکه ای به دیگری ارسال شوند. اتصال شبکه ای از طریق فناوری های

¹ Plumbing

² Wiring

³ Infrastructure integration

⁴ Reliably

⁵ Incompatible networks

⁶ Bridge

شبکه و پروتکل های استاندارد می مانند TCP/IP به دست می آید (Zahavi 1999, Mosawi et al.)
(2006).

2-10-10- انواع رویکردهای یکپارچه سازی بر اساس درجه پیچیدگی آن

Yoo (2004)، رویکردهای یکپارچه سازی را به 6 دسته کلی تقسیم می کند که از یک روش ساده و سریع شروع شده و به یک راه حل پیچیده ختم می شود. در ادامه رویکردهای مختلف را به ترتیب از ساده ترین به پیچیده ترین نوع مشاهده می کنید.

2-10-10-1- انتقال داده های مشترک¹

در این رویکرد، فایل های متنی بین برنامه های کاربردی که قرار است یکپارچه شوند، به اشتراک گذاشته می شود، برای نمونه، به منظور اشتراک اطلاعات بین برنامه های کاربردی متمایز، از زبان های اسکریپتی به همراه نوعی از میان افزارهای پیغام گرا استفاده می شود (Vinoski 2002, Yoo)
(2004). این رویکرد، با توجه به امکان ساخت سریع الگوی اولیه و همچنین توسعه پذیری آسان آن، توسط تعداد بسیار زیادی از گروه ها جهت انجام پروژه های یکپارچه سازی ساده و کوتاه مدت استفاده می شود (Yoo 2004).

2-10-10-2- یکپارچه سازی در سطح پایگاه داده

این رویکرد، به منظور اشتراک محتوای پایگاه داده ها به صورت سازگار²، به دنبال اتصال پایگاه های داده از طریق یکپارچه سازی پایگاه داده با پایگاه داده و یا ساخت یک پایگاه داده فدرالی³ می باشد. رونوشت داده ای⁴، تنها به دنبال انتقال داده بین دو یا چند پایگاه داده می باشد. در حال حاضر، بسیاری از میان افزارهای پایگاه داده گرا، با تبدیل و انطباق⁵ عناصر پایگاه داده های نامتجانس،

Common Data Transfer¹
Consistent²
Federated Database³
Data replication⁴
Adjust⁵

سرویس رونوشت پایگاه داده را ارائه می دهند. پایگاه داده فدرالی، تلاش دارد تا چندین پایگاه داده و مدل پایگاه داده را از طریق یک پایگاه داده مجازی فدرالی یکپارچه کند تا یک نمای متحدالشکل منفرد بدست آورد (Yoo 2004).

2-10-3- پردازش رابط¹

این رویکرد، از رابط های تعریف شده برنامه های کاربردی استفاده کرده و یک رابط متحدالشکل ارائه می دهد که به تمام سرویس های داخلی برنامه های کاربردی دسترسی دارد. هدف این رویکرد، متغیر بوده و می تواند وحدت اطلاعات² یا وحدت سرویس ها³ و یا هر دوی آنها باشد. JCA⁴ نمونه ای از این نوع یکپارچه سازی است که نقش واسط بین سیستم های ERP متفاوت را دارد (Yoo 2004).

2-10-4- یکپارچه سازی در سطح فرایند

یکپارچه سازی در سطح فرایند یا به عبارت دیگر یکپارچه سازی فرایند کسب و کار، برنامه های کاربردی موجود را در نظر می گیرد و با احتساب آنها در فرایندهای نهایی، به ساخت فرایندهای کسب و کار سازمانی⁵ می پردازد. مشخصه این رویکرد، ارائه لایه دیگری با نام مدیریت فرایند در بالای سایر برنامه های کاربردی است. این لایه از کنترل منطق و گردش اطلاعات بین فرایندهای موجود که برنامه های کاربردی فعلی را نیز در بر دارد، پشتیبانی می کند (Yoo 2004).

2-10-5- یکپارچه سازی سرویس گرا⁶

یکپارچه سازی سرویس گرا، به برنامه های کاربردی اجازه می دهد تا منطق یا متد کسب و کار را به اشتراک بگذارند. سرویس های وب، نمونه های خوبی از این نوع از یکپارچه سازی هستند. هدف

¹ Interfacing processing
² Information unification
³ Services unification
⁴ J2EE Connector Architecture
⁵ Enterprise-wide business service
⁶ Service-Oriented Integration

اصلی این روش، ترکیب سرویس های چندین برنامه کاربردی و ارائه مبتنی بر وب آنها در قالب یک زیرساخت سرویس دهی متحدالشکل به کاربران نهایی می باشد. کاربر قادر است با اتصال به یک سرویس ترکیبی^۱، از سرویس های متعددی برنامه های کاربردی موجود سود ببرد (Yoo 2004).

2-10-6- یکپارچه سازی درون یک سازمان خدماتی^۲

تعدادی از پژوهشگران پیش بینی کرده اند که جهت گیری آینده اطلاعات کسب و کار، به سمت یکپارچه سازی جهانی می رود، بنابراین بایستی با کمک تحلیل سازمانی، نقش هر موجودیت را در در این جهت گیری تعیین کرده و آن را به سمت مقصد نهایی هدایت کرد (Basu and Kumar 2002, Zambonelli et al. 2000, Yoo 2004). ما بایستی علاوه بر اینکه روی ساخت یک نمای متحدالشکل سازمانی تمرکز کنیم، لازم است این را نیز به خاطر داشته باشیم که سرویس های کسب و کار متمایز باید توانایی تعامل داشته باشند و بتوان با ترکیب آنها، سرویس های دیگری ارائه داد که ارزش افزوده داشته باشند (Yoo 2004).

Yoo (2004)، عقیده دارد که یکی از تفاوت های یکپارچه سازی سرویس گرا و سازمان خدماتی، این است که در یک سازمان خدماتی، برنامه های کاربردی برای نیل به هدف سرویس خود، سرویس های دیگر را، برنامه های کاربردی همکار خود می دانند، درحالی که برنامه های کاربردی در یک سرویس یکپارچه^۳ (سرویس ترکیبی) از سایر برنامه های کاربردی اطلاعی ندارند و ویژگی های هر سرویس، تنها برای یکپارچه سازهای برنامه کاربردی نمایان است (Yoo 2004).

Composite Service¹
Integration within a service organization²
Integrated service³

2-11- نتیجه گیری

در این فصل، دسته بندی ها و رویکردهای اصلی EAI که در ادبیات موجود است، مورد بررسی قرار گرفت. این دسته بندی ها، شامل دسته بندی بر اساس قوی یا ضعیف بودن یکپارچه سازی، دسته بندی بر اساس طرف های درگیر در یکپارچه سازی (سازمان، سازمان های همکار و مشتریان)، دسته بندی بر اساس نوع سیستم های نرم افزاری درگیر در یکپارچه سازی، دسته بندی بر مبنای سطح مطلوب برای یکپارچه سازی، دسته بندی بر پایه عناصر سیستم های درگیر در یکپارچه سازی، دسته بندی بر اساس لایه های معماری مورد نیاز در یکپارچه سازی، دو نوع دسته بندی بر مبنای معماری مدل محور و همچنین یک دسته بندی بر اساس درجه پیچیدگی رویکردهای یکپارچه سازی می باشد.

فصل سوم

معماری و توپولوژی
توانمند سازهای EAI

3-1- مقدمه

هنگام پیاده سازی واقعی پروژه های EAI، بر اساس نوع، تعداد و پیچیدگی سیستم هایی که قرار است، یکپارچه شوند، بایستی از فناوری های توانمند ساز مناسب استفاده شود. برای یکپارچه سازی برنامه های کاربردی، دو گزینه مطرح شده است. گزینه اول، برقراری ارتباط مستقیم بین سیستم مبدا و مقصد است که در ادبیات با نام یکپارچه سازی نقطه به نقطه¹ (P2P) مطرح است. این روش، معماری و توپولوژی پیچیده ای ندارد و برای مسائل کوچک و سبک مناسب است، اما با توجه به اینکه در این روش، پیوند سیستم ها قوی است، برای مسائل پیچیده زیاد مناسب نمی باشد. گزینه بعدی، استفاده از یک لایه واسط می باشد که این لایه با ایجاد یک زیرساخت یکپارچه سازی از برقراری ارتباط مستقیم بین برنامه های کاربردی مبدا و مقصد جلوگیری می کند که سبب پیوند ضعیف بین سیستم های درگیر در یکپارچه سازی می شود. این راه حل، امکانات زیادی را برای انجام هر چه آسان تر پروژه های EAI فراهم می کند. این نوع از یکپارچه سازی در ادبیات، معمولاً با نام محصولات EAI یا سرویس دهنده یکپارچه سازی² شناخته می شود. البته محصولات ESB³ که در معماری های سرویس گرا⁴ (SOA) مطرح هستند و همچنین سیستم های BPM⁵ نیز، جزء این

¹ Point-to-Point integration

² Integration server

³ Enterprise Service Bus

⁴ Service Oriented Architecture

⁵ Business Process Management

دسته بندی از یکپارچه سازی به حساب می آیند. با ظهور این محصولات، به معماری ها و توپولوژی های پیشرفته تری نیاز است که در این فصل سعی شده به این نیاز پاسخ داده شود. در این فصل، ابتدا به مرور معماری کلی محصولات و نرم افزارهای توانمند ساز EAI پرداخته شده، سپس یک چارچوب کلی برای یکپارچه سازی B2B ارائه شده و در نهایت توپولوژی های پیاده سازی این محصولات مورد مطالعه قرار گرفته است.

2-3- معماری و لایه بندی EAI

معماری های متفاوتی از محصولات و فناوری های EAI در ادبیات آمده است، اما یکی از بهترین آنها معماری است که توسط Erasala و دیگران (2003) ارائه شده و شامل چهار لایه است. البته Themistocleous و دیگران (2004) پیشنهاد دادند که لایه دیگری با نام خدمات اتصال نیز به این معماری اضافه شود. با در نظر گرفتن این لایه اضافی، می توان گفت که اکثر محصولات EAI، وظایفشان را در قالب لایه هایی که در شکل 3-1 نشان داده شده، ارائه می دهند.

مدیریت فرایند (قوانین کسب و کار، فرایند کسب و کار)
خدمات تبدیلی (تبدیل داده)
خدمات توزیعی (مسیریابی پیغام)
خدمات ارتباطی (انتقال داده)
خدمات اتصال (آداپتورها/کانکتورها)

شکل 3-1- معماری محصولات EAI

3-2-1- لایه خدمات اتصال¹

این لایه، مسئول ایجاد رابط های اتصال² بین برنامه های کاربردی و زیرساخت مرکزی یکپارچه سازی است. از طریق این اتصالات، عناصر برنامه کاربردی از مولفه های IT، به لایه خدمات ارتباطی ارسال می شوند (Themistocleous et al. 2004). آداپترها/کانکتورها، اجزاء اصلی این لایه هستند و این واژه ها گاهی اوقات در ادبیات به جای هم به کار برده می شوند. هدف آداپتر، ارائه یک لایه عایق بین API مولفه های IT و زیرساخت یکپارچه سازی است. آداپترها، امکان EAI غیرهجومی و پیکرندی با پیوند ضعیف را فراهم می کنند (Papazoglou and Heuvel 2005).

با استفاده از آداپترها، پیچیدگی رابط های برنامه های کاربردی از کاربر نهایی یا حتی توسعه دهنده راه حل EAI، پنهان می ماند (Linthicum 2003). اما هدف کانکتور، کپسوله سازی مکانیزم ارتباطی بین مولفه ها است. تمایز عمده آداپترها با کانکتورها این است که آداپتر، پلی برای مشکل تعامل برنامه های کاربردی³ است، یعنی مولفه ها بدون آداپترها، قادر نیستند با همدیگر کار کنند، در حالی که کانکتور، ارتباطات بین مولفه های متعامل را امکان پذیر می سازد (Papazoglou and Heuvel 2005). آداپترها، بر اساس نوع رابط برنامه های کاربردی⁴ مبدا یا مقصد، می توانند اطلاعات گرا⁵ یا سرویس گرا⁶ باشند (Linthicum 2003). در فصل چهارم، آداپترهای اطلاعات گرا و سرویس گرا توصیف شده اند.

3-2-2- لایه خدمات ارتباطی⁷

این لایه، مسئول انتقال داده ها است. این لایه ممکن است بر پایه RPC باشد، اما اغلب بر پایه یک سیستم صف بندی پیغام⁸ است که می تواند پیغام ها را بصورت همزمان⁹ یا غیر همزمان¹⁰ و با قابلیت

¹ Connectivity services layer

² Connection-interfaces

³ Application's Interoperability

⁴ Application Interface

⁵ Information-oriented

⁶ Service-oriented

⁷ Communication service layer

⁸ Message queuing

⁹ Synchronously

¹⁰ Asynchronously

ذخیره و ارسال¹ بفرستد. مولفه های IT، معمولاً با استفاده از آداپترها/کانکتورها (لایه خدمات اتصال)، به لایه خدمات ارتباطی متصل شده اند و لایه خدمات اتصال، نگاشت بین مولفه های IT و زیرساخت یکپارچه سازی را انجام می دهد (Erasala et al. 2003).

3-2-3- لایه خدمات توزیعی²

این لایه، مسئول توزیع اطلاعات (مسیریابی پیغام ها) به مولفه های گوناگون برنامه کاربردی است. این لایه به یک سبک درخواست/پاسخ³ (نقطه به نقطه⁴) یا انتشار/اشتراک⁵، کار می کند. مسیریابی پیغام، بر اساس موضوع⁶ یا محتوای⁷ پیغام ورودی است (Erasala et al. 2003).

3-2-4- لایه خدمات تبدیلی⁸

داده های ورودی مورد نیاز یک مولفه IT، ضرورتاً در قالب های پشتیبانی شده توسط آن مولفه نیست. بنابراین لایه خدمات تبدیلی، عملیات معتبر سازی⁹ و ترجمه¹⁰ مورد نیاز را انجام می دهد. ورودی های مورد نیاز یک مولفه، اغلب اوقات ممکن است از بیش از یک منبع بیایند. احتمالاً چنین ورودی هایی، در زمان های متفاوتی می رسند. بنابراین این لایه، خدمات همگام سازی¹¹ را که قبل از ارسال تمام ورودی ها به مولفه IT لازم است، ارائه می دهد. در هنگام تبدیل، از جامعیت تراکنشی نیز نگهداری می شود (Erasala et al. 2003) و سعی می شود مشکلات جامعیتی داده ای و تراکنشی به وجود نیاید.

¹ Store-and-forward
² Distribution services
³ Request/response
⁴ Point-to-Point
⁵ Publish/subscribe
⁶ Subject
⁷ Content
⁸ Transformation service
⁹ Validation
¹⁰ Translate
¹¹ Synchronization

3-2-5- لایه مدیریت فرایند

قوانین کسب و کار¹، بخش مهمی از تعاریف کسب و کار هستند و بر گردش فرایند کسب و کار نظارت می کنند. برای مثال، اگر یک مشتری، یک وام منزل را مجدداً تجارت می کند و دوره آنرا از 30 سال به 15 سال کاهش می دهد، پرداخت جدید نمی تواند از 120% پرداخت قدیمی تجاوز کند و اگر تجاوز کرد، آنگاه باید پردازش اضافی رخ دهد. لایه مدیریت فرایند²، عملکردهایی برای نگاشت قوانین کسب و کار و پشتیبانی از فرایندهای کسب و کار فراهم می کند. این لایه، اساساً لایه خدمات تبدیلی را جهت تکمیل فرایند کسب و کار، هماهنگ و کنترل می کند. لایه مدیریت فرایند، کل فرایندهای کسب و کار را مدیریت می کند و یکپارچه سازی فرایندهای درون سازمانی و بین سازمانی را امکان پذیر می سازد. عملکرد این لایه، گردش کاری سازمانی³ را امکان پذیر می سازد. این لایه به شما اجازه می دهد، در موقع نیاز، مولفه ها را به سرعت اسمبل یا اسمبل مجدد کرده و زمان رسیدن محصول به بازار⁴ را برای توسعه محصول کاهش دهید (Erasala et al. 2003).

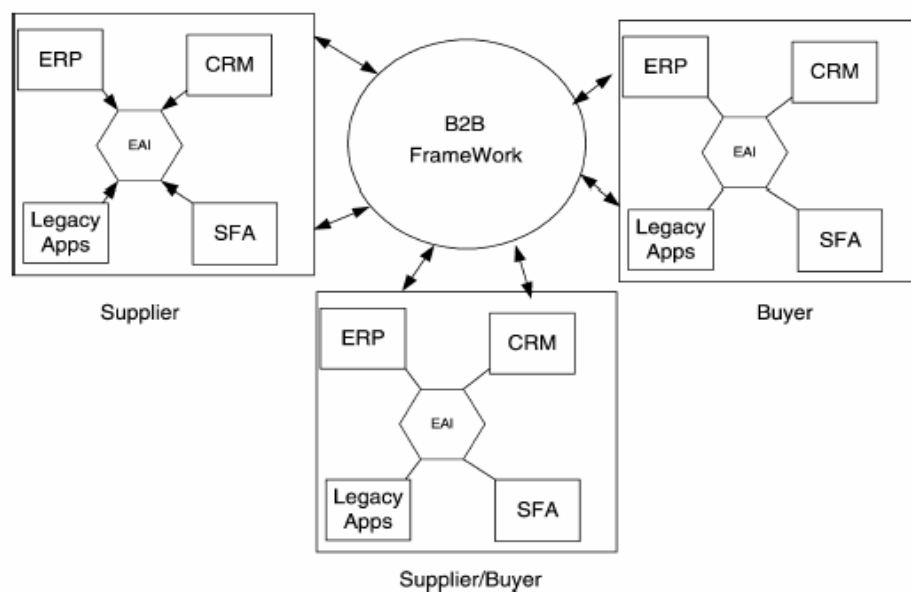
3-3- چارچوب یکپارچه سازی B2B

درحالی که بسیاری از فنون EAI در درون یک شرکت قابل اجرا هستند، اما لزوماً بین شرکت ها کاربرد ندارد. برای مثال، درحالی که یک شرکت ممکن است از فناوری پیغام رسانی برای یکپارچه سازی برنامه های کاربردی داخلی اش استفاده کند، اما عاقلانه نیست که از همکاران تجاری اش انتظار داشته باشد که وی نیز از فناوری پیغام رسانی مشابهی برای یکپارچه سازی برنامه های کاربردی اش استفاده کند (Skinstad 2000, Erasala et al. 2003). چارچوب های B2B برای برطرف کردن این نیاز تکامل یافته اند. در ادامه با تفصیل بیشتری معماری این چارچوب ها مورد بررسی قرار می گیرد.

چارچوب های B2B، الگوهای عمومی هستند که کسب و کارها را قادر به ارتباطات کارآمد روی اینترنت می کنند (Taylor 2000, Erasala et al. 2003). چارچوب های B2B، علاوه بر اینکه به

¹ Business rules
² Process management
³ Enterprise workflow
⁴ Time to market

طور موثری با سایر کسب و کارها تعامل دارند، با برنامه های کاربردی سازمانی موجود نیز به طور کارآمدی یکپارچه می شوند. آنها به جای یک رابط نقطه به نقطه، یک رابط مرکزی مشترک (یک لایه واسط مانند EAI) تعریف کرده و مکانیزم های تعامل موجودیتهای کسب و کار مختلف را ارائه می دهند (شکل 2-3).



شکل 2-3- چارچوب یکپارچه سازی B2B

چارچوب B2B، گرایش به کار در لایه مدیریت فرایند در معماری EAI دارد. برنامه های کاربردی B2B روی معنای کسب و کار¹ تمرکز دارند، بنابراین واژگان استفاده شده بین همکاران نیز بایستی هماهنگ شده و به صورت شایسته تعریف شود.

RosettaNet، یک چارچوب B2B اختصاصی است که برای هماهنگ کردن معانی تا جای ممکن، از فرهنگ های لغت² صنعتی استاندارد استفاده می کند. موردی را در نظر بگیرید که یک محصول برای فروش (با یک شماره جنس، توضیحات و قیمت) برای همکاران دیگر در دسترس باشد. چارچوب B2B، یک دنباله از تبادلات³ مبتنی بر XML را تبیین کرده و تعریف می کند که داده ها باید چطور به اشتراک گذاشته شده و هماهنگ شوند.

¹ Semantic
² Dictionaries
³ Exchange

RosettaNet از یک فرایند ارتباط با همکار¹ (PIP)، برای تعریف دیالوگ (محتوای XML) تعامل برنامه کاربردی با برنامه کاربردی) یا تبادل اسناد XML مورد نیاز بین همکاران استفاده می کند. چارچوب RosettaNet، همچنین تبادلات کسب و کار مختص قلمرو² را که فرایندهای کسب و کار اختصاصی به آنها نیاز دارند، در قالب XML تعیین می کند. این ترکیب داده های مختص قلمرو و داده های فرایند کسب و کار که در قالب XML است، به یک برنامه کاربردی آگاه از وضعیت³ گره خورده است (به طور مثال یک برنامه کاربردی جاوا یا یک نرم افزار EAI) که این برنامه کاربردی، محتوای XML را تحویل مقصد می دهد و به محتوای XMLی که خود دریافت می کند، واکنش نشان می دهد. این برنامه کاربردی پس از انجام مبادله⁴ سیستم های کسب و کار را به روز می کند، اما توجه کنید که سرویس های برنامه های کاربردی مبدا یا مقصد، به طور مستقیم توسط همکار تجاری فراخوانی نمی شود (McCoy 2000, Erasala et al. 2003).

3-4- توپولوژی های پیاده سازی EAI

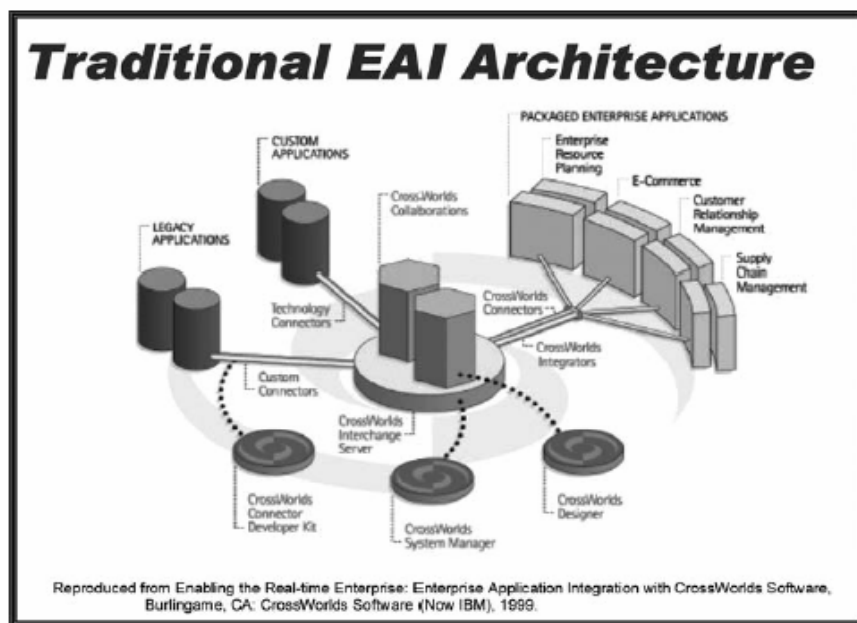
بر اساس یک ارزیابی مقایسه ای که توسط Doculab از شش فروشنده EAI انجام شده است، سرویس دهنده های یکپارچه سازی⁵، پیرامون دو توپولوژی فیزیکی عمده، ساخته شده اند: مدل هاب و اسپیک⁶ و مدل باس شبکه ای⁷ (Sachenz et al. 2001, Erasala et al. 2003).

3-4-1- مدل هاب و اسپیک

یک مدل هاب و اسپیک، معادل توپولوژی ستاره ای شبکه ای است. توپولوژی ستاره ای، یکی از ساختارهایی است که به گستردگی برای سیستم های ارتباطی استفاده شده است. استفاده زیاد از شبکه ستاره ای، به دلایلی چون آسانی کنترل آن، پیچیده نبودن نرم افزار و سادگی گردش تراکنشی آن

¹ Partner Interface Process
² Domain-specific business exchanges
³ State-aware application
⁴ Exchange
⁵ Integration server
⁶ Hub-and-spoke
⁷ Network bus

است. ایزوله کردن خرابی¹ نیز، در یک شبکه ستاره ای زیاد سخت نیست، چون برای شناسایی مشکل، خطی که دچار مشکل شده می تواند ایزوله شود (Eshel 2000, Erasala et al. 2003). در یک رویکرد هاب و اسپیک، سرویس دهنده یکپارچه سازی مرکزی (هاب)، به عنوان یک دلال پیغام² برای کنترل ارتباطات، ترجمه داده ها و فعل و انفعالات فرایندی بین سیستم های متصله عمل می کند، که سبب حداقل شدن اتصالات، نسبت به یکپارچه سازی نقطه - به - نقطه (P2P) می شود. در یکپارچه سازی P2P، اگر هر کدام از n سیستم داده شده، نیاز به یک اتصال دوطرفه با همدیگر داشته باشند، در مجموع $n*(n-1)$ اتصال لازم است، اما در این معماری، یک هاب مرکزی، اتصال را با هر برنامه کاربردی، مدیریت می کند، در نتیجه به جای $n*(n-1)$ اتصال، به n اتصال نیاز داریم و به طور قابل توجهی، هزینه یکپارچه سازی و نگهداری کاهش می یابد (احتمالاً به یک سوم). علاوه بر این، از منظر مدیریتی، این رویکرد مدیریت و نگهداری را آسان می کند (Erasala et al. 2003). یک نمونه از کاربرد این معماری، در شکل 3-3 نمایش داده شده است (Gulledge 2006). محصول WBI Message Broker ساخت شرکت IBM، یک نمونه محصول بر اساس این معماری است (Pereira 2006).



شکل 3-3- مدل هاب و اسپیک برای EAI

¹ Fault isolation
² Message broker

اما عمده ترین مشکل مدل هاب و اسپیک، عدم مقیاس پذیری¹ آن است، چون نیاز دارد، تمام اطلاعات رد و بدل شده بین برنامه های کاربردی را درون یک هاب منفرد پردازش کند. Linthicum. ضعف این مدل را در محدود به منبع بودن² آن می داند، چون سرانجام تعداد سیستم های متصله و ترافیک اطلاعاتی، منابع موجود سرویس دهنده را اشباع خواهند کرد و منجر به کاهش کارایی خواهد شد. مشابه توپولوژی ستاره ای در پیکر بندی شبکه ای، تنگناهای بالقوه و یک نقطه منفرد خرابی، عمده ترین چالش های مدل هاب و اسپیک هستند (Linthicum 2000, Erasala et al. 2003). مشاهده می شود، در جاهایی که سیستم ها از نظر جغرافیایی توزیع شده اند، پیاده سازی این معماری واقعاً مشکل است (Pereira 2006). برای حل این مشکلات، دو نوآوری عمده روی معماری های هاب و اسپیک سستی انجام شده است: ساختار ایستای فدرالی³ و پویای فدرالی⁴ (Linthicum 2001, Erasala et al. 2003).

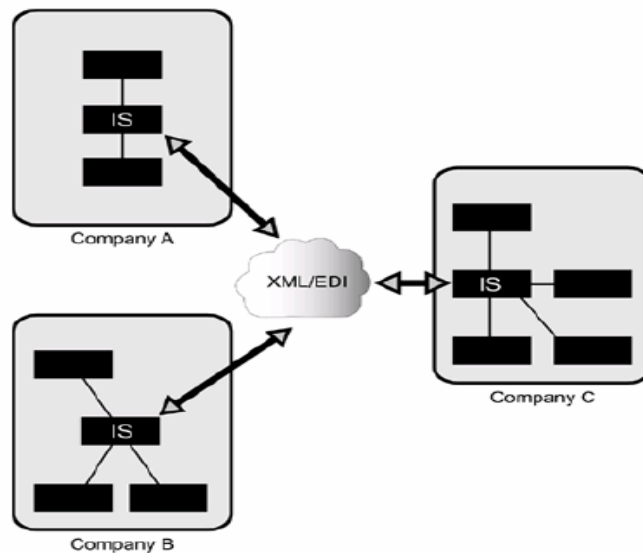
3-4-1-1- مدل ایستای فدرالی

این ساختار به برنامه های کاربردی و منابع داده ای، اجازه می دهد به طور ایستا به یک هاب منفرد متصل شوند، در همین زمان هاب های دیگر، به این هاب اصلی متصل می شوند و می توانند اطلاعات را با یکدیگر مبادله کنند (Erasala et al. 2003). همانطور که در شکل 3-4 نشان داده شده، تبادل اطلاعات بین این هاب ها، از طریق یک قالب مشترک مانند XML یا EDI می باشد. در این مدل که بیشتر برای یکپارچه سازی بین سازمانی مناسب است، چند نوع متمایز از سرویس دهنده های یکپارچه سازی، به صورت مستقل از همدیگر کار می کنند تا تشکیل یک انجمن تجاری بدهند (Linthicum 2003). در این مدل، هاب ها بار پردازشی⁵ را به طور پویا به اشتراک نمی گذارند. بنابراین درحالی که این معماری نسبت به معماری هاب و اسپیک، مقیاس پذیری بهتری دارد، اما انعطاف پذیر نیست (Erasala et al. 2003).

Lack of scalability¹
Resource-constrained²
Federated static³
Federated dynamic⁴
Load processing⁵

3-4-1-2- مدل پویای فدرالی

این ساختار، بر خلاف مدل ایستا، به هاب‌ها این توانایی را می‌دهد تا بار پردازشی را به طور پویا بین تمام سرویس‌دهنده‌های یکپارچه سازی موجود به اشتراک بگذارند. اگرچه این روش یک راه حل ایده آل برای برطرف کردن تمام چالش‌هایی است که مدل هاب و اسپیک با آن روبرو است، اما با توجه به پیچیدگی و پیشرفته بودن آن، هیچ فروشنده EAI وجود ندارد، که از این معماری پشتیبانی کند (Erasala et al. 2003).



شکل 3-4- پیکربندی ایستای فدرالی، شامل چند سرویس‌دهنده‌های یکپارچه سازی¹ (IS) متمایز

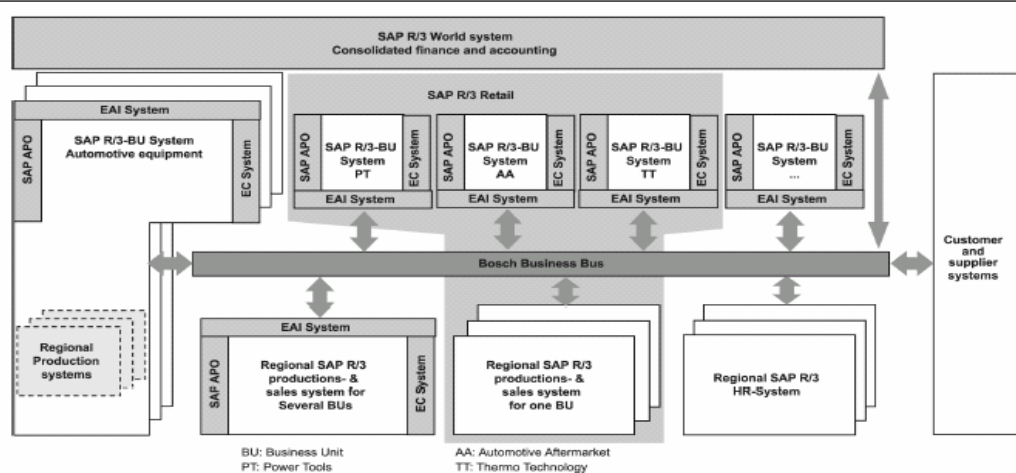
3-4-2- مدل باس شبکه ای

برای پیاده سازی فیزیکی مدل باس، باید به ازای هر برنامه کاربردی یا سیستم یکپارچه ای (ERP) که قرار است اضافه شود، آداپترهایی روی باس قرار گیرد، سپس این برنامه‌های کاربردی، برای تعامل با سرویس‌دهنده یکپارچه سازی و سایر سیستم‌های متصله، از ستون فقرات باس²، استفاده می‌کنند (Sanchez et al. 2001, Erasala et al. 2003). این نوع از معماری باس، گاهی اوقات انتشار/اشتراک³ (Pub/sub) نامیده می‌شود. برنامه کاربردی مبدا که یک⁴ رویداد را تولید می‌کند،

¹ Integration server
² Bus backbone
³ Publish/subscribe
⁴ Event

منتشرکننده¹ یا تولید کننده² نامیده شده و برنامه های کاربردی مقصد که رویداد را دریافت می کند، مشترک³ یا مصرف کننده⁴ نامیده می شوند. در این رویکرد، داده ها در قالبی که سازمان روی آن توافق کرده، منتشر می شوند. برنامه های کاربردی مبدا، داده ها یا پیغام هایی را که می خواهند به اشتراک بگذارند، از طریق آداپترها که عملیات تبدیل داده ای و مسیریابی به سمت نود بعدی را انجام می دهند، به سمت دلال پیغام باس (موتور pub/sub) ارسال می کنند. دلال پیغام، اطلاعات را مجدداً به سمت برنامه های کاربردی مقصد توزیع می کنند. مسئولیت تبدیل داده ها به قالبی که توسط برنامه های کاربردی مصرف کننده درک شود، به عهده آداپترهای منتشر کنندگان است (Pereira 2006).

در توپولوژی باس، عملیات نگاشت، تبدیل داده ای و مسیریابی به سمت نود بعدی، به صورت توزیع شده توسط آداپترها/کانکتورها انجام می شود، اما در معماری هاب و اسپیک، این عملیات بصورت متمرکز در هاب انجام می شود. این مدل در مقایسه با مدل هاب و اسپیک، مشکل مقیاس پذیری را حل کرده و کارایی بهتری ارائه می دهد. البته در این مدل، همانطور که کل سیستم رشد می کند، راهبری آن پیچیده تر و مشکل تر می شود (Erasala et al. 2003). نمونه ای از کاربرد واقعی این مدل در شرکت Bosch Group، در شکل 3-5 نشان داده شده است (Puschmann and Alt 2004).



شکل 3-5- معماری سیستم اطلاعاتی شرکت Bosch

1 Publisher
2 Producer
3 Subscriber
4 Consumer

3-5- نتیجه گیری

در این فصل، معماری، چارچوب و توپولوژی کلی نرم افزارها و محصولات EAI مورد بررسی قرار گرفت. بعد از مرور نقاط قوت و ضعف این معماری ها، می توان گفت که برای پیاده سازی EAI در سطح سازمان های بزرگ، باید از توپولوژی پویای فدرالی و یا توپولوژی باس شبکه ای استفاده کرد. اما با وجود اینکه توپولوژی پویای فدرالی، راه حلی ایده آل برای برطرف کردن بسیاری از چالش هایی است که در EAI با آن مواجه هستیم، اما با توجه به پیچیدگی و پیشرفته بودن این توپولوژی، هنوز هیچ فروشنده محصولات یکپارچه سازی به صورت کامل از آن پشتیبانی نمی کند، بنابراین بهترین گزینه باقی مانده برای پروژه های پیچیده و بزرگ، محصولاتی مانند ESB است که از توپولوژی باس شبکه ای استفاده می کنند. البته مشکل محصولاتی که از توپولوژی باس استفاده می کنند، این است که با رشد کل سیستم، راهبری آن پیچیده تر و مشکل تر می شود.

فصل چهارم

تشریح، مقایسه و
ارزیابی راه حل ها و
فناوری های EAI

4-1- مقدمه

هنگام پیاده سازی واقعی پروژه های EAI، بر اساس نوع، تعداد و پیچیدگی سیستم هایی که قرار است، یکپارچه شوند، بایستی از فناوری های توانمند ساز مناسب استفاده شود. در گذشته، به این فناوری های، میان افزار می گفتند. اصولاً به هر فناوری یا نرم افزاری که ارتباطات دو یا چند سیستم نرم افزاری را امکان پذیر می سازد، میان افزار¹ گفته می شود. میان افزارهای اولیه، کارایی لازم را برای پاسخ به نیازهای یکپارچه سازی نقطه به نقطه ارائه می دادند، اما برای مسائل پیچیده مناسب نبودند، بنابراین با افزایش پیچیدگی و بلوغ یکپارچه سازی و نیاز به راه حل های کامل تر و پیشرفته تر، فروشندگان نرم افزار، با توسعه این میان افزارها، محصولاتی با نام EAI ارائه دادند. البته با توجه به اینکه، هیچ فناوری منفردی وجود ندارد که از یکپارچه سازی تمام عناصر برنامه های کاربردی پشتیبانی کند، محصولات EAI با ترکیب چندین فناوری یکپارچه سازی بنا شده اند. بنابراین، می توان گفت که محصولات EAI، چارچوبی شامل چندین فناوری مانند سرویس دهنده برنامه های کاربردی²، آداپتورها و دلال های پیغام می باشند.

البته فناوری ها و محصولاتی که امکان EAI را فراهم می کنند، فراتر از اینها هستند و بر اساس نوع معماری سیستم های نرم افزاری، فناوری های زیادی مطرح شده است که در این فصل سعی می شود، موارد مهم آنها معرفی شده و با یکدیگر مقایسه گردد.

¹ Middleware
² Application servers

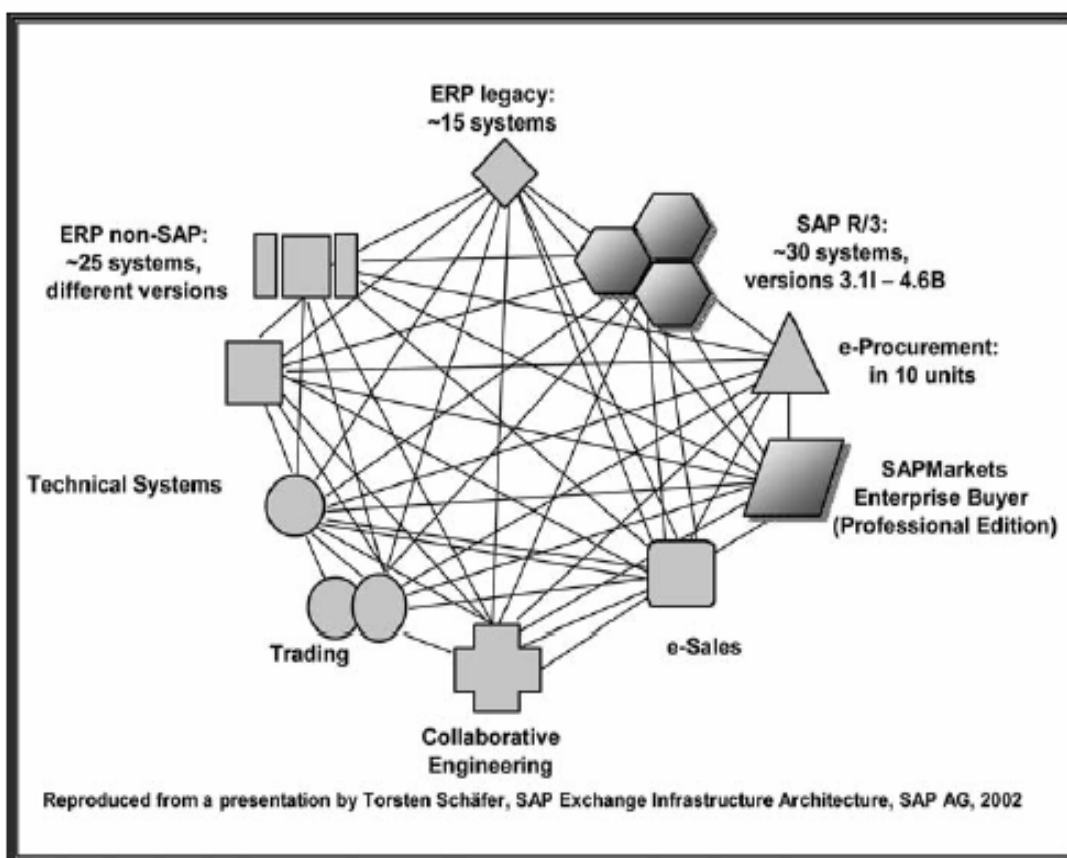
در این فصل، ابتدا فناوری های متعارف توصیف شده، سپس فناوری های رایج در یکپارچه سازی با همدیگر مقایسه می شوند. در نهایت، چارچوبی جهت ارزیابی فناوری های یکپارچه سازی ارائه می شود که هر سازمان می تواند با استفاده از آن، فناوری یکپارچه سازی مورد نیاز خود را شناسایی کرده و بکار گیرد.

4-2- فناوری های یکپارچه سازی نقطه به نقطه

یکپارچه سازی نقطه به نقطه، گران ترین نوع یکپارچه سازی است. منظور از یکپارچه سازی نقطه به نقطه، توسعه دو به دو رابط¹ بین سیستم مبدا و مقصد است. مدل داده ای سیستم مبدا و مقصد معلوم است و فردی (مثلاً یکپارچه ساز سیستم) برنامه لازم را برای تبادل اطلاعات بین این سیستم ها توسعه می دهد (Gullede 2006). با توجه به اینکه در این روش سنتی، برقراری ارتباط از طریق کد نویسی است، احتمال تغییر سیستم مبدا یا مقصد وجود دارد. در واقع برای ارتباط بین دو برنامه کاربردی، باید کد مدیریت تراکنش ها بازنویسی شود. مدل سنتی این نوع تعامل، از طریق مکانیزم فراخوانی تابع از راه دور² (RPC) بود و فناوری های مرتبط بود. گاهی اوقات از محصولات شتاب دهنده³ با نام میان افزار استفاده می شود که یک مثال خوب از این محصولات، IBM MQSeries می باشد که اکنون به عنوان بخشی از WebSphere قرار داده شده است. MQSeries یک میان افزار پیغام گرا⁴ (MOM) است که به نوشتن کد در سیستم مبدا و مقصد نیاز دارد (Gullede 2006). مزیت این میان افزارها آن است که پیچیدگی های سیستم های مبدا و مقصد را مخفی کرده و به توسعه دهندگان امکان می دهد، به جای تمرکز روی رابط های سطح پایین، بیشتر روی اشتراک اطلاعات متمرکز شوند. قابلیت اتصال این میان افزارها، آنها را راه حلی ایده آل برای انتقال اطلاعات بین برنامه های کاربردی می کند (Erasala et al. 2003). البته روش نقطه به نقطه، تنها اجازه اتصال یک به یک را می دهد و معمولاً از تراکنش های مشابه که درون همان محیط IT رخ می دهد، اطلاعی ندارد. علاوه بر این، میان افزارهای نقطه به نقطه، اساساً وابسته به فناوری و برنامه کاربردی

¹ Interface
² Remote Function Call
³ Accelerator products
⁴ Message Oriented Middleware

می باشد و میدان دید آن یا اصلاً شامل فرایندهای کسب و کار نیست، یا دید آن نسبت به فرایندها بسیار محدود است. (Anonymous 2002, Linthicum 2000a, Linthicum 2000a, Erasala et al.).
 بزرگترین مشکل یکپارچه سازی P2P، در شکل 4-1 نشان داده شده است. با افزایش تعداد مولفه هایی که از طریق رابط با سایرین ارتباط برقرار می کنند، تعداد رابط هایی که باید نگهداری شوند، به سرعت افزایش یافته و منجر به افزایش هزینه کل مالکیت (TCO) می شود (Gulledge 2006). هر تلاشی برای پیوند سیستم های موروثی از طریق این میان افزار، به سرعت منجر به یک کلاف پیچیده از پیوندها، بدون کنترل مرکزی و با مدیریت ناچیز می شود (Linthicum 2000a, Erasala et al. 2003). با توجه به تمام دلایلی که ذکر شد، تا جای ممکن باید از این نوع از یکپارچه سازی دوری کرد، مگر اینکه هیچ گزینه دیگری موجود نباشد (Gulledge 2006) و یا یک راه حل ساده و ابتدایی مورد نیاز باشد.



شکل 4-1- مثالی از یکپارچه سازی نقطه به نقطه

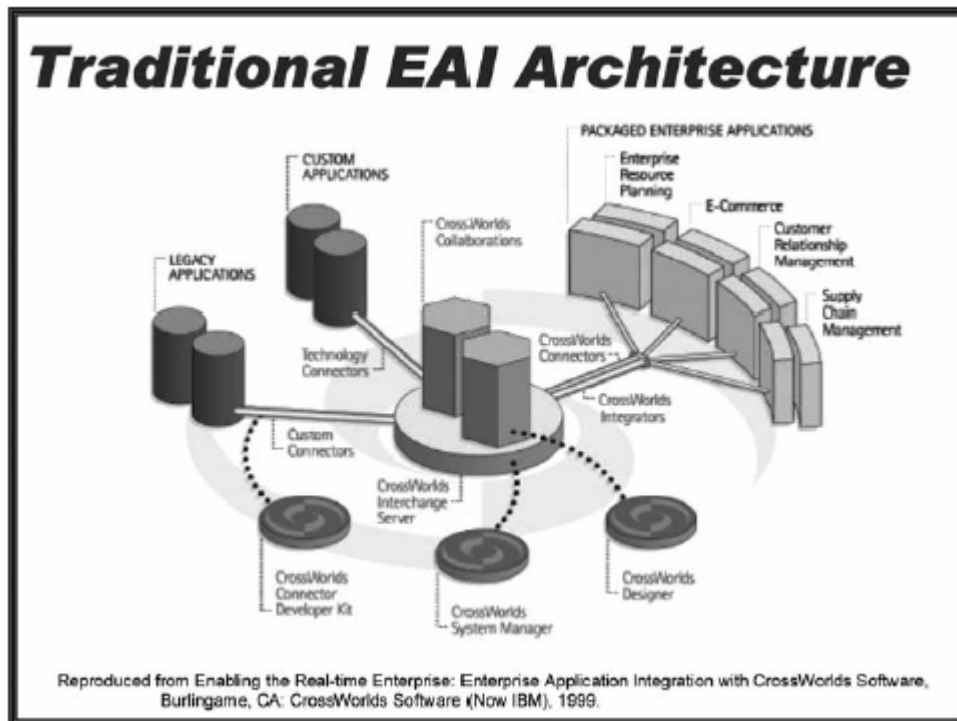
3-4- محصولات یکپارچه سازی برنامه کاربردی سازمانی (EAI)

وظیفه EAI، اشتراک داده ها و منطق فرایند کسب و کار بین نمونه های نامتجانس/متجانس از طریق میان افزار پیغام گرا (MOM) است. محصولات EAI ممکن است به وسیله فروشندگان نرم افزارهای بسته بندی شده (مانند SAP و Oracle) یا توسط فروشندگان شخص ثالث (مانند IBM، WebMethod و غیره) ارائه شوند. از محولات EAI برای اتصال چندین سیستم در سطوح برنامه کاربردی یا پایگاه داده، با استفاده از میان افزارهایی که گاهی اوقات دلال^۱ نامیده می شود، استفاده می شود (Gulledge 2006). می توان گفت که قلب هر محصول EAI، دلال آن است که به آن، موتور EAI نیز گفته می شود و معمولاً بر مبنای فناوری دلال پیغام^۲ یا میان افزار پیغام گرا ساخته می شود. محصولات EAI با بهره بردن از آداپترها یا کانکتورهای از قبل طراحی شده، اطلاعات را به سیستم ها وارد کرده یا از یک سیستم به سایرین صادر می کنند. آداپترها/کانکتورها، یکی از مزایای رقابتی ارائه دهندگان نرم افزارهای EAI هستند، چون اگر از قبل، یک کانکتور برای برنامه کاربردی مبدا یا مقصد موجود باشد، هزینه توسعه رابط کاهش خواهد یافت. EAI این امکان را فراهم می کند تا به جای اینکه برای اتصال برنامه های کاربردی به همدیگر، به صورت دو به دو رابط نوشته شود، بین هر برنامه کاربردی و زیرساخت EAI، تنها یک رابط نوشته شود. در شکل 4-2، مثالی از یک محصول EAI با معماری هاب و اسپیک نشان داده شده است (Gulledge 2006).

در این محصول، اطلاعات بعد از استخراج از سیستم مبدا، با کمک یک نوع سیستم پیغام رسانی به سرویس دهنده مرکزی ارسال می شود، در اینجا اطلاعات پردازش شده و به سمت سیستم مقصد هدایت^۳ می شود. اگر منطق مورد نیاز برای اجرای فرایند کسب و کار دچار کاستی باشد، منطق مورد نیاز را می توان بر روی سرویس دهنده هاب ایجاد کرد. در تئوری، مبادله سند بین هر دو سیستمی امکان دارد، البته اگر منطق فرایند کسب و کار سیستمهای مبدا و مقصد از این تبادل پشتیبانی کند (Gulledge 2006).

Broker¹
Message Broker²
Route³

نرم افزار EAI، با استفاده از کانکتورها و یک موتور صف بندی¹، پیغام های برنامه های کاربردی بسته بندی شده²، پایگاه داده ها، و برنامه های کاربردی سفارشی را پردازش می کند. هنگامی که یک رویداد رخ می دهد (مثلاً یک تراکنش در یک بسته ERP یا به روزرسانی جدول در یک پایگاه داده)، یک پیغام درباره این رویداد به صف موجود در هاب ارسال می شود. سیستم هایی که مشترک³ این رویداد در صف هستند، به پاکت رویداد⁴ دسترسی داشته و محتوای آنرا تحلیل می کنند، سپس اگر قرار باشد سیستم مقصد روی آن پردازشی انجام دهد، هر آنچه برای بازسازی رویداد در سیستم مقصد مورد نیاز است، مجدداً به آن اضافه می شود. موتور صف بندی تضمین می کند که تمام رویدادها به ترتیب درست پردازش شده اند، بنابراین جامعیت تراکنشی⁵ تضمین می شود (Gulledge 2006).



شکل 4-2- یک محصول EAI با معماری هاب و اسپیک

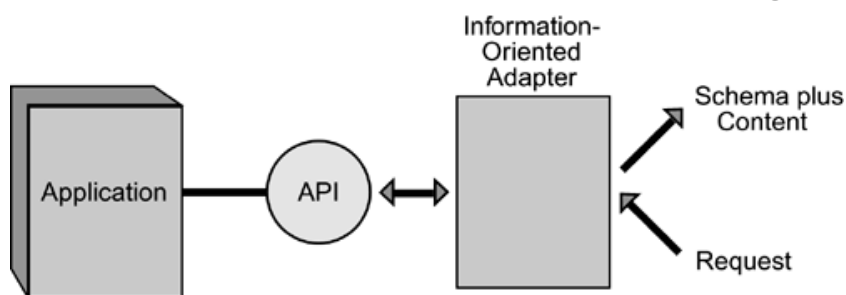
- ¹ Queuing engine
- ² Packaged application
- ³ Subscriber systems
- ⁴ Event envelope
- ⁵ Transaction integrity

شرکت های بسیاری، راه حل های EAI از پیش بسته بندی شده ای¹ را ارائه می دهند و بازار آن به شدت رقابتی است. مدل هاب و اسپیک با استفاده از کانکتورها، سالهاست که عملیاتی شده و محصولات آن به سطح بلوغ رسیده اند. با این وجود، باید بگوئیم که محصولات EAI هنوز بر اساس رابط هستند و در حالی که نسبت به یکپارچه سازی نقطه به نقطه بهبود قابل توجهی رخ داده، اما پیاده سازی و نگهداری EAI، می تواند پر هزینه باشد. اصلی ترین مزیت این راه حل ها، این است که به علت همکاری های صنعتی شان، قادرند از کانکتورهایی استفاده کنند که تا حدودی پیکر بندی شده اند² و رابط های گواهی شده ای³ دارند.

4-4- آداپترها

هدف آداپترها، ارائه یک لایه عایق بین API مولفه های IT و زیرساخت یکپارچه سازی است. آداپترها، امکان EAI غیرهجومی و پیکربندی با پیوند ضعیف را فراهم می کنند (Papazoglou and Heuvel 2005). با استفاده از آداپترها، پیچیدگی رابط های برنامه های کاربردی از کاربر نهایی یا حتی توسعه دهنده راه حل EAI پنهان می ماند (Linthicum 2003). (Linthicum 2003)، آداپترها را بر اساس سرویسی که ارائه می دهند، به دو دسته اطلاعات گرا و سرویس گرا تقسیم می کند که در ادامه توصیف بیشتر آنها آمده است.

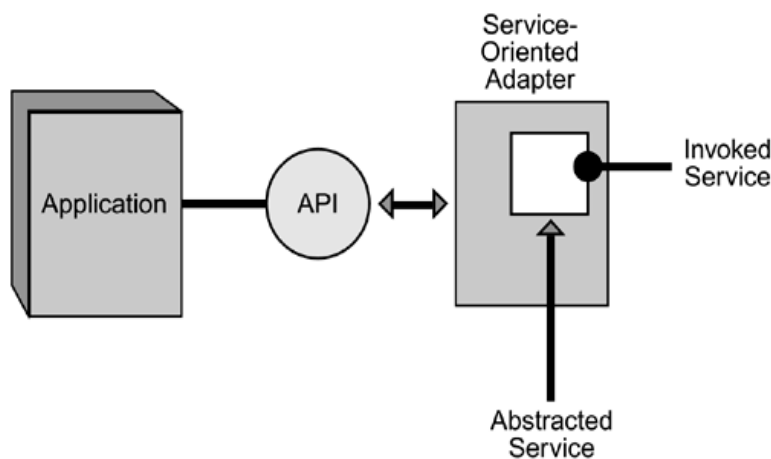
آداپترهای اطلاعات گرا صرفاً به رابط هایی نیاز دارند که بعد از رسیدن درخواست، اطلاعات را از سیستم مبدا استخراج کرده و به زیر ساخت یکپارچه سازی منتقل کنند (شکل 3-4).



شکل 3-4- آداپترهای اطلاعات گرا

¹ Pre-packaged EAI solutions
² Partially configured
³ Certified interfaces

آداپترهای سرویس گرا به جای استخراج و انتشار اطلاعات ساده به سیستم های مبدا یا مقصد، رفتار یا سرویس های برنامه کاربردی را به صورت انتزاعی در دسترس قرار می دهند (شکل 4-4). به عبارت دیگر، آنها توابع برنامه کاربردی را به شکلی در معرض نمایش می گذارند که این توابع، بتوانند درون یک برنامه کاربردی ترکیبی مانند یک تابع محلی ظاهر شوند. البته به یاد داشته باشید، با وجود اینکه، این توابع برای برنامه های کاربردی ترکیبی، همانند توابع محلی به نظر می رسند، اما پردازش در برنامه کاربردی راه دور صورت می گیرد (Linthicum 2003).

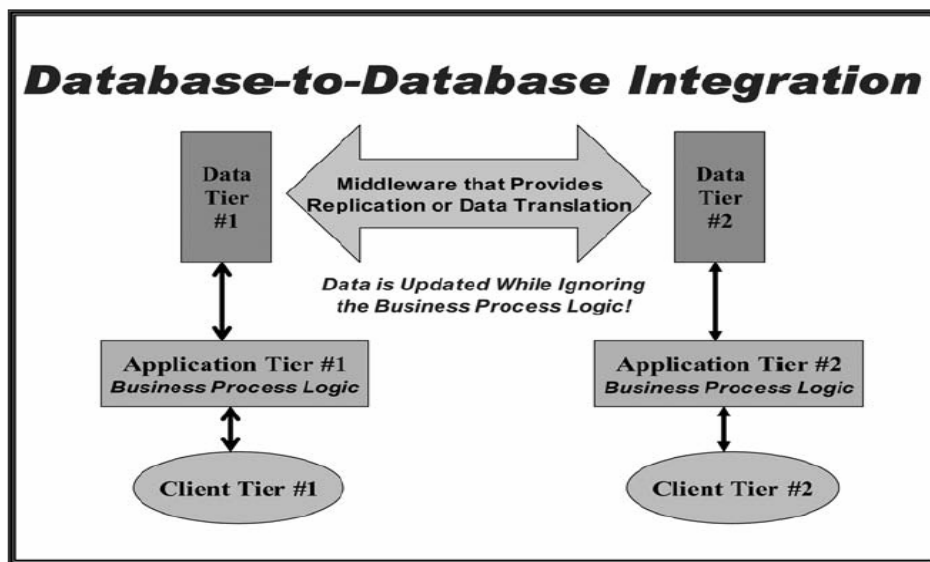


شکل 4-4- آداپترهای سرویس گرا

4-5- فناوری های یکپارچه سازی پایگاه داده با پایگاه داده

یکپارچه سازی پایگاه داده با پایگاه داده، با اشتراک اطلاعات در سطح پایگاه داده، به برنامه های کاربردی، امکان تعامل و کار کردن با هم را می دهد. راه حل ساده رونوشت¹، برای انتقال اطلاعات بین پایگاه داده ها از خصیصه هایی استفاده می کند که درون بسیاری از پایگاه داده ها وجود دارد. البته استفاده از رونوشت زمانی کاربرد دارد که تمام پایگاه داده های منبع و مقصد از شمای اطلاعاتی یکسانی استفاده کنند. شرکتهایی وجود دارند که میان افزارهایی برای شتاب دادن به این فرایند ارائه می دهند. نرم افزار پایگاه داده و رونوشت به وسیله شرکتهایی چون Pervasive's Integration Architect و DataMirro's Constellar Hub ارائه شده که اجازه انتقال اطلاعات را بین محصولات

پایگاه داده ای متفاوت با شمای متفاوت می دهد. شکل 4-5، طرح کلی مفهومی¹ این نوع از یکپارچه سازی را نشان می دهد. با وجود اینکه این رویه از یکپارچه سازی، ممکن است برای برنامه های کاربردی پایگاه داده ای به خوبی کار کند، اما برای برنامه های کاربردی سازمانی آنقدر خوب کار نمی کند. بسیاری از برنامه های کاربردی سازمانی، معماری های چند لایه ای² دارند که حتی اگر برنامه های کاربردی در یک لایه مجزا قرار داشته باشند، منطق فرایند کسب و کار به داده اصلی مقید³ شده است. بنابراین اگر انتقال اطلاعات تنها در سطح پایگاه داده انجام شود، به آسانی منجر به ایجاد مشکلات جامعیت داده ای⁴ می شود. فروشندگان نرم افزارهای سازمانی، معمولاً رابط های برنامه نویسی کاربردی⁵ (APIs) را منتشر می کنند که اجازه ارتباط را در سطح برنامه کاربردی می دهد که بهترین کار این است که از این APIها استفاده شود. اگر پایگاه داده بدون استفاده از این APIها بروز شود، آنگاه یکپارچگی سیستم اولیه، مورد تجاوز قرار گرفته و مشکلات جامعیتی یکی از نتایج احتمالی آن خواهد بود. در پیاده سازی های سازمانی، باید تا جای ممکن از این شکل از یکپارچه سازی اجتناب شود (Gulledge 2006).

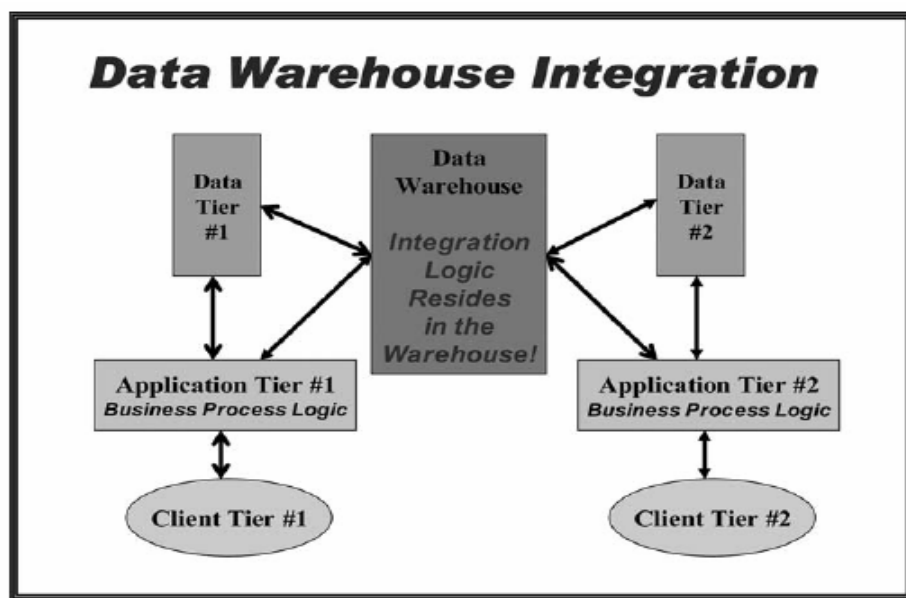


شکل 4-5- طرح کلی مفهومی یکپارچه سازی پایگاه داده با پایگاه داده

- ¹ Conceptual layout
- ² Multi-tier architecture
- ³ Bound to master data
- ⁴ Data integrity
- ⁵ Application Program Interfaces

4-6- فناوری های یکپارچه سازی با استفاده از انبار داده¹

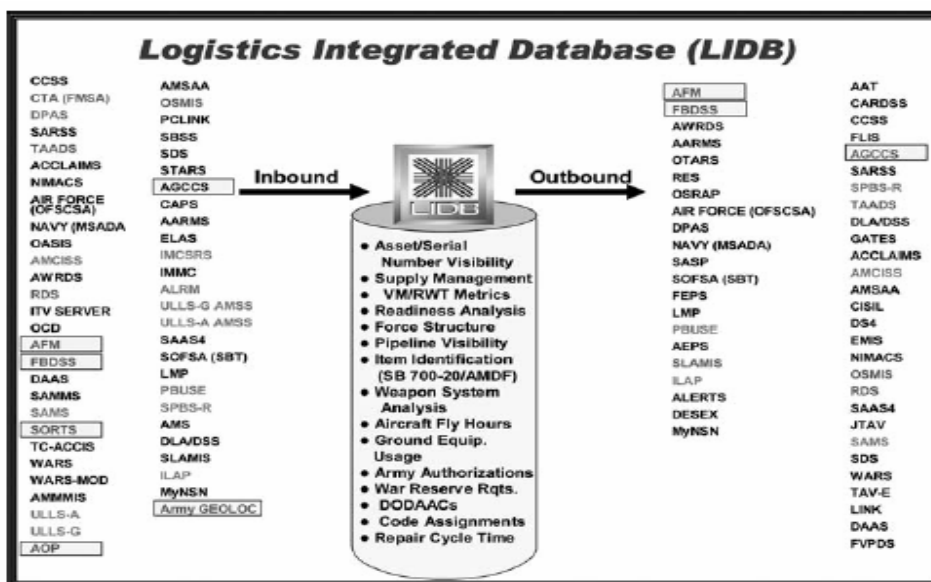
این شکل از EAI، مشابه یکپارچه سازی پایگاه داده با پایگاه داده است، اما بجای رونوشت داده بین پایگاه داده های متعدد، از یک پایگاه داده عمودی² برای نگاشت³ داده ها بین پایگاه داده های مختلف که ممکن است علامت تجاری⁴، مدل یا شمای⁵ متفاوتی داشته باشند، استفاده می کند. به عبارت دیگر، یک انبار داده جدید ایجاد شده و اطلاعاتی که ممکن است برای ایجاد گزارش، مورد تحلیل قرار گرفته یا استفاده شوند، از چند منبع مختلف، متراکم⁶ می شوند. کارایی این رویکرد⁷، به قدرت و پیچیدگی ابزارهای استفاده شده و کیفیت داده هایی که از منابع گوناگون برداشت می شود، بستگی دارد. وقتی داده ها یک بار متراکم می شوند، گزارش گیری آسان و سر راست می شود؛ اما اگر ضروری باشد که منطق کسب و کار نیز روی داده های متراکم اعمال شود، آنگاه آن منطق بایستی در سطح انبار داده ای ایجاد شود. طرح کلی یکپارچه سازی انبار داده ای در شکل 4-6 نشان داده شده است (Gulledge 2006).



شکل 4-6- نمای مفهومی یکپارچه سازی انبار داده ای

- ¹ Data warehouse
- ² Vertical database
- ³ Map
- ⁴ Brand
- ⁵ Schema
- ⁶ Aggregate
- ⁷ Approach

اگر یکپارچه سازی در سطح پایگاه داده باشد، همان مشکلات مرتبط با یکپارچه سازی پایگاه داده با پایگاه داده که در بالا ذکر شد، باز هم موجود خواهند بود. اگر یکپارچه سازی در سطح برنامه کاربردی باشد، آنگاه یکپارچه سازی انبار داده، مشابه یکپارچه سازی نقطه به نقطه شده و مشکلات آن رویکرد ظاهر می شود. این شکل از یکپارچه سازی، حتی اگر نگهدارنده گران باشد، باز هم کاملاً محبوب است. دلیلی که یکپارچه سازی انبار داده ای محبوب است، اینست که تمام دسته ها درگیر نگهداری محیط شخصی خود هستند، درحالی که داده های مورد نظر خود را در یک محیط خنثی به اشتراک می گذارند. مثالی از تلاش برای یکپارچه سازی یک انبار داده بزرگ در ارتش آمریکا، در شکل 4-7 نشان داده شده است. پایگاه داده یکپارچه لجستیک¹ (LIDB)، حاوی اطلاعات متراکم² از چندین سیستم خودمختار است، که هدفش فراهم کردن تحلیل هایی در سطح سازمان است. همانطور که شکل نشان می دهد، داده های ورودی از چندین منبع داده ای متراکم شده اند و داده های خروجی به سمت چندین منبع هل داده³ می شوند. به منظور جلوگیری از مشکلات جامعیت، تطهیر و هماهنگ سازی دائمی⁴ مورد نیاز است (Gulledge 2006).



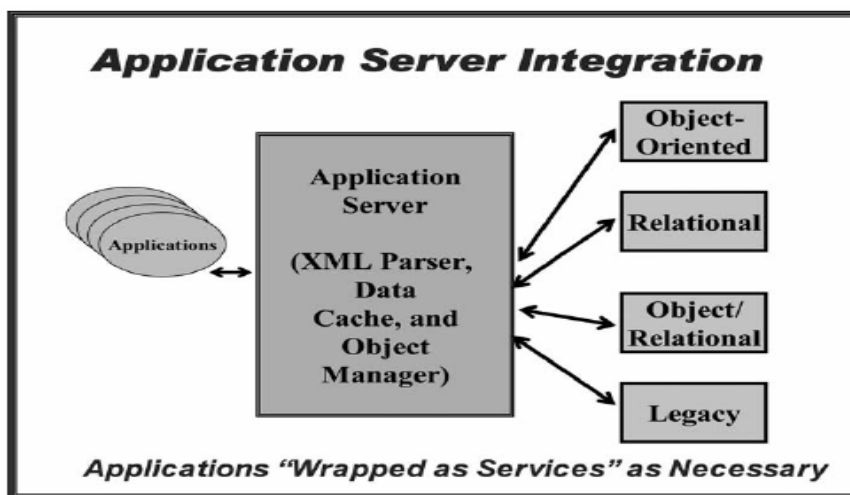
شکل 4-7- نمای مفهومی LIDB

¹ Logistics Integrated Database
² Aggregated information
³ Push
⁴ Constant cleansing and harmonization

بسیاری از راه حل های سازمانی، مانند SAP و Oracle، از راه حل های انبار داده ای برای تحلیل های سازمانی و گزارش گیری استفاده می کنند. البته همانطور که قبلاً ذکر شد، مشکل اصلی، نگهداری منطق فرایند کسب و کار در سطح انبار داده ای است. درحالیکه این گزینه، خودمختاری سازمانی را حفظ می کند، اما حقیقتاً هزینه بر است. کیفیت داده هایی که درون انبار داده، هل داده می شوند، بایستی به طور ثابت مانیتور شده و هر گونه تغییر در سیستم های مبدا و مقصد، مشکلات اساسی تست و یا کدنویسی اضافی را به همراه خواهد داشت (Gulledge 2006).

4-7- یکپارچه سازی با استفاده از سرویس دهنده برنامه کاربردی¹

این روش، یکی از پیچیده ترین روش های EAI است. یکپارچه سازی سرویس دهنده برنامه کاربردی، ایجاد یک برنامه کاربردی منفرد و متمرکز (منطقی یا فیزیکی) است که می تواند یک مجموعه مشترک از سرویس ها را برای هر تعداد از برنامه های کاربردی راه دور فراهم کند. این سرویس ها، اشیاء مشترک کسب و کار هستند که بین برنامه های کاربردی سازمانی به اشتراک گذاشته می شوند. هدف اشیاء توزیع شده² و سرویس دهنده برنامه کاربردی، اشتراک و استفاده مجدد از سرویس ها می باشد. مفهوم یکپارچه سازی سرویس دهنده برنامه کاربردی در شکل 4-8 نشان داده شده است (Gulledge 2006).



شکل 4-8- یکپارچه سازی با سرویس دهنده برنامه کاربردی

¹ Application server integration
² Distributed objects

سیستم‌های نوین، با فراخوانی اشیاء اشتراکی، منطق کسب و کار را به اشتراک گذاشته و با منابعی مانند پایگاه داده‌ها، سیستم‌های ERP یا صف‌ها تعامل دارند. در سیستم‌های ERP نوین، این اشیاء اشتراکی، ممکن است بیشتر به صورت تراکنش‌های پوشیده شده¹ در اختیار باشند. برای مثال، هنگام پیکربندی راه‌حل SAP، باید تراکنش‌ها را با گام‌های فرایندی منطبق کنید. یک گام فرایندی، می‌تواند با یک یا چند تراکنش مرتبط شود. اگر تراکنش‌هایی که با یک گام فرایندی مرتبط شده‌اند، با همدیگر تشکیل یک بسته² بدهند و در قالب یک سرویس وب پوشیده شوند³، آنگاه می‌توان آنها را بین مولفه‌های SAP و غیر SAP به اشتراک گذاشت. SAP، این شیء متراکم⁴ را " Enterprise Service" می‌نامد که بنیاد و پایه SAP's Enterprise Services Architecture می‌باشد (SAP AG 2004, Gullidge 2006).

در این روش، یکپارچه‌سازی از طریق یکپارچه‌سازی پشت‌خط⁵ برنامه‌های کاربردی و منابع متفاوت و به اشتراک گذاشتن منطق کسب و کار رخ می‌دهد. سرویس‌دهنده برنامه کاربردی، داده‌های پایگاه داده‌های رابطه‌ای یا غیر رابطه‌ای را به اشیاء عمومی اشتراکی مقید⁶ می‌کند. اصلی‌ترین مزیت یکپارچه‌سازی سرویس‌دهنده برنامه کاربردی آن است که با اشتراک متدها، پیوند قوی بین برنامه‌های کاربردی یا مولفه‌هایی که با هم ارتباط دارند، برقرار می‌شود. با توجه به اینکه منطق کسب و کار که به اشیاء مقید شده، همیشه به اشتراک گذاشته می‌شود، هرگز مشکل جامعیتی پیش نمی‌آید.

اشکال استفاده از یکپارچه‌سازی سرویس‌دهنده برنامه کاربردی که همه به آن اذعان دارند، این است که ممکن است مجبور شوید تغییرات عمده‌ای روی تمام برنامه‌های کاربردی مبدا و مقصد اعمال کنید تا آنها را در قالب یک مجموعه سرویس اشتراکی در معرض نمایش بگذارید⁷ (شاید مجبور به نوشتن لفافه‌های شیئی⁸ شوید).

¹ Wrapped transaction

² Bundle

³ Wrapped as a web service

⁴ Aggregated object

⁵ Back-end

⁶ Bind

⁷ Expose them

⁸ Object wrappers

در برنامه های کاربردی جدیدتر، سیستم ها بر مبنای یک معماری سرویس گرا (SOA) طراحی شده اند که سبب کاهش مشکلات آنها نسبت به سیستم های موروثی و قدیمی تر می شود (Gulledge 2006).

4-8- مقایسه XML و EDI

کسب و کارها، تراکنش های تجارت الکترونیک را با استفاده از استانداردهایی مانند EDI¹ یا XML²، که قالب های داده ای مورد نیاز را برای تبادل سند در چارچوب های B2B تعریف می کنند، انجام می دهند. از دهه 1970، EDI یکی از مهمترین فناوری های ارتباطات B2B بوده است. در واقع، EDI فناوری غالب برای بزرگترین شرکت ها بوده، اما کمتر از 5% کل شرکت ها آن را پذیرفته اند. کسب و کارها به دلایل متعددی مانند: هزینه نرم افزاری بالای EDI، مشکل بودن فرایند پیاده سازی آن با توجه به استاندارد جهانی واحدی که توسط EDI پشتیبانی می شود، شبکه اختصاصی گران مورد نیاز توسط EDI و غیره، معمولاً هزینه EDI را سنگین تر از مزایای آن مشاهده کرده اند (Anonymous 2002, Erasala et al. 2003). از طرف دیگر، پیاده سازی و پشتیبانی از ارتباطات ساده مبتنی بر XML نسبت به EDI، مشکلات و هزینه های کمتری دارند.

جدول 4-1- مقایسه XML و EDI

مشخصات	EDI	XML
استانداردها	ثابت	انعطاف پذیر و باز
سرعت	پردازش دسته ای	پردازش بلادرنگ ³
شبکه انتقال	شبکه های ارزش افزوده ⁴ (VAN)	شبکه اینترنت
هزینه	گران	نسبتاً ارزان تر
اندازه پیغام	فشرده	10 برابر بزرگتر
امنیت	خوب	امنیت یک مسئله است

¹ Electronic Data Interchange
² Extensible Markup Language
³ Real-time
⁴ Value-added networks

البته، اندازه پیغام های مبتنی بر XML، ممکن است در دسر ساز باشد. یک پیغام XML، احتمالاً 10 برابر بزرگ تر از یک پیغام EDI می باشد، که شاید برای شرکت های کوچک نگران کننده نباشد، اما ممکن است برای شرکت های بزرگ، یک مسئله بزرگ باشد. ارتباطات مبتنی بر XML، شاید به اندازه EDI امن نباشد. مقایسه این دو فناوری با یکدیگر در جدول 4-1 آمده است (Michel 2000,) (Erasala et al. 2003).

4-9- سرویس های وب و EAI

فناوری سرویس وب، تلفیق فناوری های (XML، SOAP¹، WSDL² و UDDI³) است که به کاربران اجازه توسعه، فهرست بندی و انتشار سرویس های کسب و کار برای تحویل و استفاده روی وب را می دهد (Venugopal, Kupper 2002, Erasala et al. 2003). سرویس های وب، با استفاده از پروتکل ها و قالب های استاندارد مبتنی بر XML، یک فناوری محاسباتی توزیع شده⁴ برای در معرض نمایش گذاشتن سرویس های کسب و کار برنامه های کاربردی روی اینترنت یا اینترنت، ارائه می دهند. استفاده از پروتکل های استاندارد مبتنی بر XML، سرویس های وب را مستقل از پلتفرم، زبان و فروشنده و یک کاندیدای ایده آل برای استفاده در راه حل های EAI می کند. سرویس های وب با به کارگیری استانداردهای اینترنتی باز، مشکلات تعامل با یکدیگر⁵ را که در راه حل های موجود، مانند CORBA و DCOM وجود دارد، حذف می کنند که این استانداردها شامل WSDL برای توصیف، UDDI برای تبلیغ و تشکیل اتحادیه دادن، SOAP برای ارتباط و WSFL⁶ برای تعریف جریان کاری⁷ می باشد (Samtani 2002, Erasala et al. 2003).

Simple Object Access Protocol 1
Web Service Description Language 2
Universal Description, Discovery and Integration 3
Distributed computing 4
Interoperability 5
Web Services Flow Language 6
7 البته WSFL هنوز یک استاندارد W3C نیست

Butler اظهار می کند که سرویس های وب، یک کاتالیزور برای EAI بوده و یک زیرساخت قابل اطمینان¹، ایمن² و سالم³ فراهم می کنند. با وجود اینکه برای یکپارچه شدن با منابع داده ای موروثی، هنوز هم به آداپترها و کانکتورهای EAI نیاز است، اما در آینده، برنامه های کاربردی که به صورت توکار⁴ از سرویس وب / SOAP / XML پشتیبانی کنند، نیاز به آداپترها و کانکتورهای شخص ثالث ناخوانده و دردسر ساز را کاهش خواهند داد. در مقایسه با فناوری EAI سنتی، سرویس های وب، ساده تر، ارزانتر، و بر اساس استانداردهای باز بوده و کارآمدی، پویایی و انعطاف بیشتری دارند (Samtani 2002, Sadhwani 2002, Erasala et al. 2003). سرویس های وب، در آینده نزدیک، یک فناوری سبک و آسان در EAI خواهند بود و یکپارچه سازی سریع و غیر پیچیده را امکان پذیر خواهند ساخت. سرویس های وب، امکان ارتباط نقطه به نقطه و با کمترین نیازمندی را برای هر شبکه خدماتی که می خواهد با دیگری کار کند، ترویج و میسر می سازند، البته بایستی هر دوی آنها از XML پشتیبانی کنند. برای پیاده سازی سرویس های وب، به هیچ سرویس دهنده پیچیده ای نیاز نیست. شما نیاز ندارید که آداپترها و کانکتورها را خریداری کنید، در ضمن به علت مکانیزم ارتباطی انعطاف پذیر آن (XML)، شاید منطق کسب و کار نیز بدون تغییر باقی بماند. علاوه بر این، برای پروژه های EAI غیر پیچیده، با توجه به سادگی سرویس های وب، هزینه های نگهداری در بلند مدت کمتر خواهد شد (Venugopal, Kupper 2002, Erasala et al. 2003). البته مشکلاتی هم در صورت پذیرش سرویس های وب گزارش شده که عبارتند از: بی تجربگی در معماری سرویس های وب، تغییر فرهنگ سازمانی داخلی برای پذیرش سرویس های وب، استانداردهای متعدد برای پیاده سازی، نرسیدن فناوری به مرحله بلوغ و نگرانی های امنیتی (Estrem 2003, Khoubati et al. 2005).

Reliable¹
Secure²
Safe³
Built-in⁴

10-4 - مقایسه EDI، EAI و سرویس های وب

Khoubati et al. مزایا و معایب این سه نوع از فناوری های یکپارچه سازی را مورد بررسی قرار داده که نتایج آن در جدول 2-4 ارائه شده است.

جدول 2-4 مزایا و معایب فناوری های یکپارچه سازی EDI، سرویس های وب و EAI

فناوری	مزایا	معایب
EDI	روابط نزدیکتر بین تامین کننده و مشتری (Soliman and Janz 2004) پاسخ سریع به همکاران تجاری (Lee et al. 2005) کاهش خطاهای اداری و دفتری (Scala and Roger 1993) اجازه بهبود کارایی و راندمان کسب و کار (Ratnasingham 1998) حذف بسیاری از وظایف نیروی انسانی (Vijayarathy and Robey 1997)	به سرمایه گذاری اولیه بالایی نیاز دارد (Soliman and Janz 2004) به سخت افزار سازگاری در هر دو طرف نیاز دارد (Soliman and Janz 2004) افزایش خطر فرایند اداره تراکنش های تجاری (Ratnasingham 1998) پیاده سازی آن ساده نیست (Ratnasingham 1998) مشکل تعیین اندازه برگشت سرمایه (ROI) (Philip and Oedersen 1997)
محصولات EAI	بهبود ارتباطات با مشتری (Themistocleous and Irani 2001) نائل شدن به برگشت سرمایه بهتر (Themistocleous and Irani 2001) افزایش کارایی کلی زنجیره تامین (Themistocleous and Irani 2004) نائل شدن به یکپارچه سازی فرایند کسب و کار (Linthicum 1999) کاهش هزینه یکپارچه سازی (Linthicum 1999)	به سطح بالایی از سرمایه گذاری نیاز دارد (Stal 2002) هیچ محصول EAI منفردی تمام مشکلات یکپارچه سازی را حل نمی کند (Themistocleous and Irani 2002) نگرانی از پیچیدگی آن (Stal 2002) به مهارت های EAI نیاز دارد (Themistocleous and Irani 2002) این فناوری ها گیج کننده می شوند (Zahavi 1999)
سرویس های وب	کاهش زمان توسعه و بکارگیری (Ratnasingam and Pavlou 2002) کاهش هزینه پیاده سازی (Huang and Chung 2003) کاهش پیچیدگی نگهداری (Huang and	نیاز به سفارشی سازی بالایی دارد (Wu and Sawy 2003) عدم پشتیبانی از تراکنش (Huang and Chung 2003) از لحاظ کارایی به بلوغ نرسیده است (Wu

(and Sawy 2003) فقدان رابط کاربری (Estrem 2003) امنیت یک مشکل است (Stal 2002)	(Chung 2003) به مهارت های برنامه نویسی کمتری نیاز دارد (Huang and Chung 2003) خطر شکست پروژه کمتر است (Wu and Sawy 2003)	
---	--	--

Khoumbati et al. سپس یک تجزیه و تحلیل مقایسه ای از این فناوری ها ارائه می کنند. این تحلیل بر اساس چندین فاکتور یکپارچه سازی مانند قابلیت نگهداری، انعطاف پذیری، مقیاس پذیری، قابلیت حمل، قابلیت استفاده مجدد، بلوغ، پیچیدگی، غیر هجومی بودن، کارایی، بلادرنگ بودن، امنیت، تراکنش و سطوح یکپارچه سازی می باشد. این مقایسه تحلیلی به همراه تشریح هر کدام از این فاکتورها، در جدول 3-4 آمده است. این تحلیل، بطور واضح نشان می دهد که EAI یک فناوری است که از تمام این فاکتورها پشتیبانی می کند. (Khoumbati et al. 2005).

جدول 3-4 - مقایسه فناوری های یکپارچه سازی EDI، سرویس های وب و EAI

فناوری ها			توصیف	مشخصات
WS	EDI	EAI		
ن		ن	اجازه تغییر، بدون ایجاد مشکل برای سایر مولفه ها یا سیستم ها.	قابلیت نگهداری
		ن	انعطاف پذیری در تغییر به معنی سازگاری سریع با حداقل تلاش و همچنین انعطاف پذیری در انجام وظایف به معنی خوب عمل کردن در محیط های متفاوت.	انعطاف پذیری
ن	ن	ن	قابلیت یک سیستم اطلاعاتی برای ارائه کارایی بالاتر در هنگام افزایش تقاضا و نیاز به قدرت محاسباتی اضافه تر.	مقیاس پذیری
ن		ن	راه حل نرم افزاری که برای یک پلتفرم توسعه داده شده، باید بتواند روی یک پلتفرم کاملاً متفاوت اجرا شود.	قابلیت حمل
ن	ن	ن	قابلیت استفاده از مولفه ها یا راه حل های نرم افزاری موجود برای ساخت برنامه های کاربردی جدید.	قابلیت استفاده مجدد
	ن	ن	اینکه آیا یک فناوری یکپارچه سازی به بلوغ رسیده است یا نه. واضح است که یک فناوری هر چه بیشتر به بلوغ نزدیک شده باشد، بهتر است.	بلوغ
ن		ن	اینکه یک فناوری یکپارچه سازی منجر به یک راه حل پیچیده یا ساده شود.	پیچیدگی

			واضح است که راه‌حل‌های پیچیده ترجیح داده نمی‌شوند.	
ن	ن	ن	هر چه یک راه حل، غیر هجومی تر باشد، اضافه کردن ماژول‌های جدید، نیاز به تغییرات کمتری دارد و هزینه و تلاش لازم برای یکپارچه سازی کاهش یافته و پیچیدگی‌ها حذف شده و انعطاف و قابلیت نگهداری آن افزایش می‌یابد.	غیر هجومی بودن
		ن	اینکه کارایی کل راه حل، پایین باشد یا نه.	کارایی
ن	ن	ن	پشتیبانی از تراکنش‌ها با حداقل تاخیر داده‌ای در جریان اطلاعاتی مورد نیاز	بلادرنگ بودن
	ن	ن	اجتناب از مداخله در ارسال داده ¹ ، تقلب، دسترسی نامناسب، افشای اطلاعات حساس و وقفه در انجام عملیات بحرانی	امنیت
	ن	ن	منظور از تراکنش، دسترسی همزمان به داده‌های به اشتراک گذاشته شده بین چندین مولفه برای انجام عملیاتی روی آن داده‌ها است. علاوه بر آن، تراکنش به جای اشتراک محض اطلاعات و دانش، به فعالیت‌های تجاری مانند سفارش دادن، صدور فاکتور و غیره نیز، اشاره دارد.	تراکنش
		ن	Linthicum (1999) اظهار می‌کند که یکپارچه سازی می‌تواند در سطوح متفاوتی (مانند سطح داده‌ای، سطح شیئی و سطح فرایندی) حاصل شود.	سطوح یکپارچه سازی

4-11- مقایسه معماری سرویس‌گرا و EAI

بسیاری از افراد حرفه‌ای در زمینه یکپارچه سازی، معماری سرویس‌گرا را به عنوان نسل بعدی فناوری‌های EAI در نظر گرفته‌اند. فناوری‌های EAI سنتی، برای یکپارچه سازی برنامه‌های کاربردی که به صورت جداگانه توسعه داده شده‌اند، به خوبی کار می‌کنند. اما باید توجه داشت که مبنای آن روش‌ها، راه‌حل‌های اختصاصی فروشندگان EAI است که سازمان را به پلتفرم‌های آن فروشندگان وابسته می‌کند. امروزه با فراگیر شدن سرویس‌های وب، به عنوان فناوری مناسب برای انتقال، معماری سرویس‌گرا و سرویس‌های وب به عنوان راه‌حل‌های یکپارچه سازی مبتنی بر استاندارد مطرح شده‌اند. بنابراین باید گفت که مستقل از پلتفرم بودن سرویس‌های وب، سبب شده تا استفاده از آنها به جای فناوری‌های EAI متعارف، بی‌نهایت جذاب شود. علاوه بر این، معرفی

1 Tampering: منظور این است که در هنگام انتقال، اطلاعات تغییر داده شده و یا جایگزین شده و سپس به گیرنده ارسال شوند.

محصولات باس سرویس سازمانی^۱ (ESB)، یکپارچه سازی استاندارد و مبتنی بر سرویس وب را فوق العاده مشهور و متداول ساخته است. نکته دیگر این است که اگر چه مقصد نهایی EAI و معماری سرویس گرا، پیاده سازی فرایندهای کسب و کار بر اساس سبب^۲ برنامه های کاربردی موجود است، اما هر کدام از آنها برای رسیدن به این هدف، از روشی استفاده می کنند که تفاوت های کاملاً ریشه ای دارد. EAI روی در معرض نمایش گذاشتن عملکرد^۳ برنامه های کاربردی موجود، تمرکز دارد که سبب می شود، مدل نهایی بر اساس سبب برنامه های کاربردی موجود شکل بگیرد. اما در مقابل، SOA روی پنهان سازی برنامه های کاربردی موجود و در عوض، در معرض نمایش گذاشتن مجموعه سرویس ها به صورت مستقل از برنامه کاربردی تمرکز دارد (Lublinsky et al. 2006). در ادامه، محصولات ESB و کاربرد آن در SOA، با توصیف بیشتری مورد بررسی قرار می گیرند.

ESB از نظر مفهومی، تکامل یافته مکانیزم ذخیره و ارسال^۴ محصولات میان افزار مانند میان افزار پیغام گرا (MOM) است و اکنون فناوری های متعارف EAI را با فناوری های هماهنگ و موزون سازی^۵، سرویس های وب، XML و XSLT ترکیب کرده است. ESB امکان طراحی، توسعه، استقرار و مدیریت SOA را فراهم کرده و چالش های هماهنگ و موزون سازی سرویس ها، همگام سازی داده ای^۶ و مانیتورینگ فعالیت های کسب و کار^۷ (BAM) را بر طرف می کند. از نظر فیزیکی، ESB ستون فقرات مورد نیاز برای پیاده سازی SOA را ارائه می کند. ESB به برنامه های کاربردی و مولفه های یکپارچه سازی گسسته، این امکان را می دهد تا با تعامل با همدیگر، مجموعه ای از سرویس ها را برای تشکیل فرایندهای کسب و کار ترکیبی ایجاد کرده و به این ترتیب، فرایندهای کسب و کار سازمان را خودکار کنند (Papazoglou and Heuvel 2005).

Enterprise Service Bus¹

Portfolio²

Expose Functionality³

Store-and-forward⁴

Orchestration⁵

Data synchronization⁶

Business Activity Monitoring⁷

4-12- چارچوبی جهت ارزیابی فناوری های یکپارچه سازی

Themistocleous و دیگران (2004)، بعد از مرور گسترده ادبیات، یک چارچوب جدید برای ارزیابی فناوری های یکپارچه سازی ارائه کرده اند که در ادامه مورد بررسی قرار می گیرد. سازمان ها برای ارزیابی فناوری های EAI، می توانند از معیارهای ارزیابی پیشنهادی که در جدول 4-4 آمده، استفاده کنند. در این جدول، معیارهای ارزیابی فناوری های EAI بر اساس عناصر برنامه کاربردی، انواع سیستم ها و لایه های یکپارچه سازی، آمده است (Themistocleous et al. 2004).

جدول 4-4- معیارهای ارزیابی پیشنهادی

معیارهای ارزیابی		
عناصر برنامه کاربردی	لایه های یکپارچه سازی	دسته بندی انواع سیستم ها
<ul style="list-style-type: none"> • داده ها • اشیاء • فرایندها 	<ul style="list-style-type: none"> • لایه انتقال • لایه تبدیل • لایه اتوماسیون • فرایند 	<ul style="list-style-type: none"> • سفارشی با سفارشی • سفارشی با بسته بندی شده • سفارشی با کسب و کار الکترونیک • بسته بندی شده با بسته بندی شده • بسته بندی شده با کسب و کار الکترونیک • کسب و کار الکترونیک با کسب و کار الکترونیک • سفارشی با بسته بندی شده با کسب و کار الکترونیک

در این چارچوب، معماری فناوری های EAI، شامل سه لایه لایه انتقال داده، تبدیل داده و اتوماسیون فرایند در نظر گرفته شده است. همانطور که مشاهده می شود، در این چارچوب، لایه مسیریابی پیغام و لایه اتصال که در فصل سوم به عنوان لایه های یکپارچه سازی معرفی شد، وجود ندارد. در این چارچوب، داده ها، اشیاء و فرایندها به عنوان عناصر برنامه های کاربردی در نظر گرفته شده و گفته شده که این عناصر را باید از برنامه های کاربردی مبدا استخراج کرده و با کمک سه لایه یکپارچه سازی گفته شده، قبل از رسیدن به برنامه کاربردی مقصد، به قالب مورد نظر سیستم مقصد، تبدیل کرد (Themistocleous et al. 2004).

همانطور که از جدول 4-4 بر می آید، انواع سیستم های نرم افزاری که ممکن است با هم یکپارچه شوند، یک تشکیل جایگشت از سیستم های نرم افزاری می دهد که توضیح کامل هر کدام از این ها نیز در جدول 2-3 آمده است.

همانطور که در جدول 4-5 مشاهده می شود، Themistocleous و دیگران (2004)، بر اساس معیارهای ارزیابی جدول بالا، فناوری های مختلف EAI را ارزیابی کرده و بر اساس عملکرد هر یک از این فناوری ها، آنها را رتبه بندی کرده اند. برآورد و رتبه بندی فناوری های یکپارچه سازی که در این جدول آمده، بر اساس شواهد و مدارکی است که از مرور گسترده ادبیات حاصل شده است. معنای علامت های استفاده شده در رتبه بندی فناوری های یکپارچه سازی، در شکل 4-9 آمده است.

علامت	توضیح
○	پشتیبانی کم
◐	پشتیبانی متوسط
●	پشتیبانی بالا
✓	این فناوری از یکپارچه سازی آن دسته بندی، پشتیبانی می کند
✗	این فناوری از یکپارچه سازی آن دسته بندی، پشتیبانی نمی کند
—	درباره آن، اطلاعات وجود ندارد

شکل 4-9- معنای علامت های استفاده شده در رتبه بندی فناوری های یکپارچه سازی

4-12-1- ارزیابی چارچوب پیشنهادی ارائه شده

جهت اثبات اعتبار کار انجام شده، با استفاده از مصاحبه ساخت یافته با تعدادی از افراد و همچنین مصاحبه غیر ساخته (مطالعه موردی¹) با سایر افرادی که درگیر ارزیابی و پیاده سازی EAI بوده اند، چارچوب ارائه شده، مورد ارزیابی قرار گرفته است (Themistocleous et al. 2004). نتایج این ارزیابی به صورت زیر است.

- مجموعه تمام معیارها، برای ارزیابی فناوری های یکپارچه سازی بسیار مهم است. سازمان ها مجبورند، تمام این معیارها و فناوری های ارزیابی را لحاظ کنند.

¹ Case study

جدول 4-5- ارزیابی فناوری های یکپارچه سازی

دسته بندی انواع سیستم ها							معیارهای ارزیابی						فناوری های یکپارچه سازی	دسته بندی فناوری های یکپارچه سازی
							لایه های یکپارچه سازی			عناصر برنامه های کاربردی				
سفارشی با بسته بندی شده با کسب و کار الکترونیک	کسب و کار الکترونیک با کسب و کار الکترونیک	بسته بندی شده با کسب و کار الکترونیک	بسته بندی شده با بسته بندی شده	سفارشی با کسب و کار الکترونیک	سفارشی با بسته بندی شده	سفارشی با سفارشی	انتقال	بند	راه های فرایند	فرایندها	آپدیت	داده ها		
●	●	●	●	◐	◐	◐	×	✓	×	×	×	●	ODBC	میان افزارهای پایگاه داده گرا
◐	●	◐	◐	◐	◐	○	×	✓	×	×	×	●	JDBC	
○	×	×	×	○	○	○	×	×	●	×	×	✓	RPC	فناوری های پیام گرا
○	×	○	○	◐	◐	◐	×	×	●	×	○	✓	MOM	
●	●	●	●	●	●	●	●	●	●	✓	○	✓	Message Broker	
●	●	●	●	●	○	○	×	●	●	×	✓	✓	XML	
○	○	○	○	○	○	◐	×	○	✓	×	×	✓	TPM	فناوری های تراکنش گرا
◐	●	●	◐	◐	◐	×	×	×	✓	×	✓	✓	Application servers	
◐	●	●	●	◐	◐	◐	-	✓	✓	-	●	✓	CORBA	فناوری های اشیاء توزیع شده
◐	●	●	●	◐	◐	○	-	-	✓	-	●	✓	DCOM/COM	
	◐	●	◐	◐	○	○	-	-	✓	-	●	✓	EJB	
●	○	○	○	●	●	●	×	✓	×	×	✓	✓	Screen wrapper	فناوری های اینترنت گرا
●	●	●	●	×	◐	-	-	✓	-	-	✓	✓	APIs	
●	●	●	●	◐	◐	-	×	✓	×	×	✓	✓	Adapters	

- در مورد معیار عناصر برنامه کاربردی، هیچ فناوری منفردی وجود ندارد که از یکپارچه سازی تمام عناصر برنامه کاربردی پشتیبانی کند، در ضمن، شما به ترکیبی از فناوری ها، برای تسهیل یکپارچه سازی داده ای، شیئی و فرایندی نیاز دارید.
- هرچند تعدادی از این فناوری ها، مانند دلال های پیغام¹ و آداپترها، از سایر راه حل ها، قویتر هستند، اما باز هم تعیین بهترین فناوری با استفاده از این جدول مشکل است. در ابتدا، باید کاربرد هر فناوری درک شود که این کار با بررسی لایه های یکپارچه سازی آن امکان پذیر است. با استفاده از این لایه ها می توان تشخیص داد که چه فناوری هایی از چه لایه هایی پشتیبانی می کنند. در هر لایه، باید به دنبال فناوری هایی گشت که از تمام عناصر برنامه های کاربردی پشتیبانی می کند.
- مصاحبه شوندگان علاوه بر لایه های فوق الذکر، لایه اتصال² را نیز در نظر گرفته اند. این لایه، مسئول ایجاد رابط های اتصال³ بین برنامه های کاربردی و زیرساخت مرکزی یکپارچه سازی است. از طریق این اتصالات، عناصر برنامه کاربردی از هر کدام از سیستم ها به لایه انتقال ارسال می شوند. لایه انتقال، این عناصر را به زیرساخت مرکزی یکپارچه سازی منتقل می کند، در این جاست که عملیات تبدیل و اتوماسیون فرایند رخ می دهد.
- در مورد معیار لایه های یکپارچه سازی، مصاحبه شوندگان گزارش کرده اند، با وجود اینکه دلال های پیغام می توانند از لایه انتقال پشتیبانی کنند، اما در عمل، برای پوشش این لایه، از دلال پیغام استفاده نمی شود، یعنی توسعه دهندگان ترجیح می دهند، از دلال های پیغام صرفاً برای لایه تبدیل و اتوماسیون فرایند استفاده کرده و از سایر فناوری ها برای لایه انتقال استفاده کنند.
- در مورد معیار انواع سیستم ها، مشخص است که دلال های پیغام، از تمام این هفت نوع سیستم، پشتیبانی می کنند. بنابراین فروشندگان محصولات EAI که از دلال های پیغام، به عنوان موتور اصلی یکپارچه سازی راه حل های خود، استفاده می کنند، برای پشتیبانی از هر هفت نوع سیستم مناسب هستند. آداپترها و XML نیز، تقریباً از تمام آنها پشتیبانی می کنند (Themistocleous et al. 2004).

Message Broker¹
Connectivity²
Connection-interfaces³

4-13- نتیجه گیری

در این فصل، تعدادی از فناوری ها و راه حل های توانمند ساز مطرح در EAI توصیف و با یکدیگر مقایسه شد، در ادامه، چارچوبی جهت ارزیابی فناوری های مختلف ارائه شد و در نهایت، تعداد زیادی از فناوری های مطرح در EAI با استفاده از این چارچوب مورد ارزیابی قرار گرفتند. اولین نتیجه ای که از این فصل، می توان گرفت، این است که هیچ فناوری وجود منفردی ندارد که از یکپارچه سازی تمام عناصر برنامه های کاربردی پشتیبانی کند، البته در این میان، دلال های پیغام که بسیاری از فروشندگان راه حل های EAI، از آن به عنوان موتور محصولات خود استفاده می کنند، از سایرین قویتر بوده و از یکپارچه سازی انواع بیشتری از سیستم ها پشتیبانی می کنند. آداپترها و XML نیز، جزء فناوری هایی هستند که کاربرد خوبی داشته و البته اکثر ارائه دهندگان محصولات EAI نیز از آنها پشتیبانی می کنند. نتیجه دیگری که می توان از این فصل گرفت، این است که با ظهور سرویس های وب و معماری های سرویس گرا، محصولات EAI، جای خود را به محصولات ESB و معماری سرویس گرا خواهند داد و نیاز به آداپترها و همچنین سایر روش های پیچیده نیز کمتر خواهد شد.

فصل پنجم

نتیجه گیری

5-1- مقدمه

EAI، مبحث تحقیقاتی جدیدی است که تحقیقات در آن، هنوز در مرحله رشد بوده و به بلوغ نرسیده است، بنابراین ادبیات آن هنوز محدود است و به کار جدی نیاز دارد. بر این اساس، واضح است که از نظر علمی، تحقیق در این زمینه، بسیار جذاب می باشد. نکته دیگر، کاربردی بودن تحقیقات در این زمینه می باشد. علت این امر، آن است که EAI، نیازی جدی و واقعی و چالش پیش روی بسیاری از شرکت های امروزی می باشد که با گذشت زمان، نه تنها رفع نخواهد شد، بلکه برای رسیدن به یک جامعه اطلاعاتی یکپارچه، روز به روز، فشارها برای استفاده از آن تشدید می شود. بر این اساس، کار روی مباحث مربوط به EAI، هم از جنبه علمی و هم از جنبه تجاری، بسیار مناسب می باشد. سوالی که ممکن است مطرح شود، این است که در حال حاضر، چه زمینه های تحقیقاتی در EAI مطرح است و جهت گیری آینده EAI به چه سمتی است که این همان چیزی است که در این فصل، سعی شده به آن پاسخ داده شود.

در این فصل، بعد از بررسی مباحث تحقیقاتی مناسب در زمینه EAI که الان روی آنها کار می شود، به محدودیت های اصلی تحقیق در زمینه EAI اشاره شده، سپس به بررسی مباحث تحقیقاتی که در راستای EAI مطرح هستند و ممکن است، جایگزین راه حل های EAI شوند، پرداخته شده و در نهایت، جهت گیری آینده و تعدادی از مسائل تحقیقاتی باز در زمینه EAI مطرح شده است.

5-2- مباحث تحقیقاتی مطرح در زمینه EAI

اکثر افرادی که در زمینه EAI، رویکرد، معماری، چارچوب یا مدل جدیدی کار کرده اند، یک کار بنیادین انجام داده اند و مدل های ارائه شده آنها هنوز اثبات نشده است. با توجه به نو بودن و فنی بودن مباحث EAI، هنوز جا برای انجام کارهای بنیادین و ارائه مدل ها و چارچوب های جدید در این زمینه وجود دارد.

پژوهشگرانی که صرفاً به دنبال اثبات، اصلاح و یا رد مدل ها و چارچوب ها فوق الذکر هستند، می توانند با انجام مطالعات پیمایشی و یا مطالعات موردی، به تعدیل یا رد این مدل ها بپردازند. بر این اساس، اگر قرار باشد، یک چارچوب کلی برای تحقیقات این چنینی در زمینه EAI ارائه شود، می توان به چارچوب ارزیابی فاکتورهای اثرگذار روی پذیرش EAI اشاره کرد که توسط Themistocleous (2004) ارائه شده است (شکل 5-1). این چارچوب، بر اساس ده فاکتور مهم که روی پذیرش EAI تاثیر جدی دارند، بنا شده که عبارتند از: هزینه ها، موانع، مزایا، فشارهای داخلی بر شرکت، فشارهای خارجی بر شرکت، پیچیدگی IT، زیرساخت IT، پشتیبانی، وجود چارچوبی جهت ارزیابی فناوری های EAI و چارچوبی جهت ارزیابی بسته های EAI.

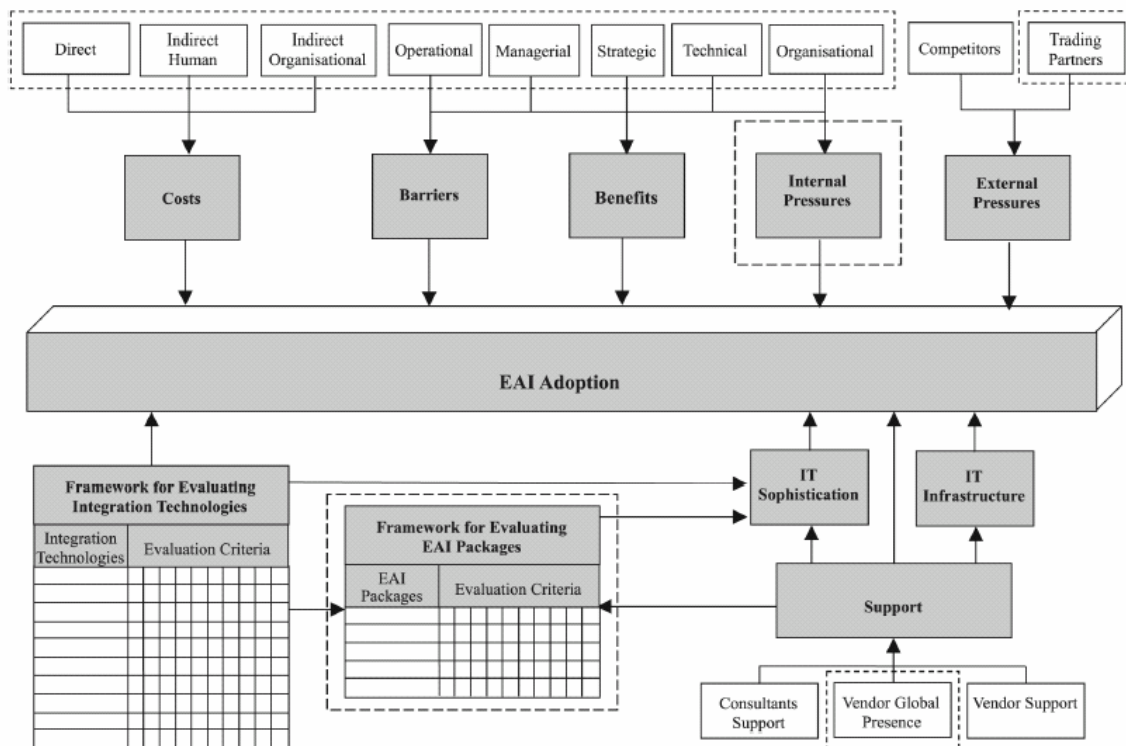
مزایای پذیرش EAI را می توان به مزایای عملیاتی، مدیریتی، فنی، راهبردی و سازمانی تقسیم کرد. همین دسته بندی برای موانع پذیرش EAI نیز کاربرد دارد. هزینه ها خود شامل هزینه های مستقیم، هزینه های انسانی غیر مستقیم و هزینه های سازمانی غیر مستقیم می باشند.

کسانی که قصد تحقیق در این زمینه را دارند، می توانند با بهره گیری از این مدل، روی پذیرش EAI با استفاده از این ده فاکتور کار کرده و با انجام مطالعات موردی یا پیمایشی، اعتبار فاکتورهای این چارچوب را ارزیابی و وزن دهی کرده و یا آن را اصلاح کنند. علاوه بر این، با استفاده از این چارچوب و مرور ادبیات EAI، می توان فاکتورهای حیاتی موفقیت و شکست پروژه های EAI را شناسایی کرده و یا اثبات کرد.

از طرف دیگر، با توجه به اینکه EAI، درگیر مسائل گردش کار و فرایندی می شود، مباحث مدل سازی و اثبات نمودارهای گردش کار ایجاد شده توسط راه حل های EAI با استفاده از روش هایی مانند شبکه های پتری¹، می تواند موضوع مناسب دیگری برای تحقیق باشد.

¹ Petri nets

همانطور که در فصل قبلی مشاهده شد، در زمینه فناوری های یکپارچه سازی، کارهایی انجام شده، اما با تکامل این فناوری ها و معرفی فناوری های جدیدتر، چارچوب های ارائه شده، بایستی مورد بازنگری قرار گرفته و فناوری های جدیدتر نیز در آن لحاظ شود. همچنین با معرفی معماری ها و سیستم های نرم افزاری جدیدتر، رویکردهای ارائه شده در فصل دوم نیز، بایستی مورد بازنگری و اصلاح قرار گیرند. تطبیق فناوری ها و راه حل های معروف یکپارچه سازی با رویکردهای مختلف ارائه شده در فصل دوم نیز، به کار مطالعاتی نیاز دارد.



شکل 5-1- چارچوبی جهت ارزیابی پذیرش EAI

5-3- محدودیت اصلی تحقیق در زمینه EAI

اعتبارسنجی مدل ها و چارچوب های ارائه شده در زمینه EAI، معمولاً بایستی به کمک مطالعه موردی¹ یا مطالعه پیمایشی² انجام گردد که برای این کار نیز به اطلاعاتی در مورد پروژه های پیاده شده EAI نیاز است. اما مشکل اساسی، جمع آوری این اطلاعات می باشد. با توجه به اینکه پروژه های EAI در سطح

¹ Case study
² Survey study

یک سازمان بزرگ¹ معنا پیدا می کند و روی یکپارچه سازی سیستم‌هایی مانند ERP کار می کند و از طرف دیگر، شرکت هایی که این پروژه های را اجرا کرده اند، محدود هستند، برای اثبات اعتبار پژوهش‌های انجام شده، با کمبود مورد مطالعاتی مواجه هستیم. بر این اساس، در حال حاضر، بحث اثبات کار با استفاده از مطالعات پیمایشی منتفی است، اما اگر قرار باشد، اعتبار کار از طریق مطالعه موردی اثبات شود، بایستی ابتدا شرکتی که قرار است مورد مطالعه قرار گیرد، تعیین شده، سپس به کار در این زمینه پرداخته شود. البته لازم به ذکر است که با رشد اجرای پروژه های EAI در سال های اخیر و نیاز جدی به اجرای آن در سال‌های آتی، این محدودیت روز به روز کم‌رنگ تر خواهد شود. در ادامه به مرور مباحث مرتبط با EAI، جایگزین های آن و همچنین جهت گیری آینده و تعدادی از مسائل تحقیقاتی باز در زمینه EAI می پردازیم.

5-4- جهت گیری آینده EAI و چشم انداز تحقیقات آینده

فناوری های یکپارچه سازی اولیه، عمدتاً مشکلات ناهمگنی یا ارتقاء انعطاف پذیر برنامه های کاربردی را برطرف می کردند، اما محیط کسب و کار پیچیده امروز، به راه حل هایی نیاز دارد که تحمل خرابی² سرویس هایشان تضمین شود و چابکی³ لازم را نیز ارائه دهند. در اینجا، منظور از چابکی، مانیتور کردن پشت سر هم تقاضاهای بازار و توانایی پاسخ سریع به این تقاضاها با ارائه محصولات جدید می باشد. جهت گیری یکپارچه سازی، از داده هایی که برنامه های کاربردی آنها را مدیریت می کردند، به سمت سرویس هایی رفته است که کیفیت، دسترس پذیری، تحمل خطا و امنیت آنها، توسط برنامه های کاربردی تضمین شود.

نیاز به یکپارچه سازی، جهت گیری معماری های نرم افزاری را از برنامه کاربردی محوری به سمت سازمان محوری تغییر داده است. بنابراین، عناصر سازنده معماری های نرم افزار، از اشیاء و مولفه های دانه ریز، به سمت سرویس های دانه درشت توسعه پیدا کرده است. در نتیجه، استفاده از معماری های سرویس گرا (SOA) به جای معماری های شی گرا و مبتنی بر مولفه، رواج یافته است.

Enterprise¹
Fault tolerance²
Agility³

متدولوژی یکپارچه سازی، از پیوند ایستای¹ برنامه های کاربردی با همدیگر در زمان کامپایل، به سمت پیوند پویای² سرویس ها با یکدیگر و ایجاد سرویس های ترکیبی در زمان اجرا³ ارتقا پیدا کرده است. بر اساس این تغییر جهت، یکی از شاخه های تحقیقاتی که نیاز به کار دارد، متدولوژی های پیشرفته برای پشتیبانی از دوره عمر ترکیب سرویس⁴ است.

بسیاری از افراد حرفه ای در زمینه یکپارچه سازی، معماری سرویس گرا را به عنوان نسل بعدی فناوری های EAI در نظر گرفته اند. فناوری های EAI سنتی، برای یکپارچه سازی برنامه های کاربردی که به صورت جداگانه توسعه داده شده اند، به خوبی کار می کنند. اما باید توجه داشت که مبنای آن روش ها، راه حل های اختصاصی فروشندگان EAI است که سازمان را به پلتفرم های آن فروشندگان وابسته می کند. اما با فراگیر شدن سرویس های وب به عنوان فناوری مناسب برای انتقال، معماری سرویس گرا و سرویس های وب به عنوان راه حل های یکپارچه سازی مبتنی بر استاندارد مطرح شده اند. این سبب شده که با توجه به مستقل بودن آنها، استفاده از آنها به جای فناوری های EAI متعارف، بی نهایت جذاب شود. معرفی محصولات ESB⁵، یکپارچه سازی مبتنی بر سرویس وب و مبتنی بر استاندارد را بی نهایت مشهور و متداول ساخته است.

اگرچه مقصد نهایی EAI و معماری سرویس گرا، پیاده سازی فرایندهای کسب و کار بر اساس سبب⁶ برنامه های کاربردی موجود است، اما هر کدام از آنها برای رسیدن به این هدف، از روشی استفاده می کنند که تفاوت های کاملاً ریشه ای دارد. EAI روی در معرض نمایش گذاشتن عملکرد⁷ برنامه های کاربردی موجود تمرکز دارد که ممکن است سبب شکل گیری سیستم نهایی نهایی بر اساس سبب برنامه های کاربردی موجود شود. اما در مقابل، SOA روی پنهان سازی برنامه های کاربردی موجود و در عوض، در معرض نمایش گذاشتن مجموعه سرویس ها به صورت مستقل از برنامه کاربردی تمرکز دارد. بر اساس مطالب فوق، یکی از مباحث مهم، بررسی قابلیت های معماری سرویس گرا و محصولات ESB و مقایسه آن با محصولات EAI و بررسی نقاط ضعف و قوت هر کدام می باشد.

1 Static binding
2 Dynamic Binding
3 Run time
4 Service composition lifecycle
5 Enterprise Service Bus
6 Portfolio
7 Expose Functionality

در برنامه های کاربردی قدیمی، فرایندهای کسب و کار به صورت غیر ضمنی درون برنامه های کاربردی تعبیه شده بود. با پیشرفت معماری های نرم افزاری، در سیستم های جدیدتر، لایه ای با نام اتوماسیون فرایندهای کسب و کار به برنامه های کاربردی اضافه شده که تا حدودی با صراحت بالاتری، فرایندها را مدل سازی و مدیریت می کرد، اما تعامل، مدل سازی و مدیریت این لایه از ساختارها و قوانین مختص محصول یا شرکت بخصوصی پیروی می کرد که سبب مشکل شدن تعامل و یکپارچه سازی این فرایندها با سایر فرایندهای کسب و کار می شود. برای حل این مشکل، در سیستم های امروزی و نسل بعدی، برای کشف، طراحی، استقرار، اجرا، نگهداری، تعامل، بهبود و تحلیل فرایندهای کسب و کار، از استانداردهای باز مبتنی بر XML مانند BPML¹ و BPQL² استفاده می شود. سیستم های امروزی، همانند معماری های سه لایه نرم افزاری³ که برای مدیریت لایه داده خود از DBMS استفاده می کنند، برای مدیریت لایه فرایندهای کسب و کار، از یک پلتفرم بخصوص به نام BPMS⁴ استفاده می کنند. با توجه به اینکه در این معماری، فرایندهای کسب و کار، مبتنی بر استاندارد بوده و در یک لایه مجزا قرار دارند، تعامل و یکپارچه سازی فرایندهای کسب و کار سیستم های مختلف با یکدیگر، راحت تر است.

با این تغییر جهت ها، واضح است که در مباحث یکپارچه سازی فرایندی، نیاز جدی به کار روی مباحث مدیریت فرایندهای کسب و کار⁵ (BPM) و ارتباط آن با EAI، احساس می شود. در ضمن، با توجه به اینکه در بعضی از منابع، پیشنهاد شده که به جای رویکرد BPM و پلتفرم BPMS از رویکرد SOA و محصولات ESB استفاده شود، مقایسه SOA با BPM و کشف نقاط مشترک و همچنین مزایا و معایب هر کدام از اینها، به کار پژوهشی نیاز دارد.

در مباحث پیچیده EAI، با ظهور سیستم های چند عامله⁶ (MAS)، بحث قرار دادن این عامل ها در سیستم های نرم افزاری متعارف و یکپارچه سازی با آن سیستم ها مطرح شده است. علاوه بر این، تعبیه یک سیستم نرم افزاری متعارف درون یک سیستم با معماری عامل گرا⁷ مطرح است. استفاده از عامل ها در یکپارچه سازی سرویس گرا و یا یکپارچه سازی فرایندهای کسب و کار، یکی دیگر از مسائل جدید مطرح در زمینه EAI است. یکی دیگر از مباحث تحقیقاتی مناسب، می تواند مقایسه معماری های

¹ Business Process Modelling Language
² Business Process Query Language
³ 3-tier Architecture
⁴ Business Process Management System
⁵ Business Process Management
⁶ Multi-Agent Systems
⁷ Agent Organization

سرویس گرا و یا سایر معماری های متعارف نرم افزاری با معماری های عامل گرا و معرفی ترکیبی بهینه از این دو برای کاربردهای مختلف باشد.

5-5- نتیجه گیری

در این فصل، مباحث تحقیقاتی مرتبط با EAI، مسائل تحقیقاتی باز در زمینه EAI، جهت گیری های فنی و تجاری راه حل های EAI و جایگزین های احتمالی آن مطرح گردید و عنوان شد که کار روی مباحث مربوط به EAI، هم از جنبه علمی و هم از جنبه تجاری، بسیار مناسب می باشد. محدودیت اصلی تحقیق در این زمینه، اثبات کارهای پژوهشی به دلیل نو بودن مبحث و اجرای کم پروژه های نرم افزاری در این زمینه معرفی شد که البته به مرور رفع خواهد شد.

فهرست منابع و مأخذ

1. Alshawi S., Themistocleous M. & Almadani R. (2004) Integrating diverse ERP systems: a case study. *The Journal of Enterprise Information Management*, **17**, 454–462.
2. Barki H. & Pinsonneault A. (2005) A model of Organizational Integration, Implementation Effort, and Performance. *Organizational Science*, **16**, 165.
3. CSC (2002) The Emergence of Business Process Management. p. 90. CSC'S RESEARCH SERVICES.
4. Ebrahim Z. & Irani Z. (2005) E-government adoption: architecture and barriers. *Business Process Management Journal*, **11**, 589.
5. Erasala N., Yen D.C. & Rajkumar T.M. (2003) Enterprise Application Integration in the electronic commerce world. *Computer Standards & Interfaces*, **25**, 69–82.
6. Gorton I. & Liu A. (2004) Architectures and Technologies for Enterprise Application Integration. In: *IEEE Proceedings of the 26th International Conference on Software Engineering (ICSE'04)*. IEEE Computer Society.
7. Gullede T. (2006) What is integration? *Industrial Management & Data Systems*, **106**, 5.
8. Harikumar A.K., Chiang C.-C. & Yang H.-S. (2005) An Event Driven Architecture for Application Integration using Web Services. In: *IEEE*.
9. Irani Z., Themistocleous M. & Love P.E.D. (2003) The impact of enterprise application integration on information system lifecycles. *Information & Management*, **41**, 177–187.
10. Jaakkola L. (2005) Applying Service-Oriented Architecture to Geographically Distributed Industrial Information Systems. In: *Department of Electrical and Communications Engineering*, p. 8 + 92. Helsinki University of Technology, Helsinki, Finland.
11. Janssen M. & Cresswell A.M. (2005) An enterprise application integration methodology for e-government. *The Journal of Enterprise Information Management*, **18**, 531.
12. Khoubati K., Themistocleous M. & Irani Z. (2005) Integration Technology Adoption in Healthcare Organisations: A Case for Enterprise Application Integration. In: *IEEE Proceedings of the 38th Hawaii International Conference on System Sciences*, Hawaii.
13. Lam W. (2005) Barriers to e-government integration. *The Journal of Enterprise Information Management*, **18**, 511.
14. Linthicum D.S. (2003) *Next Generation Application Integration: from simple information to Web Services*. Addison-Wesley.
15. Müller C. (2005) ERP II: a conceptual framework for next-generation enterprise systems? *Journal of Enterprise Information Management*, **18**, 483.
16. Manolescu D.A. & Lublinsky B. (2006) SOA Enterprise Patterns: Services, Orchestration and Beyond, COPY Draft available at <http://orchestrationpatterns.com/>.
17. Mosawi A.A., Zhao L. & Macaulay L. (2006) A Model Driven Architecture for Enterprise Application Integration. In: *IEEE Proceedings of the 39th Hawaii International Conference on System Sciences*, Hawaii.
18. N.Huhns M. & Singh M.P. (2005) Service-Oriented Computing: Key Concepts and Principles. *IEEE Computer Society*.

19. Papazoglou M.P. & Heuvel W.-J.v.d. (2005) Service Oriented Architectures: Approaches, Technologies and Research Issues. In: *The VLDB Journal manuscript*.
20. Pereira J. (2006) Enterprise Application Integration: Approaches to Integration. In: *MindTree consulting white paper*.
21. Pinkston J. (2001) The Ins and Outs How EAI Differs. *eAI Journal*, 48.
22. Puschmann T. & Alt R. (2004) Enterprise application integration systems and architecture-the case of the Robert Bosch Group. *The Journal of Enterprise Information Management*, **17**, 105.
23. Qureshi K.A. (2005) Enterprises Application Integration. In: *IEEE International Conference on Emerging Technologies*, Islamabad.
24. Raut A. & Basavaraja A. (2003) Enterprise Business Process Integration. In: *IEEE TENCON Poster Papers*, p. 1550.
25. Themistocleous M. (2004) Justifying the decisions for EAI implementations: a validated proposition of influential factors. *The Journal of Enterprise Information Management*, **17**, 85.
26. Themistocleous M. & Irani Z. (2001) Benchmarking the benefits and barriers of application integration. *Benchmarking: An International Journal*, **8**, 317.
27. Themistocleous M. & Irani Z. (2006) Towards a Methodology for the Development of Integrated IT Infrastructures. In: *IEEE Proceedings of the 39th Hawaii International Conference on System Sciences*.
28. Themistocleous M., Irani Z. & Love P.E.D. (2004) Evaluating the integration of supply chain information systems: A case study. *European Journal of Operational Research*, **159**, 393–405.
29. Themistocleous M., Irani Z. & O'Keefe R.M. (2001) ERP and application integration Exploratory survey. *Business Process Management Journal*, **7**, 195.
30. Yoo M.-J. (2004) Enterprise Application Integration and Agent-Oriented Software Integration. In: *IEEE International Conference on Systems, Man and Cybernetics*.
31. Zhao J.L. & Cheng H.K. (2005) Web services and process management: a union of convenience or a new area of research? *Decision Support Systems*, **40**, 1.