



به تدریج که جلو می‌رویم، باید فن‌آوری‌های جدیدی بسازیم که بتوانند از سیستم‌های به جا مانده از قبل استفاده کنند و آن‌ها را از نو با هم ترکیب کنند تا راه‌حل‌های جدید جا بیفتند. شاید ما نیاز داشته باشیم تاریخ را تکرار کنیم. ده سال دیگر و یک سکوی جدید، در حالی که .NET و J2EE قدیمی شده‌اند.

قسمت دوم

# کدام را انتخاب کنیم؟ J2EE یا .NET

ترجمه: بهروز نوعی پور

اشاره

پیش از این وعده داده بودیم که به دلیل اهمیت تحولات آتی صنعت نرم‌افزار و اینترنت، که طی مقالات و گزارشاتی به مقایسه میان فناوری‌های دات‌نت مایکروسافت و J2EE سان مایکروسافت می‌کنیم. در این جا قسمت دوم و پایانی مصاحبه‌ای که در شماره ۲۴ مجله در همین زمینه به چاپ رسید را ملاحظه می‌کنید. این مصاحبه در برگزیده یک بحث فنی بین این دو فناوری است که توسط مجله معتبر ویژوال استودیو انجام شده است. لازم به یاد آوری است که دن گیدا مدیر اجرایی شرکت Infragistie است که در زمینه طراحی و توسعه سیستم‌های نرم‌افزاری مبتنی بر وب فعالیت می‌کند.

انتخاب‌هایی که برای تعیین استراتژی آتی دات‌نت داشته است، بهترین را برگزیده است؟  
○ من این‌طور فکر نمی‌کنم. در واقع مایکروسافت می‌توانست بهتر عمل کند. اگر آن‌ها قصد داشته‌اند برنامه‌نویسان را قادر به تولید برنامه‌های تحت وب و توزیع‌شده (distributed) سازند، بایستی ابزارهای برنامه‌نویسی خود را از نو اصلاح کنند. اگر آن‌ها خواهان این هستند که بهترین پلتفرم را برای برپا کردن شبکه‌های TCP/IP، برای پشتیبانی از وسایل کامپیوتری گوناگون (مثل PDA) و با درجه مطلوبی از امنیت اطلاعات داشته باشند، باید حتماً این اصلاحات را انجام دهند.

دارم که براساس آن فکر می‌کنم همه برنامه‌نویسان «مجبورند» بالاخره روزی به سراغ برنامه‌نویسی شی‌گرا بروند، با چارچوب‌های جدید نرم‌افزاری آشنا شوند و بیاموزند که چگونه کارهایی که سابقاً انجام می‌دادند را با روش‌های بهتر و کارآمدتری به انجام برسانند. این مسأله مستقل از نوع پلتفرمی است که آن‌ها انتخاب می‌کنند. برنامه‌نویسان ویژوال بیسیک نیز - اگر بخواهند سراغ دات‌نت بروند - ناگزیر از طی کردن یک منحنی و روند آموزشی معین (Learning Curve) خواهند بود.

● با فرض در نظر گرفتن رقابتی که از ناحیه توسعه‌دهندگان جاوا و J2EE وجود دارد آیا می‌توان گفت که مایکروسافت از بین

● چه انتخاب‌های دیگری برای کسانی که مایلند روشی به غیر از به کارگیری دات‌نت و VB.NET را برگزینند وجود دارد؟  
○ طبیعتاً یک انتخاب مسلم همان J2EE است. اما اگر برنامه‌نویسی اصرار داشته باشد کماکان از پلتفرم ویندوز برای کارش استفاده کند، زبان دلفی بورلند - که اخیراً نیز بهبود یافته است - گزینه مطلوبی است. اما راستش را بخواهید، در واقع بیش از دو گزینه عمده نداریم: یا دات‌نت یا J2EE

● خیلی بعید است که آن دسته از کاربران VB6 که از دات‌نت خوششان نیامده است، سراغ J2EE بروند. شما چه فکر می‌کنید؟  
○ کاملاً درست است. در واقع من یک فرضیه

● به نظر من شما فکر می‌کنید که زبان C بسیاری از برنامه‌نویسان و ویژوال بیسیک را به سوی خود خواهد کشید. احساس من این است که این اتفاق دست کم در حال حاضر رخ نمی‌دهد. آیا تحقیقات شما چنین چیزی را نشان داده است؟ اگر چنین است چه دلیلی دارید؟

○ در واقع تاحدودی همین‌طور است. به هر حال C بیشتر شبیه ++C و جاوا است، بنابراین اگر برنامه‌نویسان و ویژوال بیسیک قصد مقایسه C و VB.NET را داشته باشند، طبیعتاً به VB.NET متمایل خواهند بود چرا که صرف‌نظر از تغییراتی که در آن رخ داده است، دست کم برای برنامه‌نویسان و ویژوال بیسیک، زبانی آشناست. C به احتمال زیاد اکثر برنامه‌نویسان ++C را جذب خواهد کرد و عده‌ای از علاقه‌مندان و ویژوال بیسیک نیز به سوی آن خواهد رفت. چیزی که مهم است این است که نهایتاً همه این زبان‌ها در چارچوب دات‌نت، پوسته‌های متفاوتی هستند برای زبان MSIL و صرفه‌نظر از نوع زبانی که انتخاب کرده‌اید، سورس کد شما به IL ترجمه خواهد شد.

● ما می‌دانیم که در شرکت شما (Infragistics)، برنامه‌نویسان خیلی سریع با C سازگاری پیدا کردند. چه مزایایی در این زبان باعث شد که آن‌ها برای مهاجرت به این زبان برنامه‌نویسی علاقه نشان دهند؟

○ برنامه‌نویسان ما اصولاً پیش از به کارگیری دات‌نت، با قالب برنامه‌نویسی شی‌گرا آشنایی داشتند. آن‌ها اشیای COM را با ++C می‌ساختند و از جاوا برای ساختن Java Bean استفاده می‌کردند. وقتی که آن‌ها کار با C را آغاز کردند، قبلاً تفکر جاوا و منطق عناصر COM را در پیش زمینه داشتند. برای آن‌ها مهاجرت از جاوا به C آسان بود. آنان زبان ++C را نیز به خوبی می‌دانستند بنابراین در روی آوردن به C مشکلی نداشتند. از سوی دیگر کار کردن با IDE و ویژوال استودیو دات‌نت راحت است و بهره‌وری کار را بالا می‌برد.

● آیا هیچ نگران این هستید که هرکس بتواند اشیاء NET (.NET components) ساخته شده توسط شما را decompile کند؟

○ نه چندان، اما decompile کردن اشیاء NET کار دشواری نیست. به همین دلیل قصد داریم فروش سورس کد را همراه محصولاتمان متوقف کنیم و فقط خود اشیاء دات‌نت را بفروشیم. به هر حال این دغدغه همیشه در کار تولید قطعات نرم‌افزاری وجود دارد چون هزینه زیادی صرف به

ثمر رساندن یک پروژه می‌شود.

● در مقایسه با پلتفرم جاوا، ساختن اشیاء دات‌نت آسان‌تر است یا سخت‌تر؟

○ Obfuscatorها در پلتفرم جاوا کمی بهتر و کارآمدترند، اما مطمئنم که در آینده نزدیک در پلتفرم دات‌نت نیز همین‌قدر خوب خواهند بود. به هر حال ما از روش obfuscation برای حفاظت از شماره IP استفاده می‌کنیم. در مورد جاوا، سورس کد را هم همراه محصول می‌فروشیم. ما هنوز در تلاشیم تا برای قیمت‌گذاری اشیاء دات‌نت ساخت خودمان به یک مدل مناسب برسیم.

● به عنوان یکی از اولین عرضه‌کنندگان ابزارهای کمکی برنامه‌نویسی برای عده زیادی از برنامه‌نویسان دات‌نت، فکر می‌کنید چه تغییراتی در روش برنامه‌نویسی روی پلتفرم مایکروسافت به وجود خواهد آمد؟ مشتریان فعلی شما چه مشکلاتی در این زمینه داشته‌اند؟

○ متأسفانه یاد گرفتن دات‌نت زمان قابل توجهی می‌برد و این طبیعت فناوری‌های جدیدی است که جهش زیادی در روند صنعت ایجاد می‌کنند. در میان پذیرندگان دات‌نت، افرادی که پیش‌تاز هستند مشکل کمی خواهند داشت. آن‌ها زمان به اندازه کافی برای سازگار شدن با پلتفرم جدید دارند، اما بقیه برنامه‌نویسان به راهنمایی و هدایت نیاز دارند.

● جاوا تاکنون معماری خود را برای پشتیبانی از سرویس‌های وب به صورت استاندارد در نیاورده است در حالی که آی‌بی‌ام هم اکنون از معماری سرویس‌های وب در نرم‌افزار WebSphere خود پشتیبانی می‌کند. آیا فکر می‌کنید این وضعیت باعث پایان یافتن حاکمیت شرکت سان بر دنیای جاوا خواهد شد؟

○ من این‌طور فکر نمی‌کنم. در حال حاضر به دلیل باز بودن پلتفرم جاوا و حضور شرکت‌های نرم‌افزاری متعدد در این عرصه، مجال متولد شدن یک چارچوب و معماری کاملاً جدید مبتنی بر جاوا در این بازار وجود دارد، اما فراموش نکنید که به هر حال این لازم است که کسی این چارچوب جدید را صاحب شود و آن را هدایت کند و فعلاً هیچ‌کس بهتر از سان نمی‌تواند این کار را انجام دهد. من فکر می‌کنم به هر حال متخصصان IT چنین فشاری را برای رقابت احساس می‌کنند ولی برای جاواچندان جالب نیست که به دو جبهه تقسیم شود و یکدیگر

را تضعیف کنند.

● مهم‌ترین مزیت سرویس‌های وب کدامند؟

○ سرویس‌های وب راه بسیار مناسبی برای استفاده مجدد از عناصر نرم‌افزاری است. به جای این‌که برای کاربردهای جدید، نرم‌افزارهایمان را دوباره بنویسیم اشیائی که از قبل موجود بوده است را در قالبی نو ترکیب می‌کنیم. به عبارت دیگر یک لایه بیرونی تازه بر روی اشیاء نرم‌افزار کنونی قرار می‌دهیم تا راه‌حل‌های جدیدی برای کسب و کار پیدا کنیم. این روش کمک می‌کند فناوری‌ها و سیستم‌های موجودمان را به صورت بهینه‌ای یکپارچه سازیم. علاوه بر این، سرویس‌های وب راه‌حل مناسبی برای پل زدن بین راه‌حل‌ها و سیستم‌های نرم‌افزاری J2EE از یک سو و نرم‌افزارها و راه‌حل‌های دات‌نت است. به این ترتیب می‌توانیم از قابلیت‌های هر دو فناوری به طور هم‌زمان استفاده کنیم.

● نظراتان درباره سیستم‌های ناهمگن (سیستم‌هایی که از ترکیب نامنظمی از عناصر نرم‌افزاری مبتنی بر جاوا، دات‌نت و غیره ساخته شده‌اند) چیست؟ آیا شکاف‌هایی در این عرصه وجود دارد که لازم باشد پر شوند با این که هم اکنون این پلتفرم‌های متفاوت قادرند به راحتی با هم کار کنند؟

○ قطعاً شکاف‌هایی وجود دارد و برنامه‌نویسان و شرکت‌های باهوش کارشان پیوند دادن این فناوری‌های متفاوت به یکدیگر است. در حال حاضر طراحی معماری که بتواند به آسانی کار کند و اشکال‌زدایی شود خیلی دشوار است و بسیاری از برنامه‌نویسان هم اکنون در حال کار برای ساده کردن این پیوند هستند.

● سرعت و کارایی معماری‌های ترکیبی و ناهمگن در مقایسه با معماری‌های همگن (فقط از نوع دات‌نت یا فقط از نوع J2EE) چقدر است؟ آیا فکر می‌کنید وضعیت بازار به گونه‌ای است که ما در آینده به سوی معماری‌های ترکیبی خواهیم رفت؟

○ مسأله بازده و کارایی (performance) مسأله کلیدی است. وقتی که بایک معماری ناهمگن روبرو هستید باید به شرایطی که سیستم‌ها قادر به برقراری ارتباط با یکدیگر نیستند هم فکر کنید چرا که همه اتفاقات داخل یک CPU نمی‌افتد. در سیستم‌های قدیمی چنین بود، همه چیز در یک پردازنده اتفاق می‌افتاد و حداکثر از چند حافظه

مشترک بهره می‌گرفتند، اما در محیط‌های ناهمگن مجبورید سیستم‌های متفاوتی را به هم وصل کنید، با بسته‌های نرم‌افزاری و سخت‌افزاری مختلفی روبرو هستید.

مسئله تأخیر زمانی در پاسخ (واکنش) سیستم‌ها مطرح است. همه این‌ها به وضعیتی منجر می‌شود که ممکن است باعث شود در لحظات خاصی این سیستم‌ها از همکاری با یکدیگر سرباز زنند و حلقه‌های مرده و بی‌حاصلی از ارتباطات داده‌ای پدید آید. خیلی اوقات مجبورید برای وصل کردن دو سیستم به یکدیگر از یک سیستم ثالث استفاده کنید، و این خود به خود فرآیند تعامل سیستم‌ها و داده‌ها را پیچیده‌تر می‌کند. بنابراین هنوز مشکلات زیادی وجود دارد که باید حل شود و در اغلب این مسائل، بازدهی کار موضوع کلیدی است.

به هر حال تاریخچه فناوری به ما می‌آموزد که ما مجبوریم از این نوع معماری‌ها هم داشته باشیم. این بخشی از واقعیت‌های صنعت است چرا که به مرور زمان، فناوری‌های مختلف ظهور می‌کنند و ناگزیر از رویارویی با چنین شرایطی هستیم.

زندگی خیلی آسان‌تر بود اگر فقط یک پلتفرم داشتیم، یک محیط داشتیم، اما حقیقت غیر از این است. ماهمه این فناوری‌های گوناگون را داریم چرا که مرتباً ایده‌ها و سیستم‌های تازه‌ای پدید می‌آیند. به تدریج که جلو می‌رویم، باید فناوری‌های جدیدی بسازیم که بتوانند از این سیستم‌های متفاوت کنونی استفاده کنند و آن‌ها را از نو با هم ترکیب کنند تا به راه‌حل‌های جدیدی برسیم. این مشکل هیچ وقت به پایان نمی‌رسد. شاید ما نیاز داشته باشیم تاریخ را تکرار کنیم. ده سال دیگر پلتفرم جدیدی خواهیم داشت و محیط‌ها و سیستم‌های قدیمی ما از نوع دات‌نت و J2EE خواهند بود.

● به عنوان سوال آخر، آینده راه‌حل‌های Server-Side را در مقابل روش‌های Client-Side چگونه می‌بینید؟ آیا صنعت IT به سمتی خواهد رفت که هرچه بیشتر و بیشتر از مرورگر وب به عنوان سرویس‌گیرنده استفاده کند یا این که برنامه‌نویسان سعی خواهند کرد روش‌هایی بیابند که سرویس‌های وب را در کنار

معماری‌های نرم‌افزاری قدیمی به کار گیرند؟  
○ من عقیده دارم که هر دو اتفاق خواهند افتاد. اما اگر بپرسید کدام یک بر بازار غالب خواهد بود من می‌گویم راه‌حل‌هایی که در آن‌ها سرویس‌گیرنده قادر به انجام کارهای بیشتری است، برتری خواهند داشت. نهایتاً، مردم خواهان راه‌حل‌هایی خواهند بود که در آن‌ها هم سرویس‌گیرنده و هم سرویس‌دهنده کارایی بسیار بالایی داشته باشد.

مصرف‌کنندگان معمولاً مدافع شعار کوچک شدن حجم سرویس‌گیرنده هستند، اما چیزی که واقعاً به دنبالش هستند این است که بتوانند در هر جا و موقعیتی - ثابت یا سیار - به اطلاعات مورد نیازشان دسترسی داشته باشند و این به معنای آن است که سرویس‌گیرنده‌های توانمندتری می‌خواهند که در عین حال کم‌حجم‌تر هم باشد. اما یک فناوری معین نمی‌توان هم‌زمان به هر دو نیاز پاسخ دهد. بنابراین ما ناگزیر از انتخاب هستیم. در واقع نیاز مشتری تعیین می‌کند که کدام فناوری را باید برای یک پروژه خاص برگزید.