

IBM Rational Business Developer Extension：概觀

等級：入門

IBM Rational Business Developer 及 EGL 產品線經理 [Stefano Sergi](#)

2007 年 5 月 15 日

有了 Rational Business Developer Extension (RBDe)，開發團隊便能使用 IBM 在 2006 年推出的企業產生語言 (Enterprise Generation Language, EGL)，本文提出 IBM 開發 EGL 的原因，並討論 IT 組織該如何使用此語言快速提高生產力，以因應目前市場上的 Web 及服務導向趨勢。

摘自《Rational Edge》電子月刊。

爲了改善業務效率及提升競爭力，目前全球經濟普遍追求高成本效益、透過合併或購併進入新市場，且委外服務和善用資源策略 (right-sourcing) 的需求也不斷增加，使得資訊系統越來越重要。隨著電腦網路及網際網路的蓬勃發展，逐漸改變了企業經營業務及使用資訊方式，也使傳統的獨立應用程式時代漸趨沒落。

目前許多 CIO 迫切需要新的整合資訊解決方案，以改善夥伴、客戶及員工之間的業務運作方式，但爲避免獨立應用程式影響公司的回應能力及彈性，有先見之明的 CIO 選購新商業軟體時，均以服務模型作爲架構基礎。

不過新 IT 系統的軟體開發功能，必須能支援新中介軟體及新興應用程式架構。此外，開發團隊已陷入迷思，認爲提升主從式運算能力，一定可同時提高生產力並簡化業務流程。爲因應此類需求，IBM 開發出 IBM Rational 軟體遞送平台，藉由整合的工具組、方法及最佳實務，有效控管軟體開發流程，以便能更快轉型爲 SOA 架構。

新產品名稱爲 IBM Rational Business Developer Extension，搭配主從式作業團隊，可在 COBOL 或其他舊有的語言環境工作，而功能強大的工具可快速遞送服務，不須耗費大量時間進行 Java 相關訓練。有了 Rational Business Developer Extension (RBDe)，開發團隊便能使用 IBM 在 2006 年推出的企業產生語言 (Enterprise Generation Language, EGL)，本文提出 IBM 開發 EGL 的原因，並討論 IT 組織該如何使用此語言快速提高生產力，以因應目前市場上的服務導向趨勢。

從歷史角度探討 EGL 的優點

IBM Rational 軟體遞送平台的核心，是透過 Eclipse 工作平台提供的一組 SOA 設計與建構功能，其中包括強大的高效能工具，可協助軟體架構工程師、設計師、Web 開發人員及 Java 開發人員，組織並加快其工作項目。不過，和業界多數工具領導廠商不同的是，IBM 承認 Java 雖然有非常多種系統、中介軟體 API 及生產力提升架構，但技術難度也相當高，必須花費很長的時間不斷學習。

業界並非第一天知道這個問題，而每一種新運算技術出現時，必然會帶來新的程式設計難題，使開發人員編寫新的業務功能時，必須花費額外的時間學習及編寫新介面。舉例來說，線上交易處理 (OLTP) 於七零年代早期出現後，應用程式開發人員必須學習掌握 OLTP 中介軟體的服務介面，像是 CICS 及 IMS 的 COBOL 或 Assembler API 介面；隨著關聯式資料、圖形介面及主從式運算的出現，開發人員則須學習編寫資料庫中介軟體及 GUI 服務中介軟體的精細介面；而之後出現的大規模應用程式整合趨勢，更使開發人員必須熟悉全新的整套技術，以便使用應用程式整合中介軟體 (Application Integration Middleware, AIM) 階層，包括訊息導向中介軟體及配接或分配管理系統 API。

雖然在部分情況下，完整存取系統介面功能有其必要性，但應用程式開發人員可依據一般使用型樣，以較簡單的介面提供重要的需求，程式設計師就不須進行低階編碼作業。

接下來繼續從歷史觀點來看，使用此類工具擷取 GUI 小組件或相關資料表的使用型樣，再轉換到視覺工具及語言整合，確實讓許多桌面及主從式應用程式的開發工作輕鬆許多，也是 PowerBuilder、Visual Basic 及 Delphi 風光的年代，不過較複雜或專業的功能需求，仍須撰寫 C 或 C++ 程式，但只要透過這些簡單且抽象化的程式設計環境，就能寫出絕大多數的應用程式，而且簡單易學又能提高程式設計團隊生產力。

目前是應用程式伺服器、應用程式整合及 SOA 的年代，開發人員面對的困境依舊差不多，因此 IBM Rational 開發了企業產生語言 (EGL)，作為 IBM Rational 軟體遞送平台內建的 Eclipse 程式設計工作平台。在 Rational Business Developer Extension 7.0 版中，便已整合了最新的 EGL 整合開發平台 (IDE)。

相較於 Java 及 Eclipse 環境，EGL 不僅簡單易學，還能提高開發團隊生產力，以簡易且抽象化的開發模式拓展了 Java 的深度與廣度。無論開發人員的技術背景為何，都能快速寫出跨平台的交易資料導向服務及應用程式。

EGL 並非是要取代 Java，而是要讓 IT 組織能投入並善用 Java EE 的強大功能，開發團隊也不須學習熟悉 Java 程式設計在低階部分的複雜程序。不過，和之前的 PowerBuilder 及其他 4GL 工具一樣，在某些情況下還是必須撰寫 Java 程式碼，但應用程式大部分或全部的編寫工作，僅需使用 EGL 就能完成。

EGL 的目標使用者是業務導向的開發團隊，這類團隊需要開發現代化的應用程式及服務，因此希望語言環境簡單易學且具有高生產力，但卻又負擔不起將開發人員訓練成 Java 專家的代價及風險。

何謂 EGL？

企業產生語言 (EGL) 是先進的程式設計語言，特別針對業務導向開發人員的需求設計，讓他們能善用 Java 平台的強大功能，卻又不須完全學習 Java 的複雜細節，僅須專注開發商業應用程式。EGL 具有簡單的高階功能，可讓您利用 Java 及現代 Web 技術，快速寫出全功能應用程式及服務。開發人員可利用 Rational Business Developer Extension (RBDe) 的強大功能，以 EGL 原始碼撰寫其業務邏輯，接著 RBDe 工具就會自動產生 Java 或 COBOL 程式碼，以及您所需的所有執行時期構件，讓您將應用程式部署到特定執行平台。

使用 EGL，便不須處理 Java 平台及其他相關中介軟體介面的複雜細節，開發人員便能專注處理業務相關問題。即使 Java 或 Web 技術經驗不足，開發人員也能使用 EGL 輕鬆且快速建立企業級服務與應用程式。

EGL 並不提供作業系統及子系統的低階 API，因此並不適合用來開發系統，但卻非常適於開發交易業務的服務及應用程式。另外 EGL 也具備完整的 Java 交互作業能力，而且也可與 Eclipse Java IDE 充分整合，完全符合絕大多數的系統要求。

Rational Business Developer Extension 可建立哪些應用程式？

Rational Business Developer Extension 經過特別設計，可完全符合商業應用程式的要求，其內容包括：

- **商業服務：**EGL 語言內建服務功能，讓您在最簡單且直接的方式建立並使用服務，而透過 EGL 語言，也可讓磁碟系統實作符合服務導向的完整架構原則。
- **Web 應用程式：**EGL 與 Java Server Faces 架構和工具經過充分整合，即使不瞭解 Java 及 JSF 架構的細節，也可快速輕鬆建立 Web 應用程

式。

- **報告：**EGL 也結合了開放原始碼報告引擎 Jasper Reports，可讓您建立各種專業報告。
- **批次系統：**EGL 包含批次程式功能，不須使用者介入即可自動執行所需作業，像是產生報告或批次資料庫載入/更新。
- **文字介面應用程式：**為能順利讓舊式系統轉換到現代開發環境，EGL 也可建立傳統的字元使用介面（綠色畫面）及相關的程式設計建構。如有需要，此功能也可用來進行新的開發作業。
- **GUI 應用程式：**透過 EGL 最新的加強功能，建立的應用程式便能以圖形使用者介面與使用者互動。對 GUI 介面的初步支援，讓 Informix 使用者能順利移轉目前使用主控台 UI 介面的 i4gl 應用程式，但 IBM 計畫在下一版 RBDe 中，還會推出功能更強大的 UI 介面。

EGL 應用程式如何指定執行方式

RBDe 內建程式碼產生引擎，可將應用程式及服務的 EGL 原始碼轉換為 Java 或 COBOL 原始碼。

轉換後的 COBOL 可部署到以下環境：

- z/OS 1.6 版或更新版本：批次，IMS 或 CICS 應用程式
 - IMS/ESA 7.1 版或更新版本（IMS 應用程式）
 - CICS TS 2.2 版或更新版本（CICS 應用程式）
- iSeries 5.3 版或更新版本

轉換後的 Java 可部署到以下環境：

- z/OS 1.6 版或更新版本
 - IBM SDK for z/OS Java 2 Technology Edition 1.4 版，可在 USS 中執行 Java 程式
 - WebSphere Application Server for z/OS V5 或更新版本，執行 Web 應用程式
- iSeries 5.3 版或更新版本
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式
 - WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
- AIX 5.2 版或更新版本
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式

- WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
- Apache Tomcat 5.5 版或更新版本，執行 Web 應用程式
- Windows 2000、2003、XP
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式
 - WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
 - Apache Tomcat 5.5 版或更新版本，執行 Web 應用程式
- Linux (RedHat 3 或更新版本、Suse 9)
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式
 - WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
 - Apache Tomcat 5.5 版或更新版本，執行 Web 應用程式
- HP Unix 11iv2
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式
 - WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
 - Apache Tomcat 5.5 版或更新版本，執行 Web 應用程式
- Solaris 9 或更新版本
 - JDK 1.4 版或更新版本，執行非 J/EE Java 程式
 - WebSphere Application Server 5.1.2 版或更新版本，執行 Web 應用程式
 - Apache Tomcat 5.5 版或更新版本，執行 Web 應用程式

如要部署 EGL Java 應用程式，需在目標環境安裝一組執行時期程式庫 (JAR 檔案)：Rational Business Developer Extension 7.0 版產品中免費提供這些檔案。

如要部署 iSeries 的 EGL COBOL：Rational Business Developer Extension 7.0 版產品中免費提供這些檔案。

如要部署 zSeries 的 EGL COBOL 應用程式，需在目標環境安裝一組執行時期程式庫：IBM Rational COBOL Runtime for zSeries V6 產品中免費提供這些檔案。

支援資料庫

EGL 具備簡單但強大的語言元素，可加速存取不同資料庫中的資料，程式設計師不須精通 SQL 資料操作語言、JDBC 或其他 SQL 存取程式設計技術，也不須學習檔案或階層式資料庫存取 API。只要使用一組多型態動詞，程式設計師

就能寫出全功能服務，存取下列資料儲存庫：

- DB2 7 及更新版本
- IDS 7.3 及更新版本
- Oracle 8 及更新版本
- SQL Server 2000
- Derby 10
- VSAM
- IMS/DB (DL/I)

EGL 功能強大的原因

EGL 大大提高業務導向開發人員的能力，使其能在很短的時間內，透過下圖 1 所示的 SOA 相關功能發揮極大的生產力。

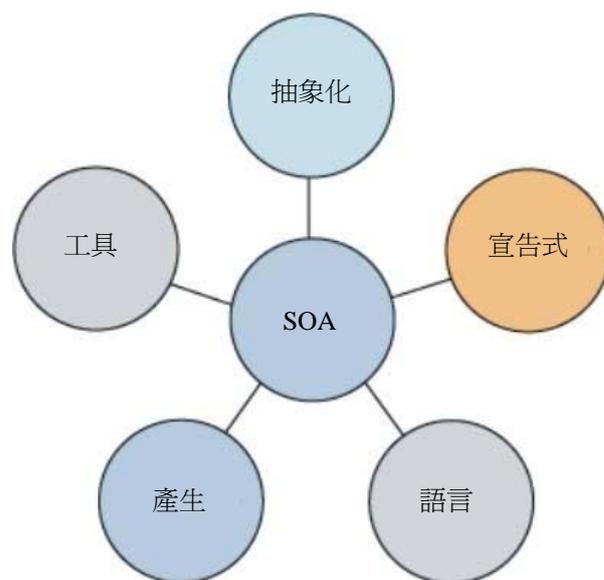


圖 1: EGL 支援多種 SOA 相關功能，讓業務導向開發人員能在很短的時間內，發揮極大的生產力。

- **抽象化**：EGL 使用精簡但強大的表示法，不僅可避免緊密耦合，還能減少介面系統及中介軟體所需的編碼作業。抽象化功能大幅簡化且加快開發人員的工作效率。
- **宣告式程式設計**：EGL 在可能且合適的情況下，納入一定程度的宣告式功能，以便減少重複且容易出錯的編碼作業。例如，若將驗證規則與某資料項目產生關聯，則每次在特定環境定義中用到此項目時，便會自動套用並執行驗證。

- **語言：**EGL 的語言表示法具備現代化及模組化等特性，功能全面又容易學習，內建多樣化的程式庫，可大幅提高開發人員處理一般作業的速度，例如資料及時間運算、字串操作等。此外，EGL 的語言具備擴充能力，可與其他語言進行完整交互作業，還提供 EGL 與原生 Java 的介面。
- **產生：**EGL 的開發技術雖然經過簡化，但也必須保證執行時期平台的最佳部署效果，才能充分利用其服務品質（Qualities of Service，QoS），並在作業中進行系統的原生管理及監視。這些功能是透過 RBDe 隨附的程式碼產生引擎所完成，可將 EGL 功能轉換為原生 Java 或 COBOL 原始碼，並建立其他必要的部署構件。
- **工具：**為進一步提高開發人員的工作效率，RBDe 提供一組依據 Eclipse IDE 架構建置的強大工具，其中包括供除錯用的 EGL 原始碼動畫功能、強大的智慧編輯功能、視覺建構功能、圖形導覽功能，以及各種專業工具，可將 EGL 表示法與圖形 Web 開發工具緊密結合、自動將 UML 模型或資料庫綱目轉換為全功能 EGL 服務及應用程式等。
- **SOA：**如同本文件一開始的介紹，服務導向對於遞送新商業解決方案的重要性越來越高，因此 EGL 完全針對此項需求來設計，語言本身也內建經過簡化的抽象化參照範例，並增加了各項工具及產生功能，完全與上述 EGL 強大功能的基本原則一致，而「服務」的程式設計功能，也可讓磁碟系統實作符合服務導向的完整架構原則。即使不瞭解 Web 服務的通訊協定及標準，如 WSDL、SOAL、XML 等，不同技術背景的開發人員也能輕鬆建立服務。

EGL 及 Rational Business Developer Extension 的價值

組織若能提供現代化且功能強大的 Web 及 SOA 解決方案，就能擁有下列優勢：

- 提高程式設計人員生產力
- 使更多開發人員能遞送現代化的解決方案及 SOA
- 消除技術藩籬，改善作業彈性及反應能力
- 降低訓練成本
- 減少錯誤並提高品質
- 簡化舊式整合方式
- 可彈性選擇部署選項，降低開發及維護成本。可用選項包括：
 - 所有主要 IBM 作業系統及執行時期
 - 跨平台解決方案

誰能因使用 **Rational Business Developer Extension** 而獲益？

EGL 解決方案功能強大，特別適合具有以下需求的應用程式開發組織：1) 開發的新專案需要 Web 開發或 SOA，且面臨技術困難及時間不足等狀況；或 2) 計畫更新及改善現有軟體開發程序及工具。接著再看看 **Rational Business Developer Extension** 設計時所考量到的各種不同平台、技術集及相關狀況。

使用 **System i** 及 **System z** 的組織

通常內部開發團隊都會分成兩組截然不同的族群：傳統開發人員，負責處理 RPG、COBOL、4GL 等；新開發人員，負責處理 GUI、Windows、Java、Web 等。如果傳統開發團隊夠大，且面臨遞送新業務解決方案的壓力，就能體現出 EGL 的價值前提。

EGL 還能改善開發團隊的孤島問題，使程式設計資源能獲得合理運用。若傳統開發人員無法輕鬆進行 Web 或 SOA 程式設計作業，而新開發人員則無法處理傳統程式碼，就不易管理不同技術的團隊，也會降低工作效率。無論是何種團隊都可輕鬆學會 EGL，讓開發人員能透過單一技術平台，一起從頭到尾開發新專案。

EGL 對這些組織還有另一項重要價值，就是能提高開發人員的工作原動力。EGL 具備現代化的強大程式設計環境，且在設計時考量到開發人員的需求，不僅能留住現有設計人員，還能吸引新的設計人員。

採用過時 **4GL** 語言（**IBM** 及非 **IBM**）的組織

目前使用或已使用過第四代產生語言（Fourth Generation Language，4GL）的開發團隊，會因以下原因而對 EGL 感興趣。首先，這些團隊多半希望使用的程式設計工具能具有強大生產力，否則當初就不會想投資 4GL。若在轉型至 Web 及 SOA 時，因供應商限定了產品功能，或新系統的需求超出 4GL 的功能範圍，導致 4GL 無法發揮原有價值，採用 EGL 便能充分善用 4GL 開發人員的技術，有效延長原有投資的效益。

4GL 開發人員可以很快學會 EGL 的用法，而移轉/轉型工具及服務則能將舊有 4GL 程式碼轉換至 EGL，使舊程式碼也能發揮出應有價值。

使用 EGL 開發環境內建的工具，就能輕鬆將以下 **IBM 4GL** 移轉至 EGL：
Informix 4GL、**VisualAge Generator**、**CSP**、**VisualGen**。

使用協力廠商工具及服務，就能將以下非 **IBM 4GL** 轉型為 EGL：**Natural**、**CA**

(COOL-Gen、COOL-Enterprise、Ideal)、HPSeer。

組織中的開發人員不熟悉 Web 應用程式或使用 Java 及 J/EE 的 SOA

若組織計畫發展 Web 及/或 SOA 應用程式，但對於 Java 及 J/EE 平台經驗不足，或開發人員專業能力不足，無法進行物件導向開發作業，便無法負擔重新訓練員工的費用及時間。為避免長期依賴及成本支出，他們通常會將開發作業外包給協力廠商，這些廠商應該研究一下 EGL。

區域及全球系統整合廠商

由於系統整合廠商專門為其他公司建置自訂商業解決方案，若採用 EGL 將能降低開發成本，新進員工的訓練時間更短，且開發團隊不會只具備單一平台技術，可彈性處理不同專案，因此特別適合改用 EGL。

此外，系統整合廠商常須處理大規模的重新編寫或程式現代化專案，若能採用 EGL 將大大提高其競爭優勢，若搭配自動化轉型工具更是如虎添翼。前文中提到的自動化工具，也能將逐漸過時的 COBOL 應用程式轉型至 EGL。

獨立軟體廠商

EGL 可大大提高 ISV 的競爭優勢，並為其創造新市場的商機。和系統整合廠商不同的是，ISV 的產品為可重複銷售的一般商業解決方案及自訂服務，對象則是希望採用現成產品的公司。EGL 最大的一項優勢是可因應不同平台，開發彈性的解決方案產品，另外還能降低開發成本並加快員工訓練進度，讓人力資源的調配更具彈性。

一般問題

以下是我們遇到過的一般問題及疑問，所提供的答案應該能滿足您的需求。

問：產生的應用程式是否有足夠效能？

答：相較於人工撰寫的相同應用程式，採用 Rational Business Developer Extension 產生的應用程式具有相近的效率及效能表現。當然，人工撰寫的程式碼經過細部調校，可取得更高效能，但在一般商業應用程式範圍並無明顯差異。

問：我沒聽過 EGL，應該沒有其他語言那麼受歡迎。

答：IBM 在 2006 年春天推出 EGL，目前許多廠商都有很大的興趣。雖然最新的 Web 標準（如 JSF）及 Web 服務等支援都是全新功能，不過 EGL 的技術

主要都是源自於非常受歡迎的熱銷 IBM 工具，像 VisualAge 在全球就有數以千計的客戶。

問：為何要採用 EGL，這不是 IBM 專用的語言嗎？

答：IBM 花了超過 25 年研發快速開發技術，而 EGL 正是這項投資的延續。隨著 Web 及 SOA 等新運算模型的出現，IBM 認為簡化的程式設計方法是組織成敗的關鍵，因此大量投資在此項策略解決方案，並提供充分的支援。許多 IBM 客戶在 IBM 快速開發技術上投資相當多，採用 EGL 對他們而言並不僅是一項前瞻決策，更是一項策略性的快速開發語言，預期將成為業界的通用標準。

問：要如何找到熟練的 EGL 開發人員？

答：IBM 擁有非常多諮詢顧問及商業合作夥伴，藉由他們的專業能力，您的專案將能在短時間內開始進行。幾個禮拜之內，您的內部開發人員就能具備必要技術，不須向外徵求稀少的技術人力資源。

問：程式碼產生器雖能提高生產力，但若過程中出現問題，則在不瞭解產出程式碼的情況下，開發人員將無法確認問題原因。

答：透過 EGL 產生器，開發人員可輕鬆從產出程式碼回溯至原始 EGL 原始碼，並在產出程式碼執行時，以除錯工具一步步檢查符號原始碼。

問：在 Java 方面，程式碼產生器會比人工撰寫更簡單快速，但卻總是需要特別的執行時期，因此很難取得原生伺服器提供的安全、延展性及可靠性。

答：使用 EGL 產生的應用程式可部署為原生 J2EE、CICS 或 IMS 程式，和部署原生人工撰寫應用程式並無差異。而產出程式碼使用的執行時期程式庫，僅是包裝成常用的輔助共享程式碼，也就是所有程式的產出程式碼中，均不會出現的重複部分。

競爭優勢

應用程式開發工具市場非常分散，且正因兩項主要趨勢而進行重大轉型：1) 開放原始碼運動；2) 網路運算及 SOA 運動。市場中充滿許多程式語言及相關的整合開發環境 (Integrated Development Environments, IDE)，每個都宣稱生產力最高且最容易使用，且也都提供許多工具、精靈及程式設計師捷徑及協助功能，包括智慧型編輯器、視覺設計程式、強大的瀏覽器及專案管理工具、互動式除錯

工具、程式碼範本或自動編碼輔助、程式碼產生精靈等。而 Rational Business Developer Extension 在 IDE 方面的強大功能首屈一指，還能充分利用成熟且先進 Eclipse 開放原始碼工具架構。

雖然詳細比較 IDE 功能超出本文討論範圍，且客觀的定位及技術比較必須以完整的準則為基礎，但 IBM 仍堅信，若以最嚴苛的 IDE 要求仔細檢驗，絕對能證明 EGL 的過人之處。

許多組織都認為若要支援其應用程式開發目標，以下要求絕對不可或缺：

- **多用途**：能支援不同應用程式要求，例如可建置的應用程式類型，像是桌面、Web、批次、SOA 等；支援不同目標平台，例如作業系統；支援不同中介軟體（例如資料庫、MOM、交易管理伺服器）。
- **可部署為原生程式**：不須特定專用伺服器，就能將執行檔部署至強大的主流延展式企業交易環境，如 J/EE 伺服器、CICS 等。
- **支援舊式系統**：可輕鬆整合並使用現有應用程式，且可為舊商業系統進行轉型及現代化作業，繼續發揮原有價值。
- **開放性（執行時期）**：支援開放標準及加入 SOA，可建立與其他系統交互運作的應用程式。
- **開放性（開發）**：語言及開發工作平台均可進行擴充。
- **語言**：兩項重要指標，能為開發人員提供多少抽象性及生產力，以及語言運算能力和功能性的完整程度。
- **易於學習**：開發人員要多久才能正式開始作業。
- **開發週期整合**：技術在強大開發週期中的整合能力。

像 EGL 這類技術的吸引力來自其基礎價值定位，也就是易學、易用、生產力高、功能強大，且是針對業務導向開發人員所設計的企業級應用程式/服務開發技術。其他工具若是以複雜的低階語言為基礎，例如 C#、C++ 或 Java，無論功能有多強大，都不會是 EGL 的對手。

評估 EGL 時，應依據上述特性清單來衡量替代工具的技術，我們相信 EGL 在這些方面，會比其他主要競爭解決方案表現更優異。

作者簡介

Stefano Sergi 在 1980 加入 IBM 擔任軟體開發人員，至今已在 IBM 軟體事業處、產品開發、行銷及銷售部門擔任過多種不同職務。作為實際參與者，以及 IBM 北卡羅萊納州 Cary 市實驗室的產品開發經理，他有機會親身體驗軟體開發的困難，之後他加入全球銷售組織擔任 WebSphere Foundation 及相關工具的銷售經

理。他帶領團隊將銷售支援從 WebSphere Studio 過渡到 Rational 產品組合，隨後則負責提升 Rational 品牌的銷售狀況。之後短暫擔任中美洲區 SOA 銷售主管，現在則是 IBM Rational 產品線經理，負責 EGL 及 Rational Business Developer Extension 的相關業務。