



Avril 2008

Qu'est que l'intégrité des processus métier ?2

Variabilité des processus métier inhérente à une architecture SOA :4

Éléments de contrôle de l'intégrité :6

IBM propose une suite logicielle qui assure l'intégrité des processus9

Introducing Process Integrity: Critical to Business Applications, SOA Compositions and Processes10

IBM SOA

Assurer l'intégrité des processus métier avec les Architectures Orientées Services (SOA)

Featuring research from

Gartner®

Qu'est que l'intégrité des processus métier ?

L'avènement des architectures orientées services (SOA) a introduit un couplage faible entre les différents composants métier, et donc qui dit couplage faible dit une perte potentielle de contrôle sur l'ensemble de la chaîne.

L'intégrité des processus est donc la capacité d'intégrer en toute sécurité, fiabilité, flexibilité et transparence dans une architecture SOA, des activités métier informatisées au sein d'un environnement facilement expansible combinant des services exposant la logique métier, des tâches humaines,

l'accès à l'information d'entreprise, et ce tout en assurant la prise en charge des utilisateurs finaux.

De plus, l'évolution des besoins métier fait que ces utilisateurs finaux ont de plus en plus un accès direct à des informations ou des transactions qui peuvent affecter des processus en cours.

Les processus vont donc devoir s'assurer de la cohérence, de la fiabilité, et de la capacité de contrôle des éléments qu'ils vont intégrer, c'est-à-dire des utilisateurs interagissant avec des transactions qui consomment de la logique métier et de l'information.

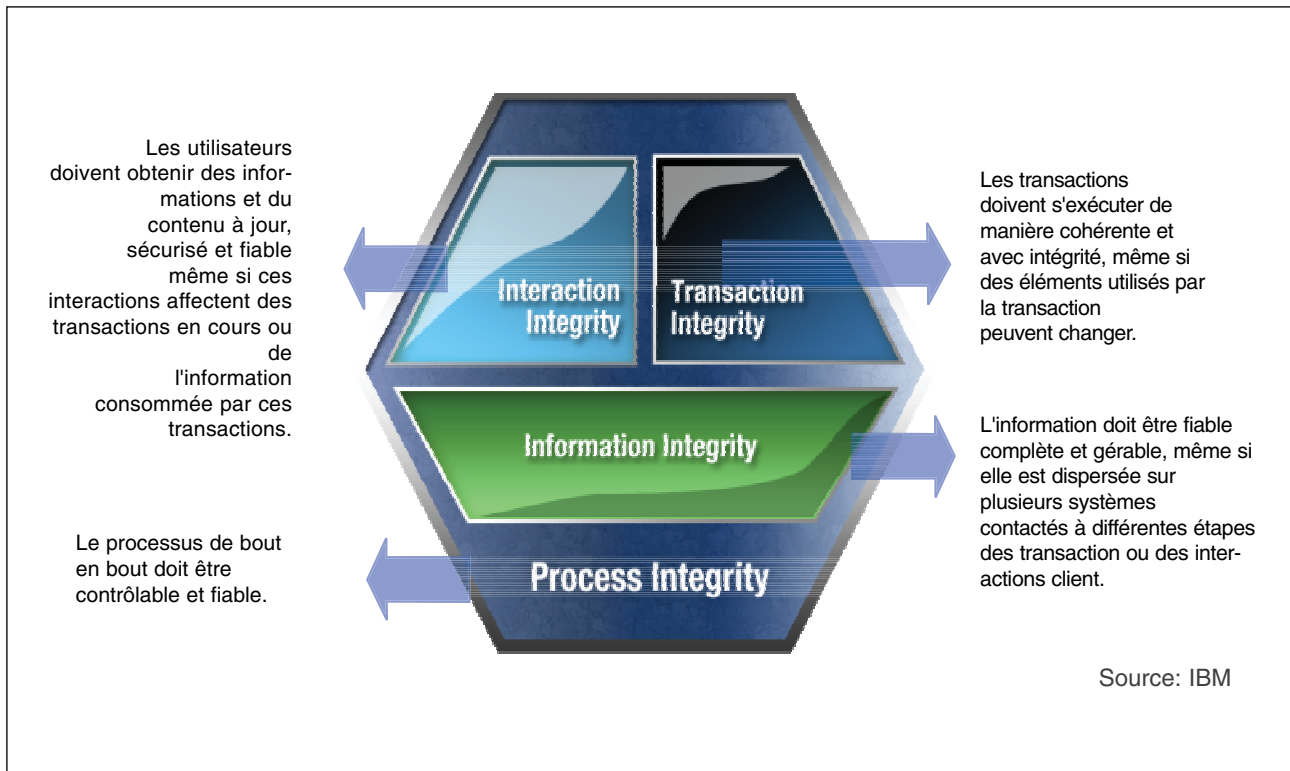
L'intégrité des processus métier répond aux questions posées par l'industrialisation des architectures orientées services :

- **Possibilité de revenir avec intégrité à l'état antérieur en cas de défaillance d'une étape du processus**
- **Accès à une information cohérente à chaque étape**
- **Disponibilité et exécution performante**
- **Sécurité et contrôle cohérent des accès**



Source: IBM

Les différentes catégories d'intégrité



Variabilité des processus métier inhérente à une architecture SOA

En reprenant la définition ISO 9001, un processus est « un ensemble d'activités corrélées ou interactives qui transforment les éléments d'entrée en éléments de sortie ». Il n'y a donc pas de notion de taille de processus ni obligation d'automatisation.

L'approche BPM (Business Process Management) va permettre d'automatiser certaines parties, tandis qu'avec l'architecture SOA, les activités peuvent être fournies par des services exposant une logique métier existante ou nouvelle.

Gérer des bouquets de services évolutifs via un processus de commande unique

Dans l'exemple suivant, nous voyons comment la variabilité d'un processus métier peut s'exprimer chez un opérateur de télécommunication.

L'opérateur propose un bouquet de services (téléphonie fixe, mobile, ADSL, télévision, etc.) que le client peut sélectionner et acheter au cours d'un processus de commande unique.

À chaque service choisi correspond un processus spécifique de mise à disposition. Par exemple, l'attribution d'un numéro de téléphone fixe est associée à une séquence de tâches basées sur un inventaire physique et des règles d'attribution avec des préfixes basées sur la région (01,02,03,04,05). Pour une connexion ADSL, l'attribution d'une adresse IP nécessite l'accès à un inventaire de réseaux IP disponibles et la connexion physique près du domicile de la ligne de téléphone avec l'envoi d'une personne pour effectuer cette connexion.

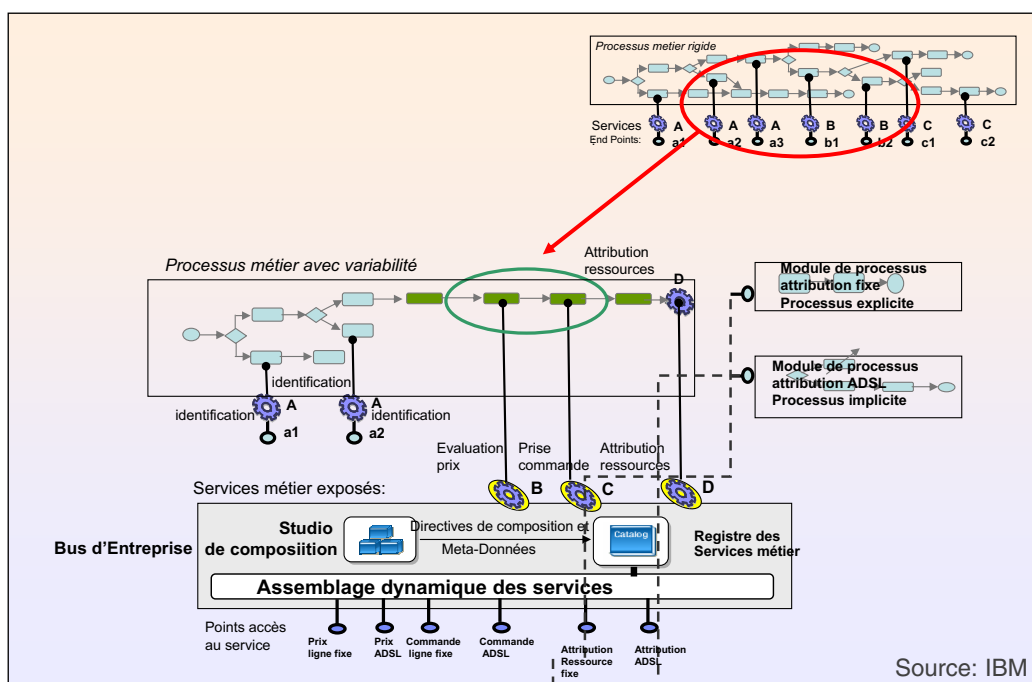
Autrement dit, le processus de commande du/des services par un client, est variabilisé par l'éventail des choix. Par ailleurs, l'opérateur veut pouvoir modifier ses offres et en introduire de nouvelles sans avoir à réécrire tout le processus.

Les séquences variables sont donc extraites sous forme de modules de processus séparés qui seront reliés dynamiquement en fonction du contexte.

Le processus métier va donc être découpé en plusieurs sous-parties, chacune étant stable ou

Les différentes catégories d'intégrité :

Figure 1: Exemple de variabilité de processus métier chez un opérateur de télécommunication



Source: IBM

Variabilité des processus métier inhérente à une architecture SOA

capable d'absorber des variations élémentaires. Les points d'articulation flexible entre les modules du processus métier de bout en bout seront généralement le résultat d'une urbanisation métier de l'entreprise, assurant ainsi un contrôle efficace de chaque sous-partie par une organisation adéquate, ainsi que la cohérence de cette sous-partie.

Changer les wagons d'un train lancé à pleine vitesse

Ces sous-parties, telles un train, vont permettre une flexibilité de l'entreprise et le remplacement à la volée de certains wagons, en fonction du contexte du contrat ou du contenu (modules de processus). Le lien flexible entre les modules va être aussi un service métier que le module tracté doit exposer au module tracteur. On peut ainsi mettre en place un processus de prise de commande capable d'enchaîner indifféremment des modules de processus différents pour supporter les variations dans la réalisation de la commande, suivant le type de produit et accéder à des services métier ou d'information différents à partir de chaque module.

De même, certaines portions du processus sont réalisées par des progiciels d'entreprise spécialisés dans la gestion de la relation client ou la réalisation de la chaîne d'approvisionnement ou de livraison.

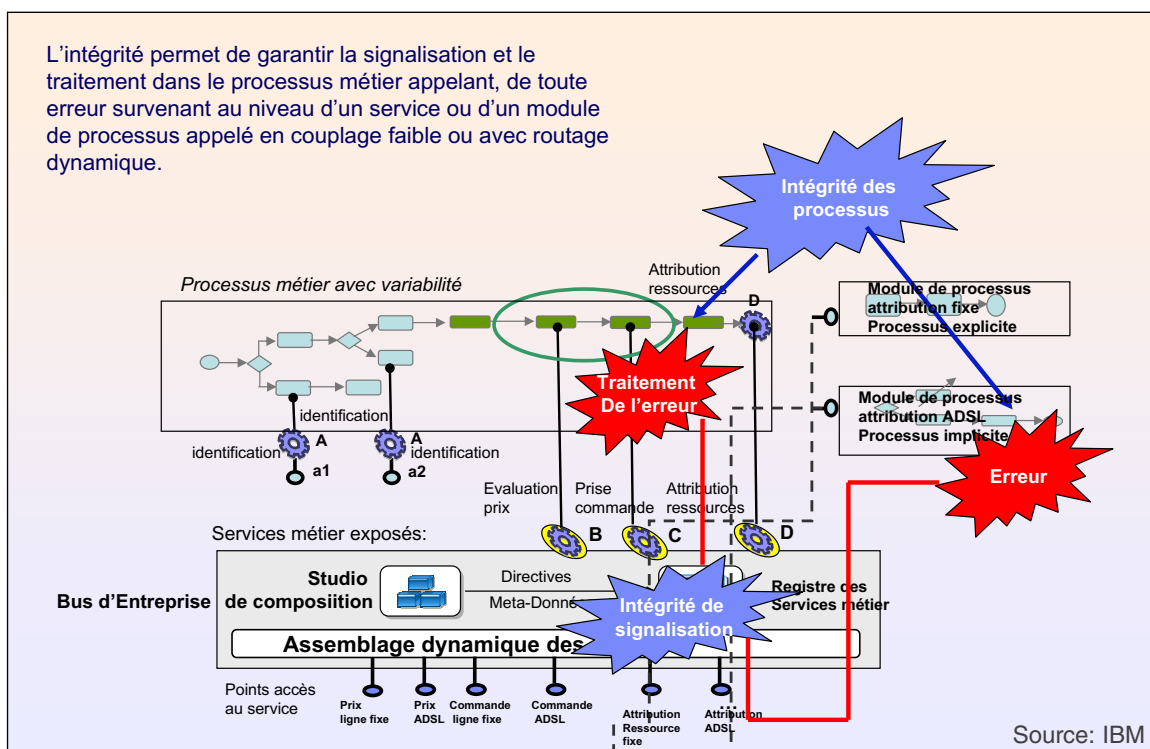
Ce couplage faible entre parties de processus, services métier, d'information, de logique ou de tâches humaines introduit énormément de variabilité dans les processus, ainsi que dans les services métier intégrés par ces processus.

Retrouver et réparer les erreurs

Le contrôle de l'intégrité et la possibilité de piloter et de prendre des actions correctives en cas de problèmes devient donc un élément essentiel de la réalisation de la valeur.

On est donc de plus en plus confronté à des processus dont la réalisation est implicite par l'association de morceaux de logique et d'interactions explicites.

Source: IBM



Éléments de contrôle de l'intégrité

Les éléments de contrôle de l'intégrité vont donc se situer dans chaque sous-partie, interaction, information et transaction, mais aussi dans les éléments de logiciel middleware reliant ces parties entre elles.

Intégrité de l'interaction

L'intégrité de l'interaction assure aux clients finaux l'accès aux processus et à de l'information au contenu à jour, sécurisé et fiable et cohérent.

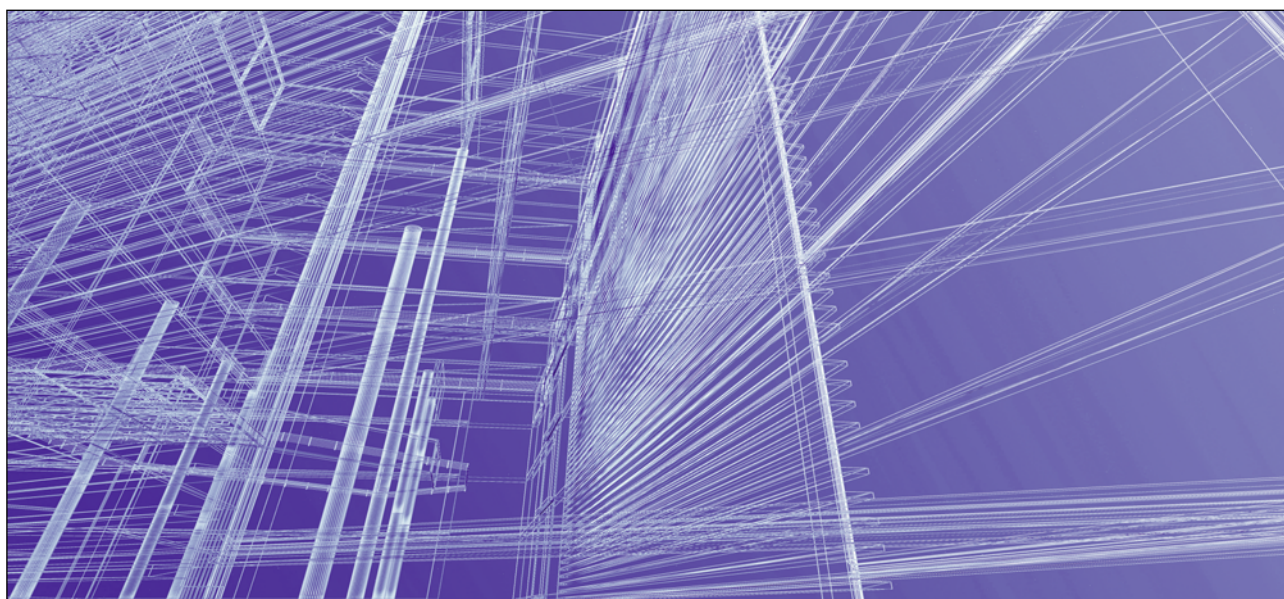
- Cet aspect assure que les interactions des personnes avec les processus critiques SOA se déroulent avec intégrité et la capacité de montée en charge appropriée quelque soit le moment, le canal ou le lieu de cette interaction.
 - Il faut donc pouvoir gérer la fourniture de contenu en fonction du rôle de la personne, ainsi que de la tâche à accomplir dans le contexte métier propre à l'interaction.
 - Il faut fournir des vues customisées et personnalisées pour les divers acteurs tels que les partenaires, les employés et les clients.
 - Enfin, il faut permettre un accès interactif et flexible aux processus et au contenu couvrant l'ensemble des sources et des moyens physiques d'accès.
- La gestion intégrée des identités des utilisateurs est un aspect clé pour la sécurité des processus. Dans des processus impliquant des partenaires dans des entreprises différentes, la fédération des identités avec des référents au sein de chaque entreprise est essentielle.

Intégrité de l'information

L'intégrité de l'information assure aux processus et aux clients finaux l'accès à une information fiable, sécurisée, quelque soit le canal par lequel cette information est consommée.

Dans la plupart des entreprises, l'information provient de plusieurs sources, celles-ci étant hébergées sur des systèmes et des plate-formes multiples, avec sur chacun de ces systèmes des contextes sémantiques, des formats et des origines différents. De surcroît, la plupart de ces systèmes ont leur propre interface et leurs règles métier pour gérer cette information. C'est dans ce contexte complexe par nature que les technologies employées pour fournir une information intègre doivent opérer. Dans une architecture SOA, les éléments suivants (information – contenu – données) répondent à ces besoins :

- Des services d'intégration de l'information pour récupérer en temps et en heure une information cohérente, précise et dans le contexte approprié à partir de plates-formes diverses et variées, et délivrer cette information en une vue unique et



Éléments de contrôle de l'intégrité

fiable avec des services métier. Les services de « nettoyage » assurent aux utilisateurs le rapprochement et la fiabilité de l'information.

- La synchronisation sécurisée des systèmes de « master data management » avec des systèmes sources ERP, GRC/CRM, ainsi que ceux développés en interne dans les entreprises.
- Des bases de données résilientes ayant un support élaboré pour la gestion et l'implémentation de la gouvernance et des stratégies d'accès (« control policies »).
- Des services de contenu pour délivrer l'information provenant de contenu non structuré.

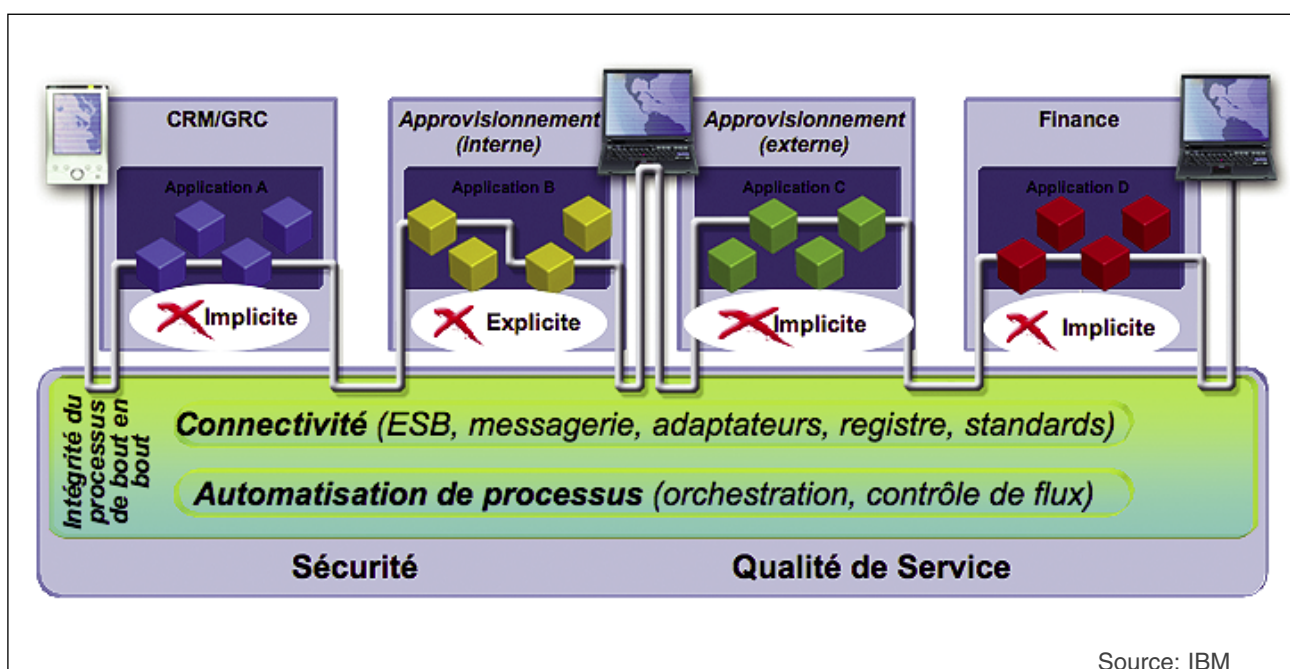
L'intégration de ressources distribuées dans un environnement à couplage faible des architectures orientées service nécessite le même niveau de contrôle que pour les transactions fournies traditionnellement. Ceci concerne à la fois les mises à jour critiques de chaque unité d'oeuvre, mais aussi les transactions longues en support des processus métier.

L'implémentation d'une infrastructure de support pour votre SOA exige que les modifications des ressources effectuées par les applications et les bases de données, et l'accès à ces ressources via un couplage faible tel que le bus de service d'entreprise, soient coordonnés et validés dans le cadre d'un contrôle effectif, complètement intégré dans le middleware permettant le SOA.

Intégrité des transactions

Les lignes métier doivent pouvoir se reposer sur l'intégrité des transactions qu'elles engendrent et traitent. En effet, l'impossibilité de finaliser correctement des transactions impacte la confiance des clients, ainsi que leur satisfaction. Les approches traditionnelles au traitement de l'information ont jusqu'à présent réussi à fournir un tel contrôle transactionnel dans des environnements très fortement contrôlé et intégré.

Les processus métier sont des séquences de transactions reliées entre elles par un flux coordonné. Dans ces processus, chaque activité ou événement, ou chaque requête client peut être décomposé en transactions élémentaires dont il faut contrôler la séquence et coordonner l'ensemble.



Il est également nécessaire de contrôler les chemins d'exception ou d'erreur, en permettant une récupération cohérente de l'état des ressources affectées par les transactions du processus, en autorisant des chemins de rechange dans le cas de changements survenus dans des séquences qui ne sont pas instantanées ou de changements irréversibles au sens technique, qui doivent être inversés au niveau métier. Par exemple, en cas de problème, l'inverse de la création d'un compte peut être son inactivation afin d'en garder une trace à des fins d'audit et non sa suppression pure et simple qui effacerait toute trace.

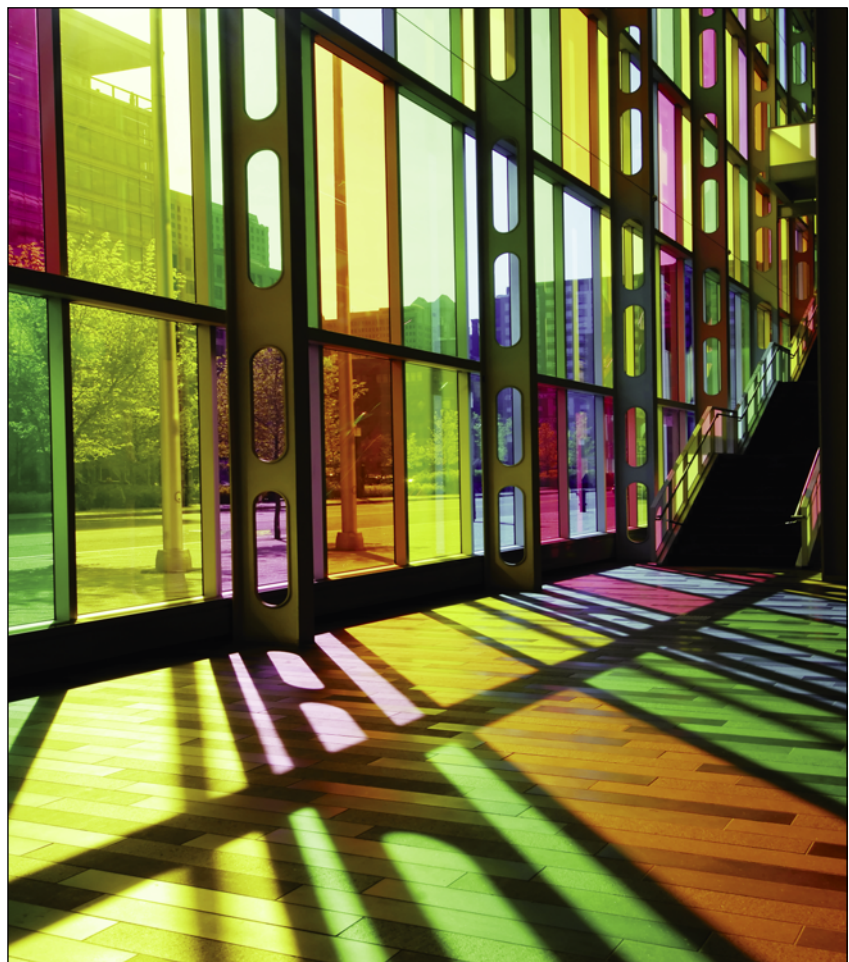
Intégrité des processus

Les processus métier en environnement vont donc mettre en œuvre à la fois des transactions, de l'information et des interactions en les reliant via un couplage faible.

Il faut donc être capable de suivre l'évolution du processus, qu'il soit explicite dans une portion s'exécutant dans un moteur de processus orchestrant les personnes et les services, ou implicite dans un progiciel ou package, tel que ERP ou GRC. Cela implique la capacité de modéliser le processus métier de bout en bout et ce, pour les portions à la fois implicites et explicites, afin de mettre en place un contrôle de ce processus effectif résultant qui permette de prendre des actions correctives et de qualité, en particulier lorsque des problèmes surviennent dans les parties de processus implicites.

Le processus modélisé est donc un sur-ensemble représentant la chaîne à contrôler, qui permet de définir des événements que le contrôle pourra capturer pour prendre des actions spécifiques.

Source: IBM



IBM propose une suite logicielle qui assure l'intégrité des processus

La technologie middleware d'IBM inclut un contrôle transactionnel intégré, ramenant le risque à une bonne conception des transactions. IBM propose une gamme de produits logiciels qui fournissent et soutiennent l'intégrité des transactions dans une architecture SOA, offrant un contrôle de bout en bout de la mise à jour des ressources.

IBM a une longue expérience reconnue dans le domaine transactionnel d'entreprise, disponible depuis longtemps avec les moteurs transactionnels CICS et IMS, ainsi qu'avec le serveur d'application WAS, le moteur de messagerie WMQ et les autres produits construits autour de ses moteurs transactionnels principaux. Ces produits fournissent le support aux environnements transactionnels et répondent à toutes les exigences SOA, que ce soit pour les transactions localisées ou distribuées.

- Produits clés pour l'intégrité des interactions
 - WebSphere Portal Server
 - WebSphere Portlet Factory
 - Lotus Forms
 - Lotus Expeditor
- Produits clés pour l'intégrité de l'information
 - IBM Information Server
 - IBM Master Data Management Server
 - IBM Content Manager, IBM FileNet P8
 - DB2, IMS, Informix Dynamic Server
- Produits clés pour l'intégrité des transactions
 - WebSphere Process Server, WebSphere ESB,
 - WebSphere Application Server
 - WebSphere Message Broker, WebSphere MQ
 - WebSphere DataPower Integration Appliance XI50
 - WebSphere Service Registry and Repository
 - WebSphere Adapters
 - CICS Transaction Server
- Produits clés pour la qualité de service
 - Tivoli Composite Application Managers
 - WebSphere Application Server
 - WebSphere XD
 - WebSphere DataPower SOA Appliances
 - Rational Performance Tester Extension for SOA Quality
 - IBM Systems Servers

Source: IBM

Introducing Process Integrity: Critical to Business Applications, SOA Compositions and Processes

We introduce the concept of process integrity and its relationship to next-generation computing and business applications. Those who ignore process integrity for emerging distributed, service-oriented, composite applications will fail to deliver a reliable, trusted user experience to their internal and external customers.

Key Findings

- Process integrity is composed of interaction integrity, data integrity and transaction integrity.
- In next-generation applications, designers may not be able to guarantee a trusted, reliable user experience because they will be delivering services that users can compose into processes. Thus, consider integrity when creating dynamic compositions.
- Business applications had ensured process integrity inherently, but no longer can as the

process shifts from being implicitly delivered by vendors to explicitly managed by users.

- Computing styles will indicate which style of integrity (interaction or transaction) is more important to project success.

Recommendations

- Make process integrity a priority in designing service-oriented architecture (SOA)-based compositions or outsourcing business processes via alternative delivery models, such as software as a service (SaaS) or business process outsourcing (BPO).
- Focus on ensuring interaction integrity, rather than just distributed transactions.
- Don't assume that the definition of the process relates only to the business transaction; rather, use the new SOA-based models to build human interaction into the process.



WHAT YOU NEED TO KNOW

In a distributed ecosystem where businesses will share software services, build new processes, include human interaction in them, and outsource processes where integrity cannot be guaranteed. The traditional approaches to creating process integrity through well-defined transaction and system boundaries may no longer be applicable in the new world of applications. In most cases, designers will not determine the end-to-end process experience. Instead, each execution flow is likely to be unique and driven more by the end consumer's experience. The need for process integrity must be viewed through multiple perspectives that include process, which maintains integrity through human interaction, as well as machine interaction.

STRATEGIC PLANNING ASSUMPTION(S)

Through 2010, SOA, SaaS, BPO and business application implementation projects that don't make process integrity an integral part of the process design and implementation will fail to deliver the expected results.

ANALYSIS

Process integrity is not guaranteed in the design of application compositions. Business applications and SOA compositions depend heavily on a broad view of what process integrity means to the end users (external and internal process participants) and the means of enabling it when necessary. Composition environments will be how applications are built. Process integrity in next-generation application design (SOA, compositions, SaaS and so forth) will be a major success criteria.

Broken End-to-End (Horizontal) Processes

A critical area for process integrity is high-level business processes that cross multiple business silos. These processes are broken up in a way that makes it difficult to manage them as a single process. Integrity is threatened because processes must be extended into different organizations, and sometimes across different technologies. SOA makes this even more critical because potentially all SOA implementations will be broken up this way. Examples

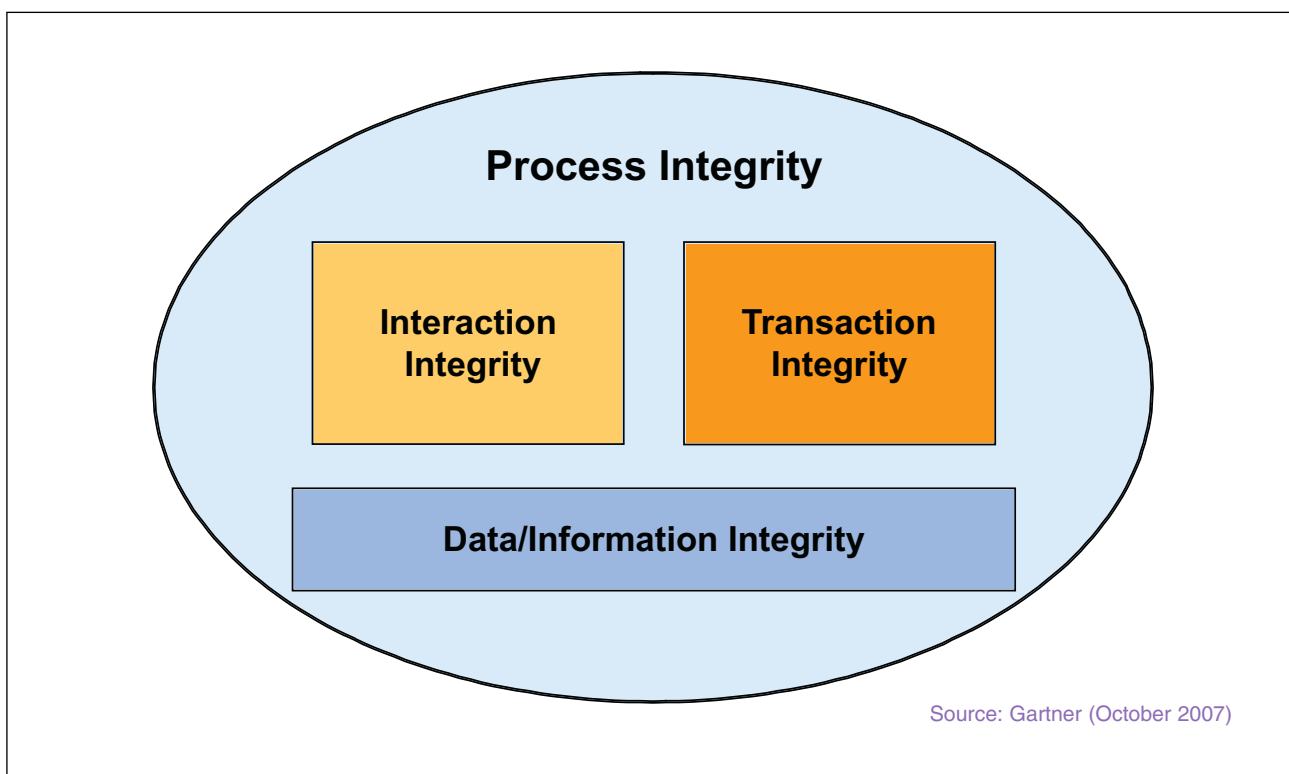
are order-to-cash cycles, invoice processing and three-way matching in purchasing.

Users of packaged business applications expect that, as they execute processes, the integrity of the interaction, transaction and data is built in. For example, if a purchase order is entered and the user chooses to create an invoice from that purchase order, then the purchase order details would remain intact as they were entered. With new alternative application consumption and development models, process integrity is not guaranteed; therefore, users who adopt SaaS, SOA, next-generation service-oriented business application and composite application platforms must view process integrity in a new light.

What Is Process Integrity?

Process integrity is the degree to which all process elements – tasks, users, systems, information and so forth – are intact and functioning in ways that ensure sustainability and long-term adaptation to changing conditions and human uses (see Figure 1).

Figure 1. A View of Process Integrity



From airline reservation systems to billing systems to mortgage calculation systems, the integrity of the process must exist beyond a certain degree of sustainability. Below this threshold, the process will fail or will deliver uncertain results that cause the overarching business function to fail or degrade.

Software infrastructure must provide transaction models and mechanisms that help ensure that process integrity – through transaction integrity, data integrity and now, increasingly, interaction integrity – is maintained. Packaged applications are a mainstay of modern business and have demonstrated sufficient (if not always stellar) process integrity. For example, if a purchase order is entered and the user chooses to create an invoice from that purchase order, then the purchase order details would remain intact as they were entered. With new alternative application consumption models, process integrity is not guaranteed.

Strategic Planning Assumption: Through 2010, SOA, SaaS, BPO and business application implementation projects that don't make process integrity an integral part of the process design and implementation will fail to deliver the expected results.

Distributed systems and distributed companies have increased the need for distributed transactions that have different process integrity requirements.

Changes in Next-Generation Applications

What is changing with next-generation applications that affects process integrity?

- **Implicit vs. Explicit Process Models:** Processes in most packaged business applications are implicit. The application provider designs and delivers the process. Users of these application suites are forced to make decisions during the implementation process of "change the business process as we execute it or change the application." A major change coming with SOA is the ability to manage the process explicitly and totally separated from the actual application design. A basic premise of SOA is that tasks are exposed as services, which users can rearrange.

But process integrity will be difficult to guarantee with explicit process models because there will be many permutations and combinations of process outcomes, and those possible outcomes can be manipulated.

- **The Rise of SOA:** The assumed integrity of traditional packaged applications is not preserved when a variety of composition elements or services are thrust into a joint relationship. As vendors and users embrace SOA for integration and process flexibility, the use of services becomes more prevalent. Although standards for building services (SOAP and Web Services Description Language) are maturing, standards that define the process have not matured to the same extent. For example, a single process provided by one vendor will have the ability to interact with a process from another vendor because the vendors used the same standards to build the services. However, they may not have used the same granularity to define the interaction, transaction and data to guarantee process integrity. Until we have standards for all the process attributes that express the end-to-end business semantics of the workflow, process integrity delivered via services developed by multiple parties will be difficult to guarantee from vendors; and users that are creating the composition must be in control of their own process integrity.
- **Growth of the alternative software consumption models:** SaaS enables service providers to host or operate processes for a business, rather than the business having to do it. In addition, SaaS providers will increasingly need to work with processes that touch data sources, applications or other processes that the SaaS provider does not solely run or manage. Adopting these solutions requires users to ensure that they have defined the end-to-end process needs to ensure process integrity.
- **Compensating transactions vs. rollbacks:** The compensating transaction is an action taken to reverse a prior transaction step. This has long been used when transaction integrity needed to migrate outside the boundaries of the original

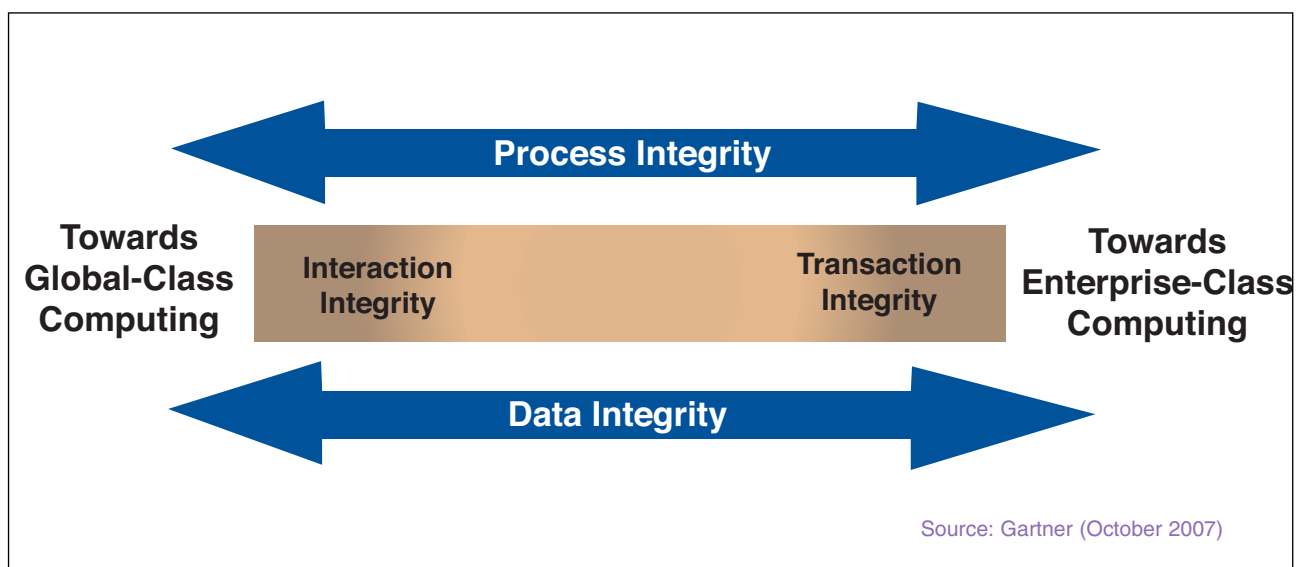
transaction. That is, a compensating transaction becomes necessary when the process progresses beyond the point that a rollback can reverse an action or if the transaction took place at a different time than the necessary correction (for example, refunding incorrect orders).

- **External (rather than in the packaged application) master data management (MDM):** Packaged application users are forced to define customers, vendors and charts of accounts during the implementation process. These master records are used as the process is executed to ensure data and transaction integrity in the integrated application. In many cases, ERP implementations were a source of managing master data. The myth of ERP was that you could buy everything from a single vendor; most users know that you can't, which makes managing master data within a single vendor suite impossible. MDM technologies are emerging to support multienterprise, multitechnology processes. Although MDM helps with the master data issues, the interaction and transaction issues associated with process integrity remain.
- **The requirement for "interactions" in addition to "transactions":** Rapid growth of the Web and Internet business applications has spurred the growth of interactions. The interaction is about getting the user's view of the process

accomplished. For example, getting a ticket for a flight is not about a single transaction but about a number of interactions between user and provider to specify preferences and to acquire proper seating. Integrity is guaranteed through an implied contract across multiple transactions and uses compensating transactions to ensure integrity. The "interaction" will be the primary unit of work for consumers of services and the applications that support them. This will force IT organizations to highlight different process requirements, compared with online transaction processing. The transaction is still important, but the ability for a user to work with content to interact with agents, collaborate in process solutions and sometimes handle their own compensating transactions will be key to success or failure. In future research, we will detail specifics about the interaction and how it changes the process integrity approach.

- **The impact of the "process of me":** The next stage in the evolution of application support for business processes is to move beyond the enterprise view of the process to encompass the true needs of the individual as a participant in multiple processes. The individual view of process has been largely ignored in creating a business application to meet users' needs. Personal productivity tools – such as e-mail,

Figure 2. Computing Styles Affect Integrity and Decisions





instant messaging and the onslaught of consumer-based technology – all have a role in how the user executes a process. As we begin to enable the person to control the design process (and possibly even build it), process integrity must be a design consideration.

Make Process Integrity a Priority for Different Styles of Computing: Rollback, synchronous process blocking and data locks are not the preferred solutions for reversing a process or ensuring integrity in the next-generation applications. Users may choose to create services or composite applications that exist in different systems or operate in multitenant environments (requiring multi-instance data and services), or even that execute outside the control of one process and system model. The use of asynchronous mechanisms – such as compensating transactions, system restore, synchronized data in multiple data stores, federated transactions and federated data – will become more critical. This is in line with a particular set of computing trends (see Figure 2). Different styles of computing will tend to indicate specific process integrity needs in a project. We break computing into two major styles for the sake of this discussion: global-class computing (GCC) and enterprise-class computing (ECC).

GCC is a Gartner description (from 1999 and kept current) of the evolving nature of computing that is less about control and more about coordination across companies, consumers, content, capabilities, contributions and communities. It is more about the freedom to individualize. The Web fits this description, as does the increasing consumerization of IT decisions in enterprises.

ECC is about control and normalization of computing to fit the viewpoint of a given enterprise. Scalability, reliability, serviceability and availability are often the key criteria. It is about scaling up, not down, and is focused inwardly rather than outwardly. In the enterprise, traditional transactions are still "king." The transaction is about the system's view of the process intended to ensure that each operational step is accomplished without losing integrity. Transactions can be long running or short; but, in all cases, the flow through system-level operations is read, write, update and commit. Integrity is guaranteed through rollbacks, commits and locks on data.

These two styles of computing are not mutually exclusive; they are complementary. Although GCC is growing rapidly, ECC remains a critical part of large-scale system design and deployment. The two styles will continue to evolve together and, ultimately, become so intertwined that supporting characteristics of both will be the norm rather than the exception.

Changing Requirements Lead to Changing Governance Assumptions

GCC challenges basic assumptions that have been around for decades. From user interface requirements (going personal rather than general), to tolerance for application errors (users now reload pages when they see an error rather than calling IT), GCC highlights the need to change our approach to process. Process integrity is no different. The biggest question is: "With distributed SOA, don't we need distributed transactions to ensure process integrity?" The answer is, possibly, "No."

Contractual agreements between companies cooperating on a globally distributed scale have been around far longer than have modern computers. These companies have handled distributed business by building safeguards into contracts, relying on human managers to adjust accounts and relying on compensating transactions to correct errors. This does not imply a wholesale shift away from using computing to automate these functions. Instead, it heralds a more broad-ranging approach to process integrity. As GCC and next-generation applications/compositions continue to be deployed, there will also be a need to ensure that the process models governing the behavior of the processes involved will also be paired with governance technologies that:

- Check integrity when changes are made
- Monitor processes to detect issues before failure
- Enforce policies that limit damage that can be caused by a distributed process failure
- Use automated recovery systems to correct problems after the fact

Gartner RAS Core Research Note G00152395,
Yvonne Genovese, Daryl C. Plummer, 23 October 2007



Architectures orientées services :

<http://www-306.ibm.com/software/fr/soa/?ca=homepage=w&met=websphere>

BPM :

<http://www-306.ibm.com/software/fr/soa/launch/bpmsoa.html>

IBM Software TV :

<http://www.softwaretv.com/web/guest/home>

© Copyright IBM Corporation 2008. Tous droits réservés .

IBM, the IBM logo, ibm.com and DB2 are registered trademarks of International Business Machines Corporation in the United States, other countries, or both.. DB2, Lotus, Rational, WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.

Cette publication d'IBM est fournie à titre informatif. Les informations qu'elle contient sont sujettes à modification sans préavis. L'éditorial IBM et l'analyse Gartner sont publiées respectivement par IBM et Gartner, sous leurs responsabilités respectives, aucune des deux n'endossant la responsabilité du contenu établi par l'autre.

Assurer l'intégrité des processus métier avec les Architectures Orientées Services (SOA) is published by IBM. Editorial supplied by IBM is independent of Gartner analysis. All Gartner research is © 2008 by Gartner, Inc. and/or its Affiliates. All rights reserved. All Gartner materials are used with Gartner's permission and in no way does the use or publication of Gartner research indicate Gartner's endorsement of IBM's products and/or strategies. Reproduction and distribution of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Gartner disclaims all warranties as to the accuracy, completeness or adequacy of such information. Gartner shall have no liability for errors, omissions or inadequacies in the information contained herein or for interpretations thereof. The reader assumes sole responsibility for the selection of these materials to achieve its intended results. The opinions expressed herein are subject to change without notice.