The Business Feedback Loop
When discussing information within a plant, we often see an emphasis on ‘visualization’ or ‘presentation’ of that information. However, the actual business is a loop with the business goal of setting process production targets for the operational business processes to follow by causing actions on the plant assets and materiel, so that measurements of the state of the assets and material can be reported back in order to determine if the business is meeting its goals.

The principles of this business loop are no different than any other feedback control system: the quality of control, in this case the ability to meet the business goals, depends on the accuracy of the measurements and the frequency with which corrective action can be taken.
Table of Contents

The Business Feedback Loop ........................................................................................................ 1

What Questions Do You Need to Ask to See if Your Business is Well-Tuned? ..................... 4
   Do you trust your performance reports enough to bet your freedom on them? ....................... 4
   Your business processes will evolve. Can your information systems evolve to match, or will they drag you down? ........... 4
   Can you afford the risk of the ‘big-bang’ change in your information systems? ....................... 4
   Where do you go for intelligence about your plant? .................................................................. 4
   Is your information treated like a museum piece, trapped in a glass case? ............................. 4
   Business performance measures are complex, involving most of your operating systems. How do you get these numbers? ................................................................. 4
   Which technologies solve my problems: messaging, data warehouses, portals, or some other technology? .......... 4

Production Application Gap Between Measurements and Corporate ........................................ 5
   Application Proliferation ............................................................................................................. 6

Integration Using Listeners: Web Service Orientated Architecture ......................................... 7
   Informal Processes ....................................................................................................................... 8

Information: The Intelligent Plant Data Model ........................................................................... 9
   Data vs. Knowledge .................................................................................................................... 10

Data Model Standards ................................................................................................................. 10
   Data Damage ............................................................................................................................. 10

Warehousing Information ............................................................................................................... 11

Virtual Calculations ..................................................................................................................... 11

Operational Data .......................................................................................................................... 11

Presentation: Operational Insight plant portal and applications .............................................. 11
   Superfluous Functionality .......................................................................................................... 12

Putting it all together ..................................................................................................................... 12
   Integration ................................................................................................................................. 12
   Information ............................................................................................................................... 12
   Orchestration ............................................................................................................................. 13
   Presentation ............................................................................................................................... 13
   Application ............................................................................................................................... 13

Closing the loop: Applications, integration, presentation, and information as a means of supporting the business .... 14

Path Forward ................................................................................................................................ 15
# Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1:</td>
<td>Business goals</td>
<td>1</td>
</tr>
<tr>
<td>Figure 2:</td>
<td>Information pyramid</td>
<td>5</td>
</tr>
<tr>
<td>Figure 3:</td>
<td>Production systems gap</td>
<td>5</td>
</tr>
<tr>
<td>Figure 4:</td>
<td>DCS/Real Time Historians/Operational Systems</td>
<td>6</td>
</tr>
<tr>
<td>Figure 5:</td>
<td>DCS/Flow of information through ERP</td>
<td>6</td>
</tr>
<tr>
<td>Figure 6:</td>
<td>Information flows between applications</td>
<td>7</td>
</tr>
<tr>
<td>Figure 7:</td>
<td>Web Service Oriented Architecture</td>
<td>8</td>
</tr>
<tr>
<td>Figure 8:</td>
<td>Intelligent plant model</td>
<td>9</td>
</tr>
<tr>
<td>Figure 9:</td>
<td>Integration of systems</td>
<td>12</td>
</tr>
<tr>
<td>Figure 10:</td>
<td>Business goals and integration</td>
<td>14</td>
</tr>
</tbody>
</table>
What Questions Do You Need to Ask to See if Your Business is Well-Tuned?

1. Do you trust your performance reports enough to bet your freedom on them?
SOX now expects all performance measures in financial reports to be auditable. The financial numbers are managed by the ERP system, but many measures of performance are derived from the plant’s throughput, yields, qualities, losses etc. You are now liable for their accuracy.

- How are these factors calculated?
- What data is used for the calculation?
- Where does this data come from?
- Who has been changing these, when and why?

2. Your business processes will evolve. Can your information systems evolve to match, or will they drag you down?
Most business applications have a very ‘hard-coded’ business process, and view of the underlying data. As your business grows, these hard-coded business processes will prevent you from adapting. For example, your current business systems report end-of-month performance, but that is ‘regret-reporting’. You want real-time reporting, but those systems do not permit it.

3. Can you afford the risk of the ‘big-bang’ change in your information systems?
Your primary business is processing, not running an IT department. You are comfortable with the process side of things, but the IT side scares you—with good reason, because many projects go off the rails.

4. Where do you go for intelligence about your plant?
If you have a complex business question such as ‘Given our current schedule and projected liftings, how much premium gasoline will be available next Friday?’, where do you go for the answer? Invariably you go to the resident ‘expert’ who has the knowledge of where to go to get the data to fulfill this request. He will then throw a spreadsheet together and deliver the answer. What happens if this expert disappears? Is this knowledge a corporate asset, or in the heads of such experts?

5. Is your information treated like a museum piece, trapped in a glass case?
Managers do not move valves. No do operators move valves; they change data so that the system moves valves. So when one sees a problem, the corrective action means changing some information. For example, there is information that a shipment will be late. This means changing the operating schedule, which means changing the operating instructions. Performance dashboards, portals, etc do not allow you to change the underlying data.

6. Business performance measures are complex, involving most of your operating systems. How do you get these numbers?
Any measure of business performance is complex. For example, ‘Net Unit Operating Margin’ probably involves most of your operating systems. The information is not just objects and attributes. It will need event information, such as material movements, unit availability, shutdowns, material grades, quality, and flow-rates. There are two problems: assembling this information, and making sure the information is correct.

7. Which technologies solve my problems: messaging, data warehouses, portals, or some other technology?
All vendors tell you they solve all of your problems. In fact, you need them working harmoniously.
Production Application Gap Between Measurements and Corporate

We are all familiar with the 'information pyramid' in which we hope that information flows freely from the corporate systems where the goals are set, through the logistics systems, and manufacturing control down to the control systems. Similarly the measurements of performance flow from the instruments back through the application systems to the corporate systems where targets can be compared with actual performance.

The reality is somewhat different because of the 'Production Systems Gap' that exists between the corporate and control systems. Corporate systems, once numerous independent applications, have been replaced by ERP systems (Oracle®, SAP® etc). Similarly the days when each site wrote their own real-time historian have long been replaced by de-facto standard historians using OPC to communicate with the instruments. The problem is the 'middle' territory between these two layers, filled with numerous, largely independent applications. Information flows from the corporate goals to the control actions via a circuitous route of email, memos, and verbal communication. Similarly, ask yourself how many spreadsheets the actual information has to pass through before it reaches the corporation's annual report. To make matters worse, many of these information flows are not audited.

How did we get here? We got there through acquisition of software applications not part of a corporate plan, yet still essential to the business, through well-meaning skunk-work development of spreadsheets and personal databases to solve real business problems, through mergers of sites, and numerous other factors.
How bad is the problem? We typically find that each operating site has 50-150 independent applications within this Production Systems Gap. Not all of these are the well-known production applications: laboratory system (LIMS), document management (EDMS), or maintenance management (MMS). However it does not mean that the others are any less critical for running the business. Each of these applications duplicates data from other systems, causing integrity problems, as well as incurring the cost of support and training to maintain these systems.

Application Proliferation

With 50-150 distinct applications per site, building and managing interfaces between applications consumes a large proportion of the IS department's time and budget. This leaves little for new initiatives. Users, frustrated by the inability of IS to integrate these applications, resort to creating their own spreadsheets coupled with undocumented copy/paste procedures to 'integrate' the information. In no time at all we have our 30,000 or more 'gagging' of spreadsheets per site.

Not all of these are formal applications. It may be the ubiquitous spreadsheet constructed to solve a particular reporting or analysis problem, gradually creeping into becoming an essential application. Often these are all shared on the central file server. If you are not convinced, count the number of spreadsheets stored there!

Do we really need all of these applications to talk to each other? If we follow the flow of information from Zone 3: the ERP, through to Zone 1: Control Systems, it starts out as targets for the planning systems, which then get converted to schedules, and finally operating instructions. Each stage is dependent on the previous, as well as on information on the current state of the plant.
Measurements are then gathered to monitor unit, inventory and logistics performance on a timely basis. However, to meet our performance analysis requirements we need to consolidate the raw data into yield reports and verify the accuracy via data reconciliation and balancing. Finally, with the confidence this gives us we can use this information as the basis for plant material and cost accounting. This illustrates how the information flows from one application to another rather than acting like icebergs floating around the sea of information.

Even this is not a complete picture, since we need to constantly feed back the ‘results’ to the targets via various production loops: target setting, non-conformance monitoring or key performance monitoring. Following the classic principles of good feedback control, the quality of control depends on the accuracy of the measurements and the frequency of the detection of the deviation. However comparing scheduled versus actual yields is not as easy as comparing two measurements: there is a considerable amount of information transformation necessary to get the measurements into a form that can be compared with the schedule. It is no wonder that many organizations cannot perform this feedback on a regular basis.

From this we must conclude that it is just as critical for a successful business to ensure the free flow of meaningful information between these production applications as it was at the ERP or Process Control ‘zones’.

**Integration Using Listeners: Web Service Orientated Architecture**

The first step in solving this problem is to make sure we can communicate with all of these 50-150 applications. Unfortunately, real-time measurement standards such as OPC do not solve this problem, as we want to access more complex information—for example a material movement with a quantity, source and destination, or a maintenance work order with associated tasks. Additionally, many of the systems with which we want to communicate only have a rudimentary programmable interface (API). This problem of making applications communicate with each other is not trivial. It is not uncommon for interfacing to consume 40% of a project budget. On going maintenance of these ad hoc interfaces also consumes a large share of the ongoing IT maintenance budget.

Our solution is to place one of our Listener modules next to each of the applications with which we want to communicate. The Listener provides a web service façade onto that system regardless of the system’s API.

A Listener is a standalone program or process running on the same machine as the target application. Usually we have one Listener per application. Every Listener is identical; the only difference is the configuration of that Listener to communicate with the target application. This also means each Listener can present a different WSDL, which defines the operations that are available for access by an external system.

So how does a Listener work?
A Listener operates as both a SOAP web service server and client. As a web service client we can invoke an operation on the Listener. When a SOAP message is pushed to the Listener (using standard web service calls), the message is received, decoded and the corresponding script code is retrieved from its configuration. It then executes the script, passing on the contents of the incoming message. Thus this script can use the available API on the application to store data in that application or some other operation. These scripts form part of the configuration of the Listener; they are easily configured and managed by the Listener's built-in debugger and tester.

**Informal Processes**

Frequently, when asked the procedures for rescheduling, say, a material shipment the response will be ‘the scheduler is new to the job so they have yet to settle on their procedures.’ Well-orchestrated processes are one of the few competitive advantages remaining. Thus they should not be left to an individual. Unfortunately ISO-9000/1 manuals might be gathering dust on the shelf. What is really needed is the ability to enforce the business processes, but also easily adapt when the process can be improved.

Immediately we see one advantage of the Listener: we can disguise the differences between functionally identical applications behind a common interface.

A more normal application is when we request information from the application system. In this case the incoming message initiates the execution of the corresponding script which in turn fetches the information from the application system using the available API. The script then assembles the response to return to the calling application.

However we do not always want to fetch data on demand: sometimes we want an application system to automatically notify other systems when certain events occur. Unfortunately most systems cannot do this, so we use the built-in scheduling functionality of the Listener to schedule tasks to run. For example, we could schedule a task to periodically check the application system to see if there are particular data conditions. If such conditions exist, then the Listener script can act as a SOAP client and invoke an operation on another SOAP server, such as an email system or workflow automation system (BPEL).

Finally, some systems have the ability to generate their own events. Many of our systems are distributed around the organization, but we do not want to lose these events. Thus the application system can invoke operations directly on its local Listener. The Listener will then forward these messages to the appropriate SOAP server. If that server is not available, the Listener will automatically buffer the message for delivery at a later time. Thus the Listener can be used for guaranteed message delivery.
Listeners are completely autonomous processes that can be administered remotely. This remote administration allows one to configure and debug new operations, change the scheduling of operations, inspect and manage the message buffer and check on the status of the Listener. Communications between Listeners are completely secure.

SOAP/Web Services form an increasingly important connectivity method. Therefore it is not surprising that standards are emerging that define web service operations: WITSML, ProdML, SP95, etc. Each of these defines a standard WSDL to be supported by the particular application. Listeners, as well as supporting user-defined operations, can emulate one or more WSDL standards simultaneously. This means the same application system and Listener can support both a ProdML and SP95 interface simultaneously: no need to bet on one particular standard exclusively.

One application that we will see increasingly in an SOA will be workflow automation: BPEL. If we take the example of a monitoring application that detects some anomalous condition, it can now use the Listener to detect this condition which, when it occurs, gets notified to the BPEL engine where we can initiate a new workflow process for handling this type of event (‘Create inspection work order’ for example). When handling this workflow process, the BPEL engine might need to access other applications, and it can do this again via the SOA using Listeners to talk to, say, the Maintenance Management System Listener in order to create an inspection work order.

**Information: The Intelligent Plant Data Model**

With the power of the Listener we have already solved a significant proportion of the problem. However, when we started we said there will be 50-150 applications in a single site, each of which might have a Listener. So how does an application program know which Listener to talk to in order to get the answer to a question? For example, I might want to know the discharge pressure of pump G-101; there may be many applications to which there is a Listener that could answer that question, but I do not want to broadcast the request to every possible system. Not only that, I do not even know terminology of that system: I am calling it ‘G-101’, but I cannot be certain the system that knows G-101’s discharge pressure calls it by the same name. Just as we do when we get lost and refer to Google® Maps, we need a roadmap to all of the information within the SOA domain.

Our solution to this is to create an intelligent plant data model. We tend to think of a database as a dumping-ground for data. Our intelligent plant data model is more like a model of the plant or enterprise. Just like Google Maps is a physical model of the roads and interconnectivity, our data model is a map of all of the assets or materiel of the business (pumps, people, plans, schedules, instruments, products, raw materials, grades, recipes, movements, etc). All of these objects are registered into the ‘Registry’ at the core of this plant data model. How do they get registered? One way for that to happen is the Listeners detect changes in their respective systems and automatically notify the plant model of these changes.

![Image of Business Process and Physical Plant](image-url)
Data vs. Knowledge

Paradoxically, even with the vast number of applications and associated databases, there is still a substantial amount of information that is not in any database. Reflect upon a unit's daily operating report. Yes, the raw data comes from the plant information systems. However, where is the knowledge that I should use 20:FI-107 as the real-time database tag for the main feed to the unit? If a new employee arrives, can they find that knowledge in a database or does he have to consult the plant's resident Einstein, the go-to-guy who has all of these answers in his head?

The Registry contains the identity of the objects, plus a few details such as which systems recognize the existence of the object and by what name they refer to that object. Thus alone, the Registry provides what is called 'Master Data Management' allowing one to identify the location and alias of objects in other systems. This allows the plant data model to determine to which system it should send the request for, say, information about a particular piece of equipment.

Now let us make the demands on the system greater: I want to know the average discharge pressure of all pumps in a particular area of my organization. I have Listeners that allow me to communicate to all of the external systems, so I know I can retrieve the information, but to which systems should I be communicating? This is where we have extended the simple Registry and object metamodel to a physical plant data model that understands not only the assets/material, but also the topology/connectivity/organization of the plant as well as the temporal characteristics—that is, how things are changing over time; not just measurement values, but what material is where and when.

This intelligent plant data model allows me to capture intelligence about the plant that is often only contained in the head of the resident 'Einstein', who might well be retiring soon, taking this knowledge with them. So now I can ask the Intelligent Plant Data Model to give me all of the discharge pressures of pumps within a certain area. The data model determines first of all, using its topology, which pumps we are talking about.

It then uses the registry to determine to which systems it must communicate to retrieve the particular information: it might well be multiple systems even though in this example we are asking for homogenous information. The plant data model then issues the SOAP service requests to the respective systems which respond with the information. The plant data model then assembles the results and returns them to the caller as if the data was within the plant data model all along. In fact, the plant data model can provide complete location independence for all of the information. Using a Listener on the Plant Data Model, it becomes the single source of WITSML, ProdML, SP95, etc. SOAP services, handling the distribution of the requests to the underlying system. Of course, the presence of the Plant Data Model does not prevent someone writing applications that go directly to the Listener, but the question would be "why?"

Data Model Standards

We are now seeing standards appearing that provide an outline data model for various aspects of the business, such as SP95 for plant operations and MIMOSA CCOM for asset management information.

Data Damage

Information originates from the 'silos' supporting the various business functions. These have evolved over the years using their own terminology, naming conventions, structures, etc. Human beings have a great capacity to align this information, quickly recognizing that FI-107 is really the same as FI:107. Computers are not nearly as smart. This gives rise to an apparent 'data-damage' problem when information has to be merged across systems. Most of this data-cleansing has traditionally taken place in the interface, a solution which creates its own management problems. Recently Master Data Management (MDM) systems have emerged to centralize this management problem. However, MDMs only provide the mappings between data; what is really required is a complete contextualization of the information from different sources. This can be achieved by the adoption of a single ontology for the entire business.
Why not adopt one of these models as the data model? The reason why we should not is that each of these standards only covers a particular aspect of the overall business. Therefore we would be guilty of creating more icebergs, albeit standardized icebergs, of information.

So does this mean that the Intelligent Plant Data Model is proprietary, locking out these standards? The answer is no. The approach adopted for our data-model is that of a physical model, capturing the physical business to any degree of detail required; after all, the physical assets and materiel are the real core of any process business. Given a physical data model we can always aggregate the information over time, over plant areas, over categories of assets to meet any business requirement. Unfortunately we cannot disaggregate information stored in a business-process-centric data model to recreate the detailed information. By adoption of this model, we can emulate any of the data model standards. Thus the Intelligent Plant Data Model can be configured to provide the equivalent of all of the SP95 information, or the MIMOSA CCOM simultaneously. The reason that we can do this is that both are referring to the same common physical data model, without the need to duplicate the information.

**Warehousing Information**

So far we have assumed that we can retrieve the information that we need on demand using the services of the Listener. However, this will not always be the case. Sometimes the system that contains the information can not be trusted to deliver that information *ad infinitum*: a spreadsheet containing the plant schedule is a classic example of here-today, gone-tomorrow information. We now turn to the Intelligent Plant Data model again to act as a data warehouse for such information. The Listeners perform the Extract-Transform functions and transmit to the Intelligent Plant Data Model’s Listener, which performs the Load into the data model. From the point of view of the information user it makes no difference; they still request the information in the same way, except that the data model no longer has to retrieve the information on demand.

**Virtual Calculations**

Why do we see the proliferation of so many spreadsheets? One reason is that they allow the user to create calculated information. Unfortunately, despite the fact that these calculated results may be key to the overall business performance, there is little or no control over the accuracy and consistency of these calculations. Would they stand up to a Sarbanes-Oxley audit? Again the Intelligent Plant Data Model can solve the problem. It can contain virtual calculation criteria. These are attributes of objects whose values are determined by a calculation. However the results of these calculations are not normally stored in any database but are calculated on demand. This has great advantages especially in the Production Systems Gap, as the information does not always arrive in the correct chronological order. For example the laboratory analysis of a material may not be available for 4~6 hours after the sample is taken. These calculations have access to all of the other information in the data model, including other virtual calculations. Like all other information in the data model, changes to these calculations can be audited and access to the code restricted.

**Operational Data**

Another information management paradox in the Production Systems Gap is the fact that, although we are overwhelmed with information, there is some information that has no home at all. For example: the disposition of each storage location might or might not be known ‘electronically’; the allowable material lineups might be known to only a few operators.

This can be solved by purchasing and deploying an application to handle this information. An alternative is to use the same plant data model to store this information. This does require a transactional interface to the database, and this is provided via definable data transformation rules (‘macros’), reporting queries, and transactions. Furthermore, a web portal such as Operational Insight provides a transactional interface to this information, allowing users to not only view the information but to act upon it by entering, updating or deleting information thereby influencing the actions on the plant such as blend recipes, berth schedules, etc.
**Presentation: Operational Insight plant portal and applications**

So far we have not indicated how we may access this information. We can write our own code to talk to the Listeners. Since the Intelligent Plant Data Model is a pure SQL database, one can use any standard SQL access tool: Excel®, Crystal Reports, Business Objects, Cognos, etc. A richer user interface is provided using Operational Insight that can access any of the information within the Intelligent Plant Data Model: graphical displays, trends, dashboards etc. Furthermore, to close the business loop we need to act on the information that we see. This means we also need the user interface to support data-entry transactions allowing us to alter, enter, and delete information. Thus continuing our example, we might receive notification of which pumps have suspect discharge pressures: we can select particular pumps and issue inspection work orders.

Operational Insight is Powered by Matrikon, which represents vendor neutrality. This product works with third-party control systems and applications.

We can take this further; it may come as a surprise that, despite the estimated 50-150 applications per site, there are still important applications that are missing, manual, or deployed in obsolete systems. We could of course purchase an application, deploy it as a standalone application and connect it via a Listener to the other systems. Sometimes, however, we do not want all of an application’s functionality. Thus we have the option of deploying that application within the Intelligent Plant Data Model. An example might be Production Accounting: we have all of the information regarding inventories, material movements, flow rates etc. We also have the topological model. Therefore it can be simpler to use the Intelligent Plant Data Model as the location of the Production Accounting application rather than deploying something standalone.

**Figure 9: Integration of systems**

**Superfluous Functionality**

Applications are frequently deployed with superfluous functionality. This is attractive to the users, who then start using this functionality instead of the enterprise standard. Frequent examples are event handling and notification, document management features, reporting sub-systems, duplicate connectivity to other systems and uncontrolled replication of master data.
Putting it all together

In summary, our technology contributions to lubricating the business control loop are:

**Integration**
The highly adaptable Listeners convert existing applications to full participants in a Service Orientated Architecture (SOA) by establishing a web service façade.

**Information**
The object catalog or Registry provides an index to all objects of interest throughout the SOA domain.

The Data Directory provides a service (MDM) for applications to locate these objects within the Listener-connected systems.

The Object Meta-Model allows the characteristics of objects to be continuously adapted to the evolving business.

Objects' attributes have access to the external information via Listeners.

Objects' attribute information can be extracted, transformed and loaded by the same Listeners and warehoused in the Plant Data Model for secure long-term retention.

Objects' attributes may also be virtual calculations, which are only executed on demand.

The data model can also act as an operational data store, fully supporting transactions so the same database is the primary storage location for the information.

Materiel, temporal, and spatial relationships between objects are configured and captured in the relational meta-model.

Data rules, report queries and transactions can be declared and configured in the database allowing the Intelligent Plant Data Model to capture any business-data rules, including standards such as SP95, and MIMISA CCOM.

**Orchestration**
Listeners present a BPEL4WS interface so they can be orchestrated by BPEL-compliant workflow engines.

**Presentation**
The Operational Insight web portal provides users and easily configured graphical, transactional and reporting user interface.

The open interface to both the Listeners and the Intelligent Plant Data Model, allows other data presentation, reporting tools, portals, etc. to access the same information.

**Application**
The system can be a framework for applications as an alternative to external, applications. Applications groups have the following solutions:

**Production Management:** material (inventory, unit, movement), upstream, batch tracking, operating instructions, shipping, laboratory information, recipe, assay, specifications, environmental;

**Asset Management:** specifications, health monitoring, meter management;
Operations Management: document indexing, logbook, instructions;

Performance Management: Solomon indices, KPI, balancing and reconciliation

How can this technology be used to improve the business? Each enabling technology is not a solution alone.

1. Integration of systems does not solve the availability of integrated information.
2. Portals do not imply intelligence.
3. Portals alone do not orchestrate synchronized businesses.
4. Workflow automation does not identify the business processes that need to be executed; users do via access to the state of the business through a portal.
5. The best applications have no value if the decisions cannot be propagated throughout the business and the information shared.

Overall architecture requires using all elements of the enabling technologies.

1. 1. Poor implementation strategy puts reliance on single technology enabler.
2. 2. Technology enablers are purchased as independent projects, not part of a coherent plan.

Closing the loop: Applications, integration, presentation, and information as a means of supporting the business

We started this tour with the business process loop. We can now summarize how we close that loop:

![Diagram showing the relationship between business goals, assets, processes, targets, and actions.]

Applications, possibly located in the Intelligent Plant Data Model, support the business process as it tracks the targets and determines the appropriate actions on the business assets.

Information, in the form of the Intelligent Plant Data Model, captures the planned, current and past state of these business assets and allows the raw information to be transformed to meaningful and accurate measurements.
**Presentation**, via Operational Insight web portal, provides timely access to these measures to ensure the business goals are being met. Furthermore, the targets can be quickly changed in response to disturbances or operational changes.

**Integration**, using Listeners as web service facades on existing applications, provides the glues to link applications, information, and presentation into an integrated whole.

**Path Forward**

1. **Integration**: If a streamlined business requires that information and decisions freely flow between applications, there needs to be a common mechanism for integrating the applications so they can directly or indirectly talk to each other.

2. **Presentation**: Visibility of real-time information simultaneously from all business areas ensures decisions can be made based on the most realistic information, instead of a conservative or out-of-date estimate of the situation.

3. **Orchestration**: Timely reaction to changing business circumstances increases margins, because the operation will be running closer to optimum. Workflow automation can automate many of the business processes, reducing training time and ensuring most situations are handled in a timely and accurate manner. After all, well-tuned business processes are the primary differentiator distinguishing profitable from unsuccessful business.

4. **Information**: A single version of the truth eliminates inconsistent information and lack of reconciliation of information between siloed systems. This means greater confidence in the information and less time spent trying to locate the truth.

5. **Application**: Applications should meet the requirements of the business processes. However, in an attempt to make their products more attractive, more and more features are incorporated into a product. If these features are used, they can lead to a fragmentation of the business processes and silos of information. For example, an application such as a Maintenance Management System may have

6. **Implementation**: A deployment strategy is required that combines the risk management necessary with breakthrough technology, along with the process change management procedures to prepare the business to adopt the business processes, as well as conventional project management.

---

For more information:

For more information about Production Intelligence, visit our website www.honeywell.com/ps or contact your Honeywell account manager.

www.matrikon.com

mpi@matrikon.com

**Honeywell Process Solutions**

1250 West Sam Houston Parkway South

Houston, TX 77042

Lovelace Road, Southern Industrial Estate

Bracknell, Berkshire, England RG12 8WD

Shanghai City Centre, 100 Junyi Road

Shanghai, China 20051

www.honeywell.com/ps