Interface Engines for Healthcare: 21st Century Trends

Updated March 2006

Setting the Scene

The Healthcare Information Technology (HIT) industry has never been more dynamic or demanding then it is today. HIT is increasingly being looked at as a critical component within Health Information Management (HIM) and the goals of improving patient care, reducing medical errors, reducing costs, and increasing patient information access and security. In order to achieve these very important but difficult goals, effective and accurate integration of the component systems and data repositories is crucial.

There are many vital initiatives and capabilities that are only possible through the efficient integration of information from many different components in the HIM industry including:
Electronic Medical/Health Records (EMR/EHR)
Personal Health Records (PHR)
National Health Information Network (NHIN)
Regional Health Information Organizations (RHIO's)
Disease Management
Computerized Physician Order Entry (CPOE)
E-prescribing
Streamlined Claims review and processing

The term created for the ability of different systems to seamlessly and automatically share and utilize information without customization is "Interoperability". If true Interoperability is achieved the task of sharing information between disparate systems will become virtually "plug and play". However, due to the incredible diversity and complexity of healthcare related information and participants, this ability is realistically many years away at best and may never be fully achieved.

The solution to this integration challenge is the interface engine.

In the HIT industry, interface engines have played a vital role in many organizations, usually hospitals, for many years. Implementing and maintaining an interface engine for healthcare has proven to be a highly complex task, given the wide variety of data types and range of formats utilized within the industry.

In an ideal HIT infrastructure, interface engines operate smoothly behind the scenes to coordinate data from multiple systems, performing hidden but critical functions vital to the success of the organization. In fact, the only time when most healthcare organization's users will notice their organization's interface engine is when it fails, and the transfer of clinical data between the organization's systems is halted. In this way the interface engine is similar to the transmission in a car. Few people get excited about what kind of transmission their car has and while it operates smoothly no one pays much attention, but if it fails, the entire car is useless.

Today, an interface engine is considered a "mission critical" component in most healthcare organization's IT infrastructure. In terms of effective communication in a healthcare organization, an interface engine is just as important as the email server or telephone system.

Making a wise choice of interface engine and 3rd party resources can bring great efficiencies to a healthcare organization. Likewise, a poor purchasing decision, or inadequate installation, design, or support can lead to chaos.

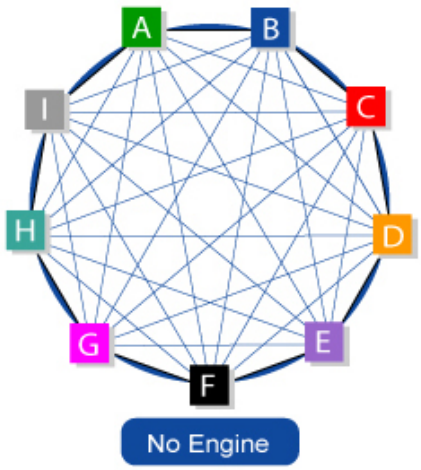
We have found, over the years, the most important factor in developing an efficient, effective, and stable interface scheme and structure is in the experience the interface engine builder/programmer possesses in the area of interface theory. In other words, what is the ***best*** way to build an interface? Knowing how best to create an interface contributes roughly 70% towards the success of any interface project, while knowing the tool involved, in this case the engine, contributes roughly 30%. However, if the engine being used is too complicated to use or contains inherent inadequacies, it doesn't matter how well versed in the area of interface theory the person using it is, the ultimate success of the interface project will be seriously impaired.

This white paper sets out to isolate and comment on a number of trends in the current marketplace for interface engines in healthcare. We have included some advice for potential buyers in light of these trends. We have also included a grid comparing the major interface engines, addressing most of the important features (see appendix A).

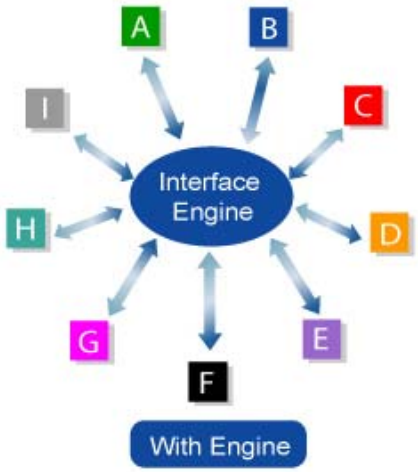
What is an Interface Engine?
Put simply, an interface engine is a healthcare organization's "communications exchange" for clinical and business data, ensuring that information passes smoothly and quickly from one healthcare organization computer system to another.

An interface engine is a software program designed to simplify the creation and management of interfaces between separate applications and systems, either within the organization or with other affiliated organizations. Interface engines undertake messaging between systems, and normally manage any mapping, translation and data modification necessary to ensure the effective exchange of data around the organization.

Another way to think of an interface engine is the centralized, automated "hub" of a messaging network. Rather than connecting all systems to each other individually, (a highly complex, time-consuming and unreliable process), an interface engine acts as the intermediary for all messaging between healthcare organization systems, as illustrated in the following diagrams:



1. With many difference applications running within a hospital, establishing maintaining separate interfaces between them can be complex, technically difficult



2. An interface engine eliminates some of this complexity, acting as a single point of management for all interfaces around a hospital

A fully functioning interface engine will provide more than simply messaging and interface management; it should also provide guaranteed delivery of messages, message queue management, message verification tools, and visible status capabilities.

These visible status capabilities should include the vital function of immediately generating effective alerts for the IT staff in the event of a problem. *The IT support staff should always be the first to be aware of an interface problem, not the last.*

Healthcare Facilities Looking to Upgrade/Replace

The current market for Interface Engines is in full evolution for health care providers, vendors, and consultants alike.

Most major healthcare organizations have been operating some kind of data engine/ interface engine for several years. Many are finding that these older legacy data engines are being “sun-set” by their vendors, and will no longer be maintained under current support agreements.

The requirements of the new HIPAA standards for healthcare communications are also prompting many organizations to look at upgrades and improvements to their Interface Engines. Until HIPAA, most healthcare organizations had no need to support the X12 messaging standards that HIPAA is based upon. Many healthcare organizations are now finding that an upgrade of their engine to comply with HIPAA will be enormously expensive.

The current Interface Engine market can therefore be described as a “replacement market”, with many engines in place now being found inadequate, outdated or simply no longer supported by vendors. This makes now the perfect time for a healthcare facility to evaluate the engine product offerings, to find an engine that not only meets today’s criteria, but future criteria as well.

Unlike most product areas within health IT, the number of vendors which healthcare organizations have to choose from for a good interface engine is very small, with fewer than ten major vendors from which to select.

Some Vendors Changing Focus

Most vendors have also widened the focus of their products and are marketing their interface engines in other industries such as finance, insurance and retail. The interface engines themselves have been expanded to become generic "integration engines" for business, with wider functionality. Certain vendors seem to be finding the wider e-business market an easier place to sell their wares, as healthcare installations are typically complex, time-consuming and thus less profitable. In other cases, vendors are looking to sell their clinical, financial or management software and the interface engines are an afterthought.

This may not be a good thing for clients within healthcare, since the focus, (and therefore the development and support efforts), of many of these vendors is now devoted elsewhere.

The interfacing needs of healthcare are very specific, and are sometimes not properly addressed by a vendor or a product that is too "generic". We tend to recommend interface engine vendors that remain solely or primarily focused on healthcare.

Many major interface engine vendors have also re-branded and undergone consolidation over the past couple of years. Perhaps the most well-known examples of evolution among interface engine vendors are the merger of Healthcare.com and Xcare.net into Quovadx (with the engine formerly known as Cloverleaf), the consolidation New Era of Networks (NEON) and CAI into Sybase, and SeeBeyond and their E*Gate (formerly Datagate) engine being purchased by Sun Microsystems.

Among the newer players in the Interface Engine market are Microsoft's BizTalk product, Orion Systems Rhapsody product, and iNTERFACEWARE's Iguana product.

The changing role of Consulting resources

In the past 3rd party interface engine consulting resources have been primarily focused on knowing the specific interface engine tools, which was required because the tools were so complicated.

Many consulting resources are still operating in this fashion, often being used for servicing and supporting older versions of engines that are no longer supported by vendors.

However, currently and in the future the most valuable 3rd party consulting resources will not be those that know a single engine, but instead will be those that have many years of experience in creating all kinds of interfaces for many different types of healthcare focused organizations, working with multiple engines and dozens of vendor systems. The more proper term for this type of resource is "3rd party expertise resource". Installing an easy to use interface engine can allow the facility to also realize savings in 3rd party costs, since the 3rd party expertise resources may simply act as a

supplement to an in-house staff on an as-needed basis, instead of being used for specific blocks of time on a long term basis.

Improving user friendliness
An ideal engine is one that the IT staff can install, configure, turn on and then forget about unless something goes wrong.

A few years ago, installing and maintaining an interface engine required extensive programming expertise and often required IT staff to learn proprietary code to carry out routine maintenance and editing. Also increasing the amount of time needed to learn and build was that most engines did not employ time saving control devices such as drag-and-drop, point-and-click, cut-and-paste, etc. Some engines still do not use these commonly used devices.

In addition, some of the engines had to be installed and configured entirely by the vendor because they were so complex.

Today, many interface engines make extensive use of graphical tools and clear user interfaces, which has improved the usability and clarity. While we would not claim that any of the Engines currently available can be operated by a completely non-IT literate operator, most Engines are improving their user-friendliness.

In addition, common, non-proprietary languages such as C++ and JavaScript are gradually replacing proprietary coding languages. However, some engines still require specialist knowledge of languages such as MUNK or TCL. IT staff with skills in these “rare” languages can be difficult to recruit and retain.

Good examples of this trend are seen in the new GUIs used in Siemen’s OPENLink product, and the user interfaces for Orion’s Rhapsody. Rhapsody provides a good EDI designer feature, which aids the creation of message definitions. Quovadx’s Cloverleaf and Sun Microsystems E*Gate also have a strong reputation for clear graphical tools.

We believe “easy-to-use” features are an important factor in IT staff recruitment and retention. Most IT staff don’t want to spend whole days mired in administration tasks for an interface engine. The easier the engine is to operate, the more likely the staff will be confident about that engine’s ability to perform. There are also significant cost savings in terms of staff time when using an easy-to-use engine.

HIPAA components and other protocols (HL7, etc)
A good engine will support all of the major messaging and protocol standards used in healthcare (HIPAA, HL7, XML, NCPDP, HCFA, ASTM, UB92 etc.)

All of the Interface Engines now available for healthcare have some kind of capacity for mapping/translation to the new HIPAA formats. The way this “HIPAA-readiness” is provided, however, varies among the products.

Potential healthcare buyers should look for engines with HIPAA message definitions preloaded as part of the basic product. Extensive HIPAA components are included in the OPENLink, Quovadx, Rhapsody and E*Gate engines.

Microsoft's generic Biztalk product requires the additional purchase of a "HIPAA accelerator" which handles HIPAA transactions.

Interface Engine buyers should also look for native support for HL7 and XML, both of which look to be important healthcare messaging protocols in the future. With HL7 v.3.0 moving closer to wide industry acceptance in the near future, vendors should indicate what their contingency is for the support and upgrade to this new version.

HIPAA Certification

Engine vendors should provide evidence of the certification of their HIPAA messaging components against the latest available standards. Most vendors provide this either through an agreement with Claredi Corporation www.claredi.com, or through membership and participation in the HIPAA Conformance Certification Organization, HCCO www.hcco.us.

Claredi Corporation is a commercial enterprise, providing HIPAA EDI standard transaction testing and certification solutions, and is endorsed by the American Hospital Association.

HCCO is a cooperative, non-profit organization of participants from throughout the healthcare sector that develops guidelines and provides accreditation services through their Common Compliance Assessment Program (CCAP).

Interface engine buyers who need to demonstrate conformance for the purposes of the HIPAA Transactions and Code Sets Rule should ask their vendor to provide either HCCO CCAP or Claredi certification for the components of the engine involved in processing HIPAA messages.

Self-customization

Another important factor in multi-protocol support is the provision of tools for customization of message protocols and message definitions. No single vendor seems to follow the standard messaging specifications exactly. Someone (i.e. the healthcare organization IT staff or 3rd party resource) will have to customize their interfaces to support the vendor's requirements.

This customization is relatively easy to carry out on certain engines, while on others it can prove a programming nightmare. Some engines still rely on hand-coded text files for message definitions, but others offer handy graphical tools for composing messages and mapping.

Microsoft's BizTalk product provides a robust set of message moving services. However these toolsets may be too cumbersome for many IT departments to use. A more traditional interface engine application, with its integrated setup, administration and monitoring may be more suitable for providers who may not have the necessary resources to develop BizTalk services and functions.

Ensuring an engine is "future proof" is vital. No healthcare organization wants to replace their engine every few years. It is critical the engine is flexible enough to allow the addition and extension of functionality, avoiding the cost and hassle of an entire engine replacement when IT requirements change. It is also equally important to have an engine that is supported by a solid company that is committed and focused on the specific needs of healthcare facilities.

Customer Support
Support and staff training is a key part of any purchase agreement. Some vendors have better reputations for support and service than others, as do the various 3rd party resources. The KLAS ratings may be a good indicator of a company's quality of service. Talking to other healthcare organizations who already operate an certain engine is a good way of investigating the quality of a vendor's support framework.

User Resources
Having access to other sources of information and expertise is also an important component in the overall success of an interface engine within the healthcare facility. These resources can help fill in the experience gaps of the staff using the engine.

One effective resource is a dedicated email listserve, available to the user community, one in which the users can ask questions, offer suggestions, and share tips and information. Ideally, the email listserve is limited to only those using the engine, so those with other purposes, such as recruiters and competing vendors, can not misappropriate information, which can seriously impede the effective flow of information. For this same reason, the ideal listserve will have little or no involvement from the manufacturer of the engine, so it truly remains a "*users*" listserve.

Another effective and low cost method of accessing a high quality resource is to create a relationship with a third party that has extensive experience and knowledge about interface theory, as well as the particular engine being used. This third party resource would act in an "on call" capacity, being able to answer a short phone call or email quickly and correctly, as well as help out when needed.

The Hidden Costs of Software Upgrades
Some healthcare organizations have found there is a "sting in the tail" of an Engine purchase when it comes to upgrades and add-ons. Make sure you understand the costs involved in upgrading to new versions of an Engine ***before*** buying!

Among healthcare clients we have spoken to, Microsoft's BizTalk product provides a good example of "hidden costs". While the initial price tag of BizTalk is very attractive, the "add-ons" required (SQL server, high-spec hardware, additional HL7/HIPAA accelerators etc) soon push the price up.

Staff Training
Staff training on a new Interface Engine normally can take one to two days for simple monitor, support, and troubleshooting functions, and up to several weeks for more in depth interface design and build skills. Many vendors offer an extensive range of training 'certifications' on their products for integration.

While these certifications serve an important purpose, they may not in themselves be a good indicator of an individuals preparedness to tackle the often complex integration challenges common in today's healthcare IT industry.

However, because there are so many different integration methods and technologies in use, as well as so many different standards and interpretations of those standards, no amount of training can replace the value of actual experience, building interface after interface and addressing the challenges that each one may bring.

Open Architecture and Platform Independence

It is important to choose an Interface Engine that can operate on a platform and architecture that fit in well with those already being supported by the healthcare organization. Today's Interface Engines can run on a variety of platforms, and often use open standards to ensure maximum interoperability with an organization's existing systems.

For example, Microsoft BizTalk and Siemens OPENLink both run only on Microsoft NT/2000 based platforms or high availability configurations. However, these platforms are usually already being supported by most healthcare organizations and therefore fit in well with an organization's current skill sets and support procedures.

By contrast, E*Gate offers LINUX support, while Orion's Rhapsody can run on any platform that can run a Java Virtual Machine.

Sybase has a good reputation running on varied high-availability configurations.

Architecture – "Hub and Spoke" vs. "Network Architecture"

Traditionally, integration engines have been built along the "hub and spoke" model, where a centralized engine becomes the single point where messages from all sources are processed before being re-transmitted. An advantage to the "hub and spoke" model is that it makes it simpler to monitor and support, however is also can leave the user with a single point of failure (a rare but possible occurrence).

An alternative to the "hub and spoke" model is a "distributed network architecture" approach where the message processing is distributed among multiple interface engines on the network. If created properly this architecture may offer better failover support than the traditional "hub and spoke" model but can also lead to increased difficulty in monitoring and support.

However, for most healthcare organizations, an integration engine using the "hub and spoke" approach is still highly appropriate and allows the full value of the inherent synergy in having a "single input to multiple output" interface scheme on a single engine. Network architecture is something to be considered but it often requires a lot of consultancy and design work up-front, prior to implementation. It is not yet clear whether network architecture offers any compelling advantages over "hub and spoke", either in price, flexibility or future expandability.

Web Services Support
Web services provide a standard way to discretely package anything (a database, a specific query, some business logic) and make it accessible to anything else (another database, a WAP-enabled phone, or even an external partner's business logic). The most common protocols used to achieve this integration are XML, SOAP, WSDL, and UDDI, with XML and SOAP being by far the most common.

Most current versions of the interface engines reviewed here offer some kind of support for web services, but if you are considering using your interface engine to deliver web services, it is worth asking your vendor in detail exactly what they support. For example, some engines only provide SOAP client communication points, which are able to query an external SOAP server, but these engines may not offer full processing of web service routes.

If you know at the time of purchase what you want to achieve with web services, discuss this with your vendor to ensure that the engine you are purchasing will support your requirements.

Fax/Email/PDF support
Interoperability with Fax and Email is becoming a niche area where some vendors are making important progress. These functions not only improve the monitoring and support of the interface engine itself, but will certainly become important for improved communication between patient and provider.

Sybase offers an Email input option, which can convert standard email into a formatted message.

Orion's Rhapsody can convert formatted messages into PDF and RTF format documents, which are ideal for automatic routing to computer fax, printer and email.

Other interface engines can have customized solutions created to perform many of these same functions.

Conclusions
Given the current developments in the Interface Engine market, we believe potential healthcare customers should consider the following important questions before making a purchase:

1. How easy is the engine to install? Does the vendor offer an easy migration path from your existing engine product?

2. How easy is the engine to configure and use? Does it allow customization and extensions to be carried out by the customer or 3rd party resource themselves, without vendor involvement?

3. How reliable is the engine? Does the engine have a reputation for "lights out" operation (24x7) with minimal interference by IT staff?

4. Does the vendor have a primary focus on healthcare? Will my future support and service arrangements be jeopardized by a vendor who isn't fully committed to its healthcare customers?

5. Does the interface engine builder/programmer have a comprehensive understanding of interface theory?

6. What are the hidden "add-on" and ongoing costs of purchasing the engine?

A good interface engine gives the healthcare organization the ability to get data from wherever it is, do whatever data manipulation is required and deliver that data to whatever destination, repository, or device they wish, and do all of this invisibly to the customers of the IT department. In the ideal situation, the healthcare organization's IT staff should be able to do all of the monitoring and the bulk of maintenance and support themselves.

The ultimate goal when choosing an interface engine? Having a solid, reliable engine that facilitates a freedom to innovate, being enabled, not constrained by the tools you've got installed.

Appendix A – Major Interface Engine Comparison Grid

	Microsoft BizTalk[1]	Sybase e-Biz Impact[2]	Sun e*Gate[3]	iNTERFA-CEWARE IGUANA[4]	Siemens OpenLink[5]	Orion Rhapsody[6]	Quovadx Cloverleaf[7]
HL7 Native Support(v.2.1-3.0)	X	X	X	X	X	X	X
X.12 Native Support	X	X	X		X	X	X
HIPAA v.4010a Native Support	X	X	X		X	X	X
Native XML Support	X	X	X		X	X	X
Native ODBC Support	3RD PARTY	X	X	X	X	X	X
Native JDBC Support			X			X	X BPMS add-on
NCPDP Native Support			X			X	X
Native EDIFACT Support	X		X			X	X
DICOM Support (Image + data extraction)			X		X	X	X Etiam add-on
Non-proprietary programming language		C++/TCL	Java, XSLT, BPEL Min. Prog. req.	Python, C++, JAva, VB6, Delphi, .NET	No language required. User exits done in Perl, C++, etc.	Javascript/Java	X
"Information Service Center" Model*			Implements Svc-oriented architecture		X		X Intelligent Broker add-on
TCP/IP Native Support	X	X	X	X	X	X	X
SSL for TCP/IP	X		X	X	X	X	X
Flat File	X	X	X	X	X	X	X
Java RMI			X		X		
JMS input/output	3RD PARTY		X			X	X BPMS add-on
E-mail output (POP3, SMTP)	X	X	X	X	X	X	X Intelligent Broker add-on
HylaFAX output			X custom			X	
Printer Output			X		X	X	X Screen Rejuvenator add-on
SNA Architecture	X	X	X	X	X	X	X
Serial RS-232		X	X		X	X	
MQ Series, MSMQ	X	X	X	X	X	X	X
Screen Scraping		X	X		X		X Screen Rejuvenator add-on
Printer Port Capture		X	X custom		X		X Screen Rejuvenator add-on
Sophisticated Rules Engine	X	X	X	X			X BPMS add-on

	Microsoft BizTalk[1]	Sybase e-Biz Impact[2]	Sun e*Gate[3]	iNTERFA-CEWARE IGUANA[4]	Siemens OpenLink[5]	Orion Rhapsody[6]	Quovadx Cloverleaf[7]
Highly Intuitive GUI	X	X	X	X	X	X	X
24/7 Customer Support	X	X	X	X	X	X	X
FTP Client	X	X	X	X	X	X	X
HTTP (Client and Server)	X		X		X	X	X
Multimedia files sent as BLOBS			X		X	X	X
Character Set Conversion (eg. EBCDIC-ASCII)	3RD PARTY		X	X	X	X	X
Maintenance/ editing engine without stopping engine	X		X	X	X	X	X
Evaluation copy available before purchasing	X		X as part of proof of concept	X		X	X
Web Demo	X			X	X	X	X
Graphical Charting of Engine Stats	X		X custom	X	X	X	X
Graphical Monitoring of Route/Comm Point status	X		X	X	X	X	X
XSLT Stylesheet transforms for webpage viewing of messages	X		X	X	X	X	X Intelligent Broker add-on
PDF/RTF transforms for fax/print/email transmission	3RD PARTY	Email add-on available	X custom		X	X	
Message Encryption Filters Native	X		X	X		X	X
Email Input	X	X	X		X	X	
Runs on cluster server for failover support	X		X	X	X	X	X
Published API for extending product	X	X	X	X	X	X	X
Native support for ERP applications	X	X	X				X

	Microsoft BizTalk[1]	Sybase e-Biz Impact[2]	Sun e*Gate[3]	iNTERFA-CEWARE IGUANA[4]	Siemens OpenLink[5]	Orion Rhapsody[6]	Quovadx Cloverleaf[7]
HIPAA Message Validation	X		X	X	X	X	X Via EDIFECS partner's add-on
Encryption	X		X	X	X	X	X
Digital Certificates, Authentication	X		X	X	X	X	X
LDAP Support			X				X
SOAP Support	X		X	X	X	X	X Via Intelligent Heatlh Broker add-on
Win NT/2000	X	X	X	X	X	X	X
Linux			X	X		X	X
Windows XP	X		X	X		X	X
High Availability Configurations	X		X	X	X	X	X
Claredi Agreement			X				X
HCCO Membership		X	X		X	X	X

1. Updated information provided by Microsoft 3/16/2006
2. No information provided by Sybase
3. Updated information provided by Sun Microsystems 3/13/2006
4. Updated information provided by iNTERFACEWARE 3/14/2006
5. Updated information provided by Siemens 3/28/2006
6. Updated information provided by Orion Systems 2/27/2006
7. Updated information provided by Quovadx 3/17/2006