IQware's Rule-Based Technology: A Brief Tutorial

Version 1.3
March 10, 2008

(Protected by US Patent #7,322,028, issued 1/22/08)

Table of Contents

1.0 IQware Solutions

The Problem

Software costs too much, takes too long and doesn't do what you want. Delivering software that is on time, on budget and that meets specifications is a process that is expensive, time-consuming, risky, error-prone and nearly endless.

The traditional software development method is the endless "code-compile-test-debug" loop. This method produces inflexible applications and requires legions of software engineers or programmers. Good programmers are hard to find - good software engineers are harder to find - and both are very expensive after you've found them.

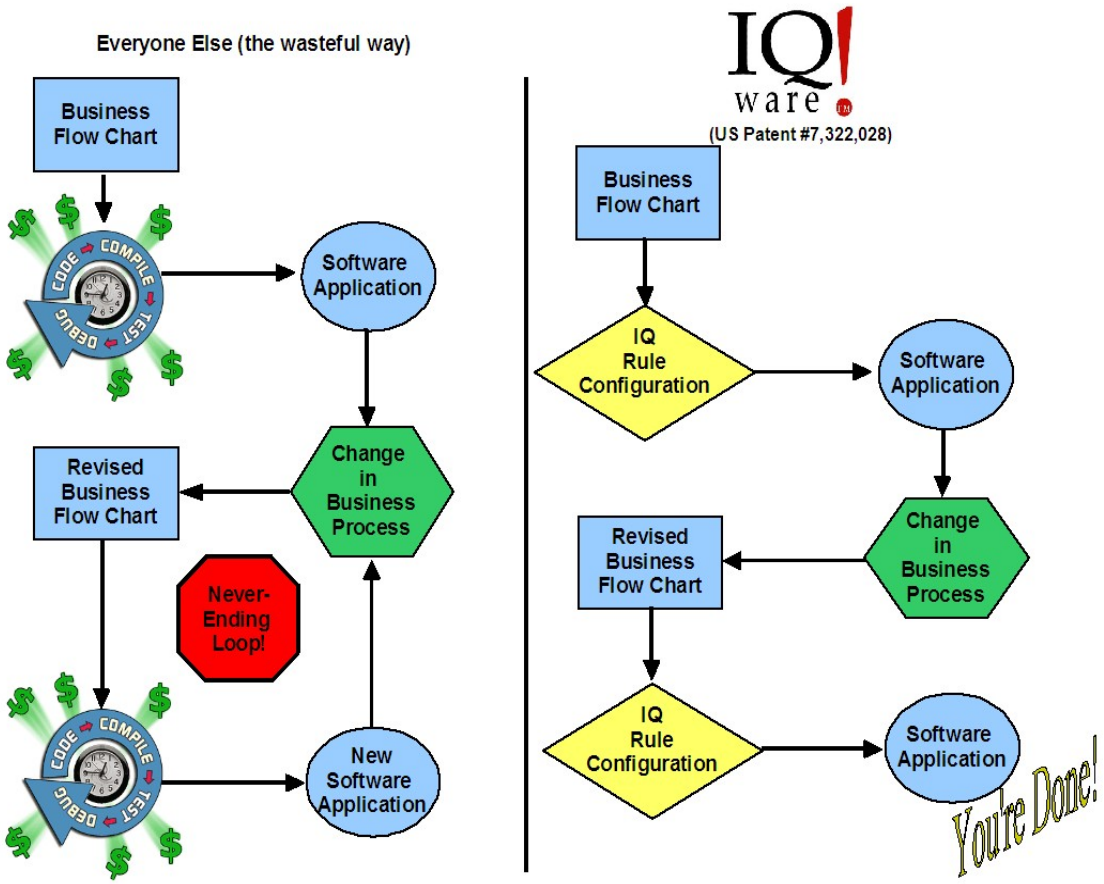
Software development tools are ubiquitous - your competitor probably uses what you use. Outsourcing reduces hourly rates, but how do you compete economically when everyone is doing that too? And when it's all said and done, is the completed application nimble enough to adapt to your customers' ever-changing needs? What's your unique edge?

There are three ways to gain an "edge" and compete successfully:

- Deliver superior & unique software that your competitor cannot match.
- Deliver similar software functionality - but create it in a very different way that significantly reduces the labor cost.
- Deliver similar software functionality - but create it in a very different way that significantly reduces the time-to-delivery.

IQware gives you all three. With IQware you have an edge.

IQware is both a standalone product and a rule-based development tool. IQware's patented (US Patent #7,322,028) architecture significantly reduces - and may even eliminate - the need for expensive, error-prone custom code development. This patented feature gives you two advantages by greatly reducing **both** time-to-delivery and the total labor hours . This IQware advantage lets you compete effectively with ultra-low-cost offshore software developers by delivering better software solutions sooner and at a total lower cost.



How to Deliver an Application

Traditional way

(WARNING: Only use when you don't care about time and money):

- Time Consuming
- Costly
- Error Prone
- Virus Prone and Insecure
- Complex/Hard to Understand
- Inflexible
- Removed from the business process

IQware's way (U.S. Patent #7,322,028)

(to be used when you do care about time and money)

- Rapid
- Low Cost
- Secure and Virus-immune
- Interoperable
- Flexible
- Able to take the time to understand the needs of the end user

What Happens when your business model changes?

- Re-Code, Re-Compile, Re-Test, Re-Debug - A costly and time consuming process!
- Reinstall the latest versions on all operating systems

- Rule-Based software makes changing your system simple - change a rule, change the outcome.
- Immediate Results - IQware is centrally managed, so changes take place automatically on all operating systems

2.0 A Brief History of the Problem and IQware's Solution

IQware was designed to solve the problem of never-ending, costly custom code development. IQware was also designed to answer the question of how to deploy and maintain critical software applications in an environment of never-ending changes in operational requirements, computer platforms and delivery mechanisms. Before IQware, software engineers and university researchers focused almost exclusively on the development environment. However, this approach simply created better development tools that allowed software developers to create the same bad (or worse) software applications - only faster and cheaper.

Ongoing technology change is the hallmark of the computer and software industries. New hardware products are coming out at an incredible rate, while new software products are being introduced even more quickly. Although the data analysis, information management and control capabilities brought by these new products have certainly been of great benefit, the products, when installed and configured, seldom meet expectations at anywhere near the projected costs.

One significant problem is constant, never-ending software development. Although the capabilities and features of software packages have increased greatly during the last decade, the need for software engineers and programming services has grown even more rapidly. This is true across various types of software including general packages and those directed towards business and manufacturing automation. The successful system integration of common software packages invariably demands that a great deal of custom code be designed, written and debugged, which is a costly and error-prone activity.

Another emerging and largely unanticipated problem is cyber attacks and other malicious software. Nearly all software systems were designed assuming a benign operating environment in which program operation would proceed without interference. Unfortunately, this assumption is violated nearly every day, especially since 9/11/2001. Cyber attacks are increasing in frequency and in severity and vast majority existing software systems are unable to operate properly under such conditions.

"Stop-gap" measures such as firewalls and anti-virus software have been employed to provide some degree of protection. However, none of these techniques and technologies can provide 100% protection. Firewalls have been successfully breached by teenagers and anti-virus software only works for known virus instances. Software skill sets have migrated across the globe with the natural consequence of a dramatic rise in the sophistication of malicious software used in cyber attacks. This situation is expected to worsen as the software tools and technologies available to the attackers rival those available to the defenders.

Systems integration firms and end users are devoting an ever-increasing fraction of their budgets to cyber defense, software design, programming and verification. Pedestrian software activities such as configuring a firewall, creating custom menus, formatting dialog boxes, creating custom displays and the like can usually be done via traditional "point-and-click" or at most by editing a script file. Unfortunately, this traditional level of flexibility is woefully inadequate to meet end users' changing needs in a timely and cost-effective manner.

More sophisticated software efforts, such as implementing a new business rule, conforming to new data analysis & reporting regulations, increasing system security, executing a custom menu option, and the like invariably requires significant custom code development in a computer language such as Visual Basic, Visual C, C++ or Java. Additionally, to obtain the desired level of end-user functionality, still more custom code must be designed, written and debugged. Further, if the original application was not architected well, it may not even be possible to add this new functionality without extensively rewriting it. The cost of these software design and programming efforts is significant and is considerably greater than the cost of the original out-of-the-box software package.

Nearly all companies maintain Information Technology (IT) departments to handle software installations, upgrades, and custom code development. Significant fractions of companies' operating budgets are allocated to IT departments for these tasks. Companies who are not even in the software business must still make large financial and personnel investments in Information Technology. Thus, both human and capital resources are inefficiently expended on tasks far outside the companies' area of core competency and industry. Cyber defense and software security efforts exacerbate this situation and are beginning to consume a significant fraction of both corporate and government resources.

New operating systems and platform products are introduced into the market at a phenomenal rate. This rapid change often requires application software to be redesigned just to continue to operate with the new computer platform, operating system and/or other application. Thus, many companies' software development efforts and expenses are now driven by external market forces that are completely unrelated to their core business or market. Unfortunately, money spent on this type of software development may not even directly improve productivity, product quality or increase sales. It may even negatively impact profitability and cash flow. The same is true for cyber defense and software security efforts. These activities negatively impact cash flow because they are required for successful business operation yet they do not have direct business benefit in terms of sales, product quality improvement or cost reduction.

IQware was invented to solve these problems

3.0 Software Development

Traditional software development and applications are too inefficient and insufficient to be effective in today's fast-paced world. Applications need to be deployed faster, less expensively and they need to work correctly "out-of-the-box". IQware understands the issues involved with software development and directly addresses those issues, offering a real solution.

3.1 The Problem

The problem that IQware addresses is ongoing, never-ending, costly, error-prone software development in the Master Data Management/Business Intelligence (MDM/BI) space. That's it. Every company does it – especially those that are NOT in the software business. In order to solve this problem, we must understand both how and why it happens. Then, we can deal with it.

Traditional software applications are developed through design and programming efforts. Nearly all decisions are made at compile time and only a very restricted set of user inputs is allowed at run-time. There is almost no prior planning - or provision - for significantly altering the behavior of the delivered system in response to significant changes in the operational environment and/or significant changes in the tasks that the system needs to correctly perform. Business needs change and IT can't keep up.

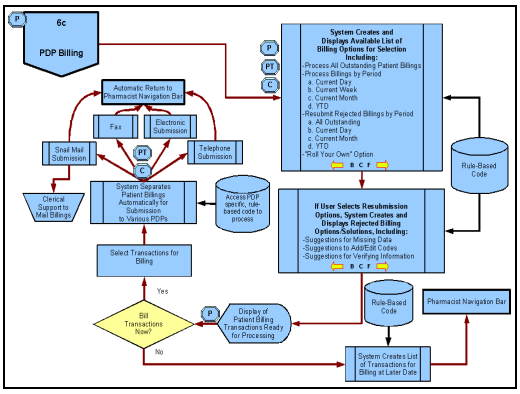
As an example, the vast majority of dedicated data management applications are really subsets of MDM and BI functions that are intended to fill a specific – and immediate - need. That is why most of these applications start out as "home-grown" code, spreadsheet macros, JAVA script, XML scripts, etc. – then they expand over time to a standalone application.

As the organization's needs grow, such applications are edited and modified until they resemble a patchwork quilt – or a plate of pasta – that is nearly unmanageable. Development and support costs increase while functionality does not keep pace. New hardware and platform changes force additional development costs that do nothing for the bottom line and do not provide any new relevant capability.

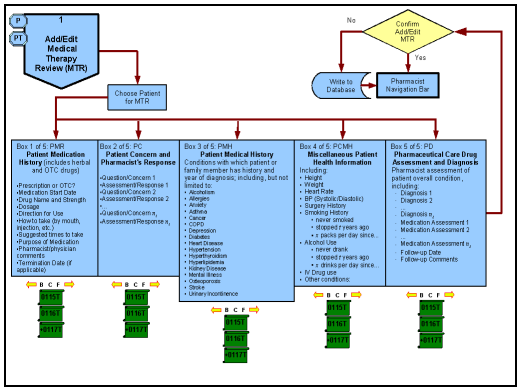
At this point, the dedicated application must either be rewritten or discarded. Either way, it costs a lot of time and money to adapt – not to mention the disruption to your ongoing business operations.

Whichever approach is taken, the new software system must be implemented and deployed. This is always an expensive undertaking and results in a serious loss of operational capability and continuity. The never-ending "code-compile-test-debug" loop always drains dollars and delays delivery. It is the Achilles' heel of all software development.

This situation is so bad that many organizations are forced to find cumbersome "work-arounds" because their IT systems cannot adapt to new functional requirements and operational needs, which forces yet another trip around the lengthy, error-prone "code-compile-test-debug" loop.



As another more concrete example, if a pharmacy wants to implement a Medication Therapy Management (MTM) application based on the flow chart to the right, that pharmacy can do so using traditional software packages. It will take time and dollars to design, build, and test, but a usable system can be produced.



The problem starts when changes are needed. Changes in Federal rules and regulations, software operational requirements, audit procedures, reporting formats, etc. will each cause a change in how the system has to operate. This generates a new and different model, as shown in the bottom flow chart. If the traditional software approach was used, the pharmacy would have to re-engineer, re-compile, re-test and "re-debug" its existing application. Of course, that costs a lot of money, delays delivery and misuses staff. Not good.

3.2 The Solution

These costly trips around the lengthy, error-prone "code-compile-test-debug" loop should not only be avoided – they should be prevented. IQware does exactly that.

Having to go back to your IT department every time you want something changed and going through this painful process is also expensive and time-consuming. IQware eliminates this disruption and lets you keep pace with a changing environment. IQware understands these issues and provides a complete solution. IQware grows as your needs grow, is secure and platform-independent (US Patent #7,322,028).

IQware solves the problem of ongoing, never-ending costly error-prone software development by offering a patented software system that is flexible, interoperable, and displays immediate results whenever a change is made. IQware breaks the vicious, costly and error-prone software development cycle.

3.2.1 Flexibility

IQware's rule-based architecture is far more flexible than traditional software. By design, it can be reconfigured as needed so it can adapt to radical changes in its operational requirements and/or its deployment environment. IQware can also adapt to significant changes in the tasks that the system is required to do. This eliminates the need for extremely expensive re-design, re-programming and re-deployment of large-scale software systems. IQware's design also ensures 100% continuity of operations – even while system changes and/or reconfiguration is underway. This is a unique advantage of IQware and makes it suitable for all critical infrastructure applications.

IQware's software is a rule-based Business Intelligence (BI) and Master Data Management (MDM) system that morphs into the solution you need - without writing new code. By setting up the rules properly, IQware will perform the BI and MDM functions that you want. IQware gets the right data to the right place at the right time. Like alphabet blocks used to spell a variety of different words, rules are arranged and rearranged to serve the particular functions of each application.

On the user side, IQware is rule-based so it is configured to act exactly as you want it to act. By setting up the rules, IQware functions in an intuitive way that makes sense for your specific operations. The rules control how the IQware's screens "look and feel." The rules also control how and where IQware gathers data, how it analyzes data and how it presents information to you.

On the configuration side, IQware is also very easy to use. The rules are set up graphically so you can see what you are doing as you do it.

IQware's rule-based architecture means that rules, rather than traditional programming, govern IQware's operation. This feature is fundamental to the architecture of IQware and is also one of the primary reasons for IQware's tremendous flexibility. Rules control data acquisition, data analysis, data presentation and data storage. By changing the rules, you change how IQware behaves. These changes may be made while IQware is running.

3.2.2 Interoperability

Other applications are not interoperable; i.e., they cannot automatically deploy on different platforms. Using such applications means that different versions of the software must be purchased and installed for each platform. This process is very labor-intensive, costly and error -prone.

IQware's patented (US #7,322,028) rule-based system can deploy across any client platform, including all Windows, Linux, Apple, Sun machines, etc. With IQware, you can upgrade your

computers at any time and be confident that IQware will function perfectly on the new machines. IQware works equally well on ALL platforms, so you only need one version to install and maintain. This feature can save you a tremendous amount of time, money and headaches as you do not need separate software, separate license agreements, separate installations, or new software which requires re-testing, re-debugging, re-checking and re-writing!

3.2.3 Immediate Results

IQware can be centrally deployed and managed so that changes in the rules can take effect almost immediately to the deployed system. If you have a large number of users and/or they are geographically distributed, then a central deployment means that there is never a need to slide a DVD/CD into each machine to install/use/upgrade or react to a changing environment. For command center and first responder applications, this ability is critical because it ensures that as the operational picture changes, everyone sees it right away

4.0 IQware Software Overview

IQware, Inc. makes industry-specific, patented (US #7,322,028) content management and delivery software, sometimes called "*Master Data Management (MDM)"* or "*Business Intelligence (BI)"* Software. Content management and delivery may include medical records, pharmaceutical data, drug interactions, demographic data, business operational data, command / control center data, etc. Content management and delivery may also include POS (point-of-sale) advertising, ad tracking, effectiveness monitoring, etc.

IQware does centrally-managed, individually-tailored, context-sensitive content acquisition, content management, content delivery and content presentation across multiple vertical markets. The software performs interactive data acquisition, analysis, reporting, auditing, presentation, mining and archiving. The unique value of IQware software is that it is secure, immune to desktop viruses, platform-independent and rule-based.

IQware's MDM/BI software translates raw data into useful information through assured, accurate interactive analysis and presentation. IQware helps convert that information into knowledge by supporting its successful application - and then by managing the results.

IQware also provides information assurance through a variety of patented (US #7,322,028) internal, secure mechanisms. Information assurance is an essential component of MDM because errors embedded in the "original content" – or in the acquired raw data – are significantly amplified and magnified by the "downstream" IT systems. Such IT systems have few reliable, independent mechanisms(s) for information verification. The consequences of such errors percolating through the organization are severely expensive at best - and fatal at worst.

IQware can acquire data from any source. This makes IQware ideal for deployment across existing and disparate IT systems. IQware's products also extend rather than compete with the functionality of traditional Enterprise Resource Planning ("ERP") systems.

IQware focuses on the regulated industries, which have the following requirements and/or desires:

- Tailored, secure content management, analysis, delivery and presentation
- Information assurance and validation
- Interactive and comprehensive reports
- Interoperability and compatibility with new, emerging technologies
- Many disparate data sources
- Multiple disparate IT systems
- Very high security with complete, tamper-proof audit trail
- Integrate information disparate IT and legacy IT systems
- Real-time (or nearly so) performance

These industries generally have one of more of the following characteristics:

- Deployment of legacy systems
- Inadequate information assurance
- Inadequate information integration
- Inadequate tracking capability
- Inadequate auditing capability
- Inadequate reporting capability
- Cannot meet regulatory compliance with existing IT systems
- Cannot meet performance requirements with existing IT systems

IQware's patented (US #7,322,028) software has three critical attributes:

- It is hacker-proof and immune to desktop viruses (US Patent #7,322,028)
- It is rule-based so it can be changed on-the-fly and without programming
- It is interoperable so it works with emerging hand-held wireless devices

With heightened customer interest in IT security, IQware software provides customers a solution to secure software application requirements. The rule-based nature of IQware's products significantly reduces or eliminates the need for expensive custom code development. This unique, patented (US #7,322,028) feature allows third-party system integrators to compete effectively with ultra-low-cost offshore software developers because they can do a better job in far less time using IQware – *and* provide virus-immunity as well.

IQware's products are currently targeted toward the following industry segments:

- Pharmacies (retail MTM implementation and regulatory compliance)
- Pharmaceutical production (regulatory compliance, manufacturing)
- Health care (regulatory compliance, information integration, reporting auditing)
- Federal government (communications, auditing, tracking, analysis)
- Military (DoD, information integration, analysis, reporting, C^3I)
- State / Local government (infrastructure monitoring, control and auditing)

5.0 IQware's Approach to Application Development & Deployment

5.1 Application Development

The Rule Processor (RP) is what reads & executes rules. These rules contain the parameters for the events and actions that comprise the rule. Rules are of the form "upon event(s), take action(s)". Mathematically, rules are of the form "if $f(e_i)$ then do $g(a_i)$" where $\{ e_i \}$ is the set of allowed events and $\{a_i\}$ is the set of allowed actions. The events and action definitions are sufficiently general to handle a wide variety of "business process flow" requirements. The event-action pairs are orthogonal so that any event(s) can cause any action(s) to be taken – no limitations.

The Rule processor uses the "Reference Monitor" security rules where required for proper enforcement. A portion of the rule parameters contain "O/S permission" parameters, "users allowed" parameters and "access mode" parameters. The security processing is done in one place so that it is easy to test, verify and modify.

IQware has a utility, called IQ Build, that sets up the rules. IQ Build is an object-oriented graphics editor that is "WYSIWYG" so that the application appears as you would like it to appear as you build it. This graphics editor is the main tool for creating an interoperable, secure, rule-based, centrally managed application.

IQware's "rules" are like letters of the alphabet. Letters are the basic building blocks from which words (and by extension language and meaning) are created. The alphabet of the English language has not changed for hundreds of years, but new words are constantly being created and meanings are constantly evolving.. For example, the phrase "disk drive", nonsensical in Shakespearian times, has a concrete meaning to today's audience – no new letters had to be invented in order for the phrase to gain meaning. IQware has all of the letters (rules) necessary to create new phrases or functionality, they only need to be configured to meet your needs. Because of this patented (US #7,322,028) rule-based process, IQware meets the changing needs of any business in a cost-efficient, timely manner.

IQware's rules are categorized according to function. The functional categories of items that are governed by rules are the following:

- Object/Screen display
- User interface (UIF) object rendering and control
- Database operations
- Data acquisition
- Data analysis
- Data transfer (from anywhere to anywhere)
- Report preparation, formatting and output
- External O/S actions (e.g., run a job, execute command procedure, etc.)

Rule elements are grouped into eleven (11) classes. Classes 2 through 11 each require a parameter list for complete specification of the parameters for the rule. The “summary” class does not require a parameter list.

- Summary (format, ID, location, status, next)
- Event(s)
- Action(s)
- Data Source
- Data Destination
- SQL Source
- SQL Destination
- O/S Permissions
- Access Modes
- Users Allowed
- Audit rules

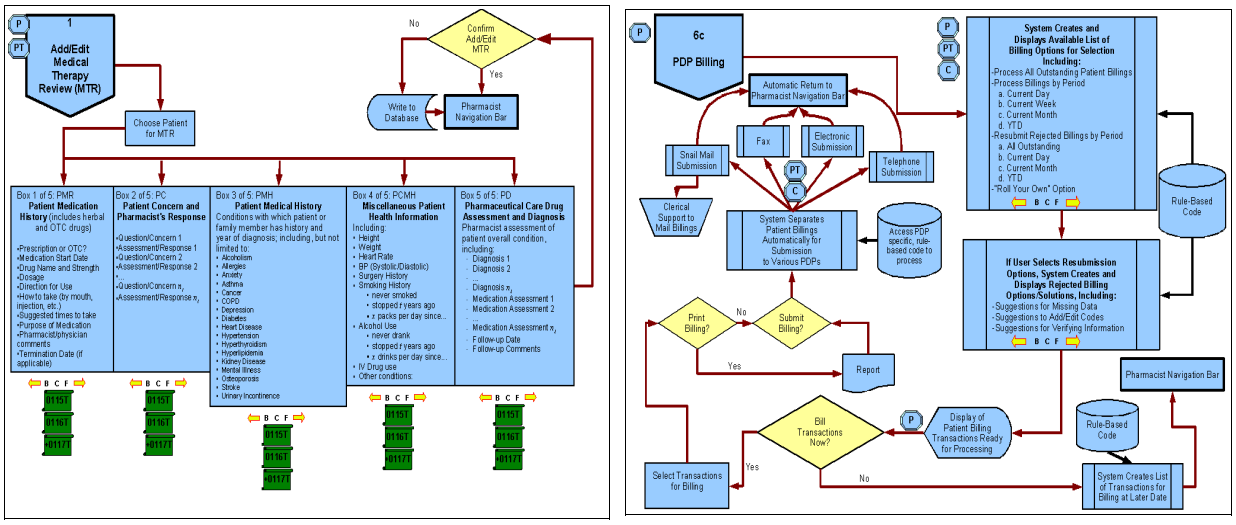
5.2 Application Deployment

Rules are run in a separate program, Rule Processor (IQ_RP). Rule processor allows you to interact with the rules as defined in Build. One of the hallmark features of IQware's patented (US #7,322,028) rule-based software is that changes can be made "on-the-fly." Changes in process do not have to be re-coded, re-compiled, re-tested, and "re-debugged"; instead, rules need only be reconfigured or added. Once configured – a process that is complete in a matter of moments – the rule is fully-functional. The user does not even have to log out of his or her session in order to implement the new function.

5.3 Using IQware Software

Because this rule-base architecture is new, an example of how IQware functions in the real-world is beneficial. IQware automates your business process using Build Utility and Rule Processor.

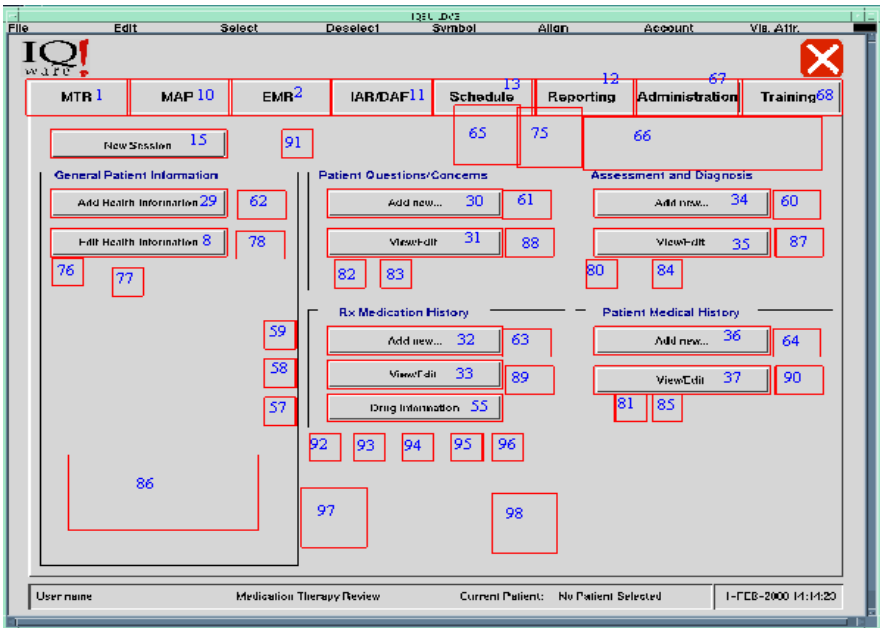
One Example: Designing a Medical Therapy Management System Using IQware
The process of designing an interface using Build begins with the IQware team creating a process flow chart incorporating all of the aspects of your business process. Two examples which require process flow charts (pictured below) are actions required to perform a Medication Therapy Review (MTR) session (left) and billing for pharmacy activities (right). These actions include gathering a patient's medical information and assigning proper billing codes among many other activities necessary for auditing and regulatory compliance.



IQ Build creates an interface incorporating all of the attributes of the business flow chart. Each screen consists of a variety of symbols, some of which are rules, each of which can be configured and re-configured "on-the-fly" based on the changing needs of the client. Functionality, as well as other attributes like size, position, and color, can be reconfigured using IQware's Build utility without the need for reprogramming or system down-time as regulations and businesses processes inevitably change.

Configuring Rules

As another example, rules may be edited on the completed MTR screen (see below, outlined in red and numbered) to alter functionality, thus eliminating the need for reprogramming or upgrading to a different software version.



Examining the four rules associated with beginning a new MTR session provides a good example of the flexibility of application built with the Build Utility. The four rules are as follows:

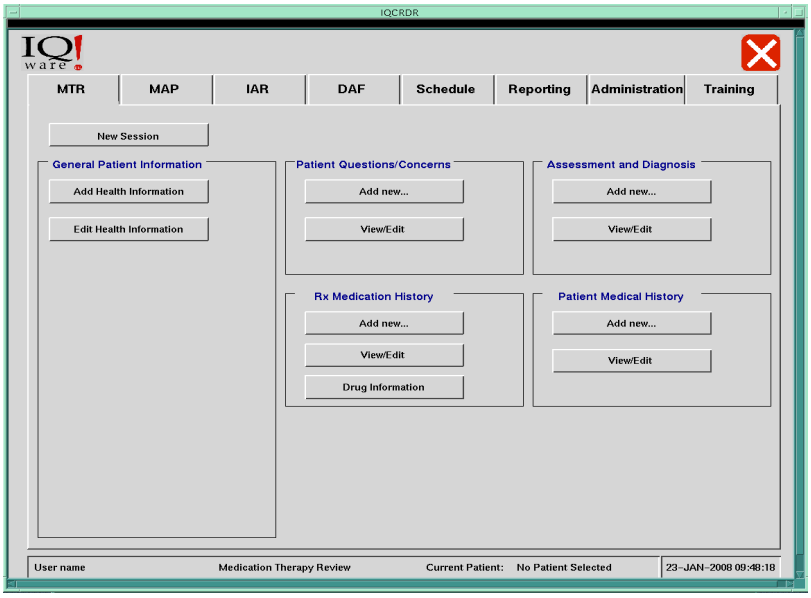
- Rule 15 produces the above Data Entry Box (DEB) prompting the pharmacist to enter information about a patient's current MTR session. This is the only rule associated with this action that is visible to the end user.
- Rule 65 states that when the user clicks "OK" on Rule 15, patient information is retrieved from the database and stored internally for use during the session.
- Rule 75 states that when the information is retrieved for Rule 65, a different piece of patient information is retrieved from the database, also stored internally for use during the session.
- Rule 66 states that when Rule 75 is complete, all of the information regarding the new session is stored in the database. This step ensures that the database records are saved every time a new session is started for a patient (for auditing, billing, and regulatory compliance).

The end user is only required to click onto the screen one time in order to execute all four of these rules. If the parameters for starting a new session or the amount of information the pharmacy would like to store regarding a new session changes, you would only have to alter one of the rules, or simply add a new rule anywhere on the screen to make this change - the physical location of the rule plays no role in the functionality, only the rule configuration matters.

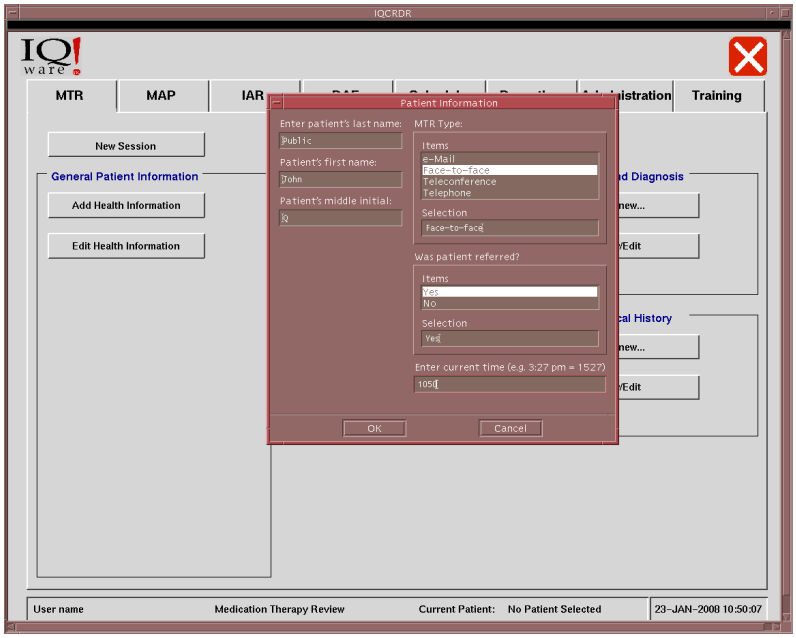
This same principle can be applied to to screens for any application.

Using Rule Processor

The appearance of the screen in IQ_RP is identical to that in Build, with the exception of the lack of menu bar at the top of the window. In the MTM example, the pharmacist sees the following screen:



The rules are executed by completing actions as specified in Build. For instance, Rule 15 (as discussed above) states that "On Mouse Click, Use DEB" (the DEB has four rows and two columns). Therefore, when the user clicks on the rule (labeled "New Session"), a 4 row by 2 column DEB appears:



Functioning rules that do not specify the action "On Mouse Click" cannot be initiated by the user. Instead, those rules involve retrieving, transferring, and/or arranging information either stored internally for use during the session or externally in the database.

6.0 Security

Inflexibility is not the only problem inherent in traditional software packages; traditional software packages are not secure. IQware, on the other hand, is built from the ground up to be secure. Unlike traditional software, IQware is desktop-virus-immune and does not require a "band-aid" or "magic software" in order to be so.

6.1 The Problem

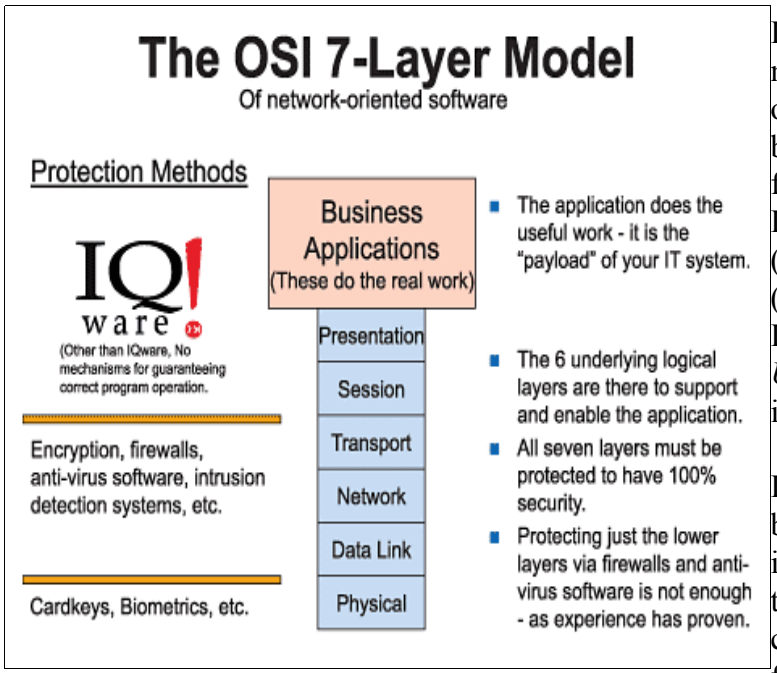
Web-based apps are not completely secure and encryption does not make them completely secure. The web browser simply interprets an HTML string (or XML string) and implements the instructions contained therein, which may include the launching of JAVA (or similar) applets. The problem with that approach is that sensitive information is contained in XML/HTML string. A well-written spyware app can easily access this information (in decrypted form) and send it anywhere. This is a growing problem across all markets.

Software that has not been built correctly from the beginning cannot be 100% secured because the flaws are built-in. In order to "fix the problem", all existing applications would have to be discarded and redesigned from the ground up. The huge installed base prevents this activity from happening because of the tremendous cost and inconvenience. Software makers are simply not willing to expend that kind of time and money on this, so their software continues to have flaws and is vulnerable to attacks from computer viruses, cyber attacks, lapses in computer security, etc. Furthermore, no software can transform an insecure system into a secure system.

Insecure System "Magic Software" Secure System

6.2 The Solution

If a house is built on a faulty foundation, no matter how competent the carpenters, the house can still fail structurally. IQware's foundation is secure and its entire architecture is secure, so using IQware keeps users free of unwanted intrusions and faults. IQware's unique software is architected from the ground up to be secure. It uses the TCB's ("Trusted Computing Base") proven security attributes to ensure compliance with object access and activity audit trail requirements. Other competing MDM/BI systems are not architected this way and so they inherit all the bad attributes of the operating system and any other "built-in" software products that they may have used.



IQware is secure and virus-immune, meaning that IQware's fundamental operation will not be compromised by any desktop virus, worm, script file or other malicious software. IQware's "Trusted Computing Base" (TCB) is rated DoD B2/C2 (depending on the version). Further, IQware's TCB was rated as *"Cool & Unhackable"* at DEFCON 2001 held in Las Vegas (see www.defcon.org).

IQware is secure and virus-immune because it is architected that way – it is not a security blanket or band-aid to use after the fact. IQware is quite complementary to the various firewall and anti-virus products on the marketplace. As the diagram on the left illustrates, software can be modeled as a 7-layer structure. These layers are logical layers that perform different functions. Each layer performs function to the layer above it in the diagram. The top layer is the application layer – it's the layer that does the useful work. The underlying six layers are only there to support the top, or application layer.

Effective security requires proper architecture, coding and deployment of the O/S, the application, and all layers in between. A secure system must feature:

Policy

- **Security Policy** - System must enforce a well-defined security policy.
- **Marking** - System must associate all objects with access control labels (sensitivity & access modes).

Accountability

- **Identification** - System must identify individuals and their various authorizations in a secure manner.
- **Audit Trail** - System must keep & protect audit trail so actions may be traced to responsible party.

Assurance

- **Evaluation** - System must have hardware/software mechanisms that can be independently evaluated to assure that policy & accountability are enforced.
- **Continuous Protection** - System must continuously protect trusted mechanisms that enforce policy & accountability from tampering.

IQware does all of this.

7.0 Feature Summary

IQware is both a stand-alone application and a tool to implement a dedicated application that delivers the right data to the right place at the right time. IQware is rule-based which means that the capabilities and functions that IQware delivers are controlled by rules. All decisions by IQware are made at run-time – not at compile time. When you change a rule, you change what IQware does – right away – with no programming required. This unique IQware capability delivers unprecedented functionality and is patented (US Patent #7,322,028).

In short...

IQware's solutions will:

- Improve processes
- Improve operations
- Integrate disparate databases to avoid duplicate efforts
- Run more effectively and efficiently than "traditional" software
- Provide an audit trail on all transactions and interactions: accountability
- Provide Common Operational Picture
- Allow Real-time Access
- Eliminate Information Overload
- Perform Threshold Analysis in Real Time
- Maintain an audit trail of all events

IQware helps you:

- Reduce, or avoid, Costs (efficiency)
- Reduce, or eliminate errors (avoid rework)
- Create data visibility across organization at both summary and detail levels
- Improve management planning and control
- Provide analytical work versus transaction processing

IQware Solutions are able to provide these solutions via patented technology

- Rule-based architecture:
- Significant cost reduction because changes in operational rules do not require changes in software: re-validation not necessary
- Reduces need for expensive, error-prone custom code
- Greatly reduces system integration time because custom code is not required, quality is improved

- Secure Architecture:
- Protects government agencies from costly recovery from hackers and viruses
- Protects against disruption, enterprise data theft and corruption
- Maintains audit trail of all security-related events, maintains tamper-proof Access Control List (ACL) - meets Sarbanes-Oxley requirements

IQware knows the problems created by traditional software applications. They are inflexible, they are not interoperable, they are expensive, and they are not secure.

IQware is the solution to these problems.