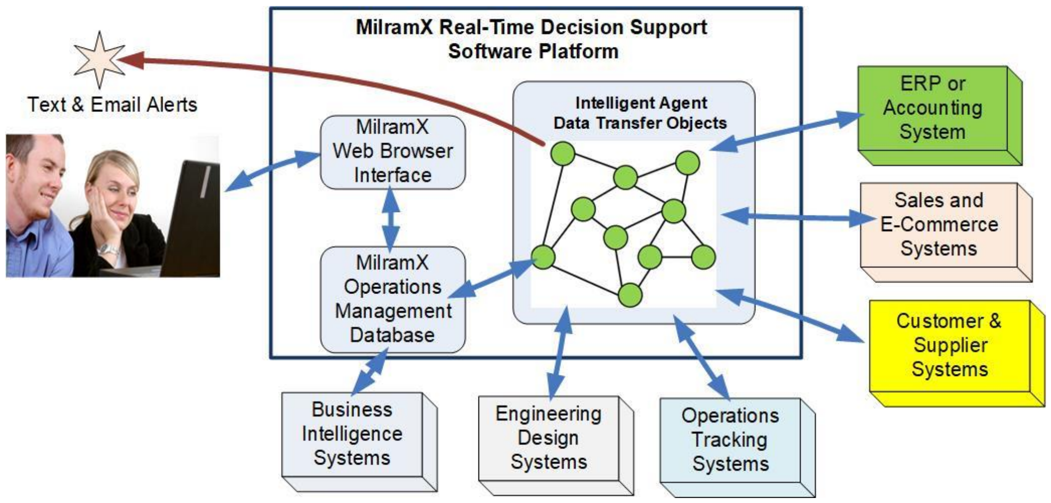
MilramX Enterprise Real-Time Decision-Support Software Platform Technical Overview


MilramX Real-Time Decision Support Software Platform
Text & Email Alerts
MilramX Web Browser Interface
MilramX Operations Management Database
Intelligent Agent Data Transfer Objects
ERP or Accounting System
Sales and E-Commerce Systems
Customer & Supplier Systems
Business Intelligence Systems
Engineering Design Systems
Operations Tracking Systems

Introduction

MilramX™ is a software platform for rapidly implementing real-time decision-support applications which are capable of working in both an autonomous decision-making as well as an advisory mode.

Typical applications include:

- Automatically processing customer orders into actionable purchase, receiving, manufacturing, picking, packing, shipping and installation orders.
- Automatically planning and scheduling operations in real-time to make sure that customer orders get shipped, delivered and installed on time.
- Tracking the flow of customer orders and materials through multiple manufacturing, distribution, and installation sites.
- Automatically providing people throughout an organization with the real-time operations management information they need to do their jobs efficiently.
- Automatically exchanging information with supply-chain trading partners for operational coordination and materials traceability.
- Warning people when they are about to make an operational mistake and automatically alert managers when situations arise that they need to pay attention to.

These and other operations management problems that KnarrTek solves with its real-time AI based technology enable efficient management of operations within industrial organizations with less people. This results in increased profits for KnarrTek’s clients and solves the problem of recruiting, training, and retaining skilled staff people. Many of these solutions also enable an increase in sales by speeding delivery and improving customer satisfaction.

Development of MilramX

The driving force behind the development of MilramX was to provide a software platform which would make it quicker and less costly to implement real-time AI systems, which:

1. Are able to process data from many different sources in near real-time, in parallel, as within an industrial enterprise many events can occur at the same time which impact the management of operations.
2. Are able to find and detect changes in data from many different sources and develop or change action plans or other recommendations, based on those updates.
3. Are able to communicate these plans, schedules, and recommendations to people via the systems they use on a routine basis, including sending Emails or Text messages to their mobile phones.
4. Are able to automatically exchange needed information with supply-chain trading partners, including materials traceability data.
5. Are able to integrate data from a variety-sources into a single knowledge-base which can be used for interactive decision making by operations managers and their staff.

These real-time AI applications differ in their decision making from regenerative AI applications, such as Chat GPT, which are being hyped in the media.

Real-time AI applications can analyze and act on new data as it comes in, in near real-time, from multiple sources, without human intervention. Regenerative AI systems, by contrast, answer questions posed by people by searching the contents of a large volume of essentially static data, such as from the Internet.

The goal of real-time AI systems is to quickly provide a workable decision that can be refined over time. By contrast, the goal of a system like Chat GPT is to provide the best information that it can, on a one-time basis, without time limitations on its search (except for people’s patience).

Both use models in their decision processes. MilramX applications use models of manufacturing, distribution, service and installation processes. Chat GPT and similar applications use a Natural Language Process (NLP) model to interpret people’s questions and structure the result into a query that can be executed against their data source(s). Regenerative AI processes then use the results to create an appropriate NLP response or even to generate images. Very clever, but not real-time AI.

AI applications, like Chat GPT, facial recognition, and the detection of breast cancers, rely on non-linear adaptive matrix correlators (so-called Neural Networks) which can be trained on large historical data sets. Real-time AI systems, by contrast, learn primarily from data as it arrives in real-time, using a variety of algorithms. Thus, a MilramX based system may learn how long it takes to make or deliver products based on real-time data, as it is collected over time, by a system such as BellHawk, and then use this in its automated decision making.

Real-Time AI based systems such as MilramX and Neural Network based AI systems, like Chat GPT, both have their roots in research work funded by DARPA and other US Government agencies and may use many of the same algorithms but in different ways.

Early versions of real-time AI systems put men on the moon and are used today in self-driving vehicles, industrial robots, in robotic vacuum cleaners and many other modern conveniences.

Development of intelligent-agent real-time AI software platforms, such as Activation Framework, which were funded by the US Air Force and NASA, focused on real-time decision support applications such as detecting when problems arose with an aircraft and advising pilots how to react. This led to the development of MilramX and its application to manufacturing and distribution operations-management problems, where early work on distributed real-time AI, funded by DARPA, also found its application.

With MilramX, intelligent agents monitor many sources of data in real-time to determine when actions need to be taken and then either autonomously make the appropriate decisions or alert managers or their staff when action needs to be taken and then provides them with the needed information.

MilramX is frequently used with the BellHawk real-time operations tracking software in operations management decision support applications. BellHawk itself integrates a number of real-time AI algorithms to make it easy to capture operations tracking data and to warn equipment operators and materials handlers when they are about to make a mistake. BellHawk can also make short term operational decisions, such as adapting schedules when problems arise.

MilramX can, however, also be used without BellHawk, basing its decisions on data from a wide variety of other systems.

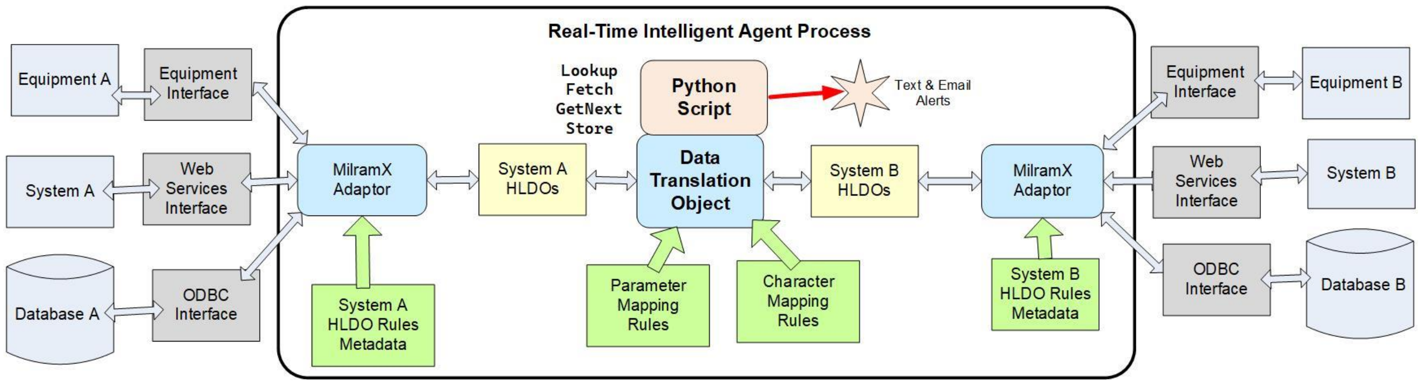
Intelligent Agents

The traditional way of writing operations management decision support applications is as a single large program. The issues with this approach, in real-time applications, are:

1. Many things happen continuously at multiple locations within an industrial enterprise. By the time a single application has finished sequentially analyzing one situation, many other events may have occurred. As a result, many decisions do not get made, or people do not get the information they need, until it is too late.

2. You need a large amount of computing power to try to make the large singular application run fast enough to meet the needs of all the processes and people involved.

3. You need to shut down and recompile the decision support application every time there is a change to operating procedures or people’s needs within the organization or additional information needs to be sent to trading partners. This can be problematic in many industrial applications where 24x7 operation is required



By contrast, MilramX uses the paradigm of intelligent agents which are run as separate processes in parallel on multi-core processors running the Windows operating system. In this way each intelligent agent process can monitor a one or more sources of data for changes and then make decisions, advise people, and/or communicate with other agents in near real-time.

The algorithms used by these agents may range from a simple set of rules to complex non-linear self-adaptive statistical correlators. While these algorithms typically depend on standard libraries, it is important to recognize that the decisions made by each agent are typically unique to each specific organization and how it does business.

The solution adopted by MilramX to solve this problem is to make as much use as possible of third-party software libraries for performing mathematical and other real-time AI computations and then to use a Python script as the application specific “glue” to contain the decision support rules and other such computations.

The benefit of this is that the Python script can be dynamically linked to the intelligent agent process, as can the Dynamic Link Libraries (DLLs) that it uses. This enables the Python scripts to be dynamically updated while the overall decision support application is running 24x7.

Another problem solved by MilramX is how to minimize the complexity of exchanging data with complex databases, web-services interfaces, or other application programming interfaces. In this MilramX:

1. Provides Python users with simplified instruction set of Store, Fetch, Get, and Lookup to interact with a wide range of external systems. These simplified instructions consume and produce named sets of **parameter-name:value** pairs, called High Level Data Objects (HLDOs), in the form of JSON strings.

2. Having, these simplified instructions reference named Adaptors, which translate between the simplified instructions, with their HLDOs, and the actual interfaces to databases, web-services interfaces, and APIs.

3. Provides a rules-based engine to check on the validity of the data being exchanged with each external system, as well as to assist in the translation to and from HLDOs. These rules are stored as XML metadata which can be imported and edited in the form of Excel spreadsheets.

4. Provides a mechanism to enable a Python script to easily send Email and Text messages to people and their mobile phones.

5. Provides a mechanism to enable a Python script to easily deliver files, such as by secure FTP or as a secure mail attachment, to a remote system.

The code for different adaptors, such as that for interfacing to different SQL databases, are integrated into each transfer process as DLLs. Also, each single adaptor may use metadata for many different sources of data, such as different tables within a single database. In this way, a common DLL containing the code to translate between many HLDOs and the physical interface to a device interface, such as an ODBC interface to a SQL database, may be shared between many intelligent agent transfer processes and yet executed separately, in parallel.

The Python script and its immediate interface to the adaptors is called a Data Translation Object or DTO. These can also be written in a .Net language, such as VB.Net, for more complex applications. Doing so, however, eliminates the ability to dynamically change the SQL script, as the whole of the transfer process then has to be recompiled in order to make changes to the behavior of the intelligent agent.

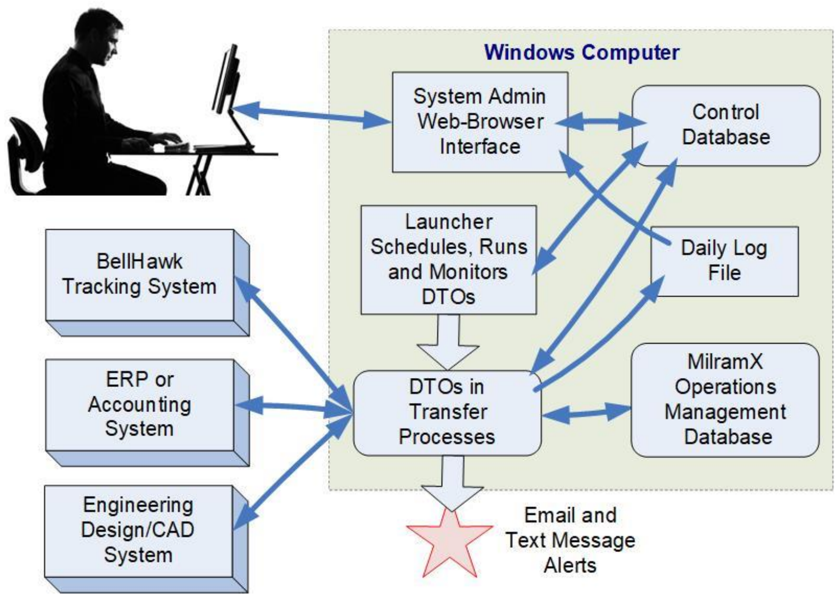
For simple rules-based applications, the translation between the HLDOs from one adaptor to another can be driven by rules loaded into MilramX and stored in its Control database as can the translation between different characters embedded in the HLDO strings. A standard DTO can then be used, without scripting, to perform simple real-time data exchanges between databases or other devices.

The transfer process, when it first starts running, reads an initialization file which contains all the initialization information for the adaptors. It then runs the DTO code and Python script, to carry out the designated transfers.

One major advantage of this approach is a substantial reduction in the amount of computing power required as intelligent agents typically only work on changes to data, as they occur, and do not have to search large databases every time they need an answer. Also, if needed, intelligent agents can be spread across multiple IIOT (Industrial Internet of Things) computers at different sites for enhanced performance and redundancy.

As well as being used to monitor and analyze updates to data sources and to use this data for planning, scheduling and advising, DTOs can also update MilramX' operations management database. This database is used to contain the current status and history of operations within the enterprise which can be interrogated through a web-browser interface. It can also serve as the source of information used by DTOs as well as be interrogated by external business intelligence applications.

Scheduling and Monitoring DTOs


Windows Computer
System Admin Web-Browser Interface
Control Database
Launcher Schedules, Runs and Monitors DTOs
Daily Log File
BellHawk Tracking System
ERP or Accounting System
Engineering Design/CAD System
DTOs in Transfer Processes
MilramX Operations Management Database
Email and Text Message Alerts

Experience has shown that it is much better, in a real-time decision support system, to use a large number of simpler intelligent agents, each with its own specialized function, which work together, rather than a small number of much more complex agents. This facilitates the easy development, deployment and upgrading of individual agents without the need to shut down the whole decision support application.

But this then raises the question of how do we rapidly develop all these agents and, even more importantly, be able to add, modify, and schedule agents, while a large decision support system is running in real time, with many people depending on the information it produces.

MilramX solves this problem by using:

1. Data Translation Objects (DTOs) which are written as Python Scripts or .Net Subroutines. These are encapsulated in Transfer Processes, along with all the needed supporting DLLs (Dynamic Link Libraries) and run in parallel as separate (.exe) processes under the Windows Operating System, preferably on computers that can support the concurrent execution of many code threads.

2. A MilramX Control Database – this is a SQL Server database that contains the scheduling and status information about the DTOs, as well as data about the latest time and date each DTO was run and when each DTO is scheduled to run next. It is also used to hold data such as error messages and other data used by the DTOs.

3. A MilramX Control Website that enables remote monitoring and control of the transfer processes. Through this interface, MilramX administrators can schedule and monitor the DTOs, see and investigate errors, and edit and retry any transfers that failed.

4. A MilramX Launcher program. This Launcher.exe program runs continuously as a background system process. It monitors the setup and status data in the Control database to

decide when to run each DTO and then launches the corresponding transfer process. It also monitors the transfer processes and tracks for transfer processes that have exceeded their maximum run-time and kills them, if needed.

5. A repository of Python scripts, which are referenced by the DTO setup data stored in the Control database. These scrips can be changed "on the fly" by the Administrator without needing to recompile any code or restart MilramX, enabling the addition or updating of transfers while other transfers continue to run 24x7.

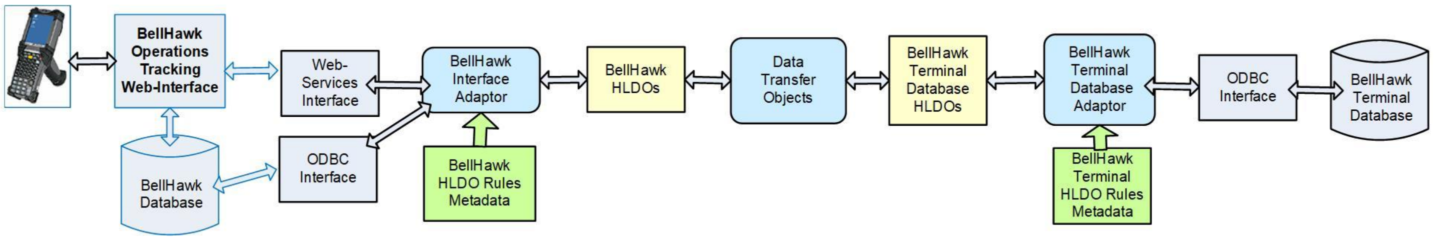
One of the issues that MilramX solves is that data transfers between systems can take a long time and, if these transfers occurred one at a time, they could take a very long time.

To this end, MilramX does the following:

1. By default, only transfers the latest updates to databases or latest data available through web-services interfaces and then stores the result in the target database or system.
2. Enables users to schedule when transfers occur to minimize the needed transfers. This is done through a web-browser interface provided with MilramX.
3. Encapsulates data transfer code objects (DTOs), plus adaptors and other supporting code into transfer processes which can be executed in parallel.

This is especially beneficial if MilramX is run on a multi-threaded processor, where the transfer processes can be executed in parallel.

Reducing DTO Code Development Time


BellHawk Operations Tracking Web-Interface
BellHawk Database
Web-Services Interface
ODBC Interface
BellHawk Interface Adaptor
BellHawk HLDO Rules Metadata
BellHawk HLDOs
Data Transfer Objects
BellHawk Terminal Database HLDOs
BellHawk Terminal Database Adaptor
BellHawk Terminal HLDO Rules Metadata
ODBC Interface
BellHawk Terminal Database

If we started with a clean sheet of paper and designed and coded each data object transfer from an external system to and from the terminal database from scratch, this would take about 8 hours of code development each. This is especially true when you come to recognize that as much as 80% of the code is devoted to data checking and making sure that bad data from one system does not corrupt another system. Most of the remainder of the time comes from dealing with the idiosyncrasies of the database structures or web-services interfaces.

MilramX typically provides over 90% of the needed code pre-built or automatically generated by:

1. Using the Tau-Adaptor rules-based engine to dynamically generate the code needed for translating Store, Fetch, and Get Next Record requests into SQL or web-services interactions with the external system. This includes performing extensive error checking on the data being transferred, based on the metadata rules.

2. Automatically translating data stored in complex database structures, with many indirects, or needing multiple web-services calls, to and from simple High-Level-Data-Objects (HLDOs) which are essentially named JSON strings, consisting of name:value attribute pairs.

3. Providing a Mapping Rules based mechanism to use rules to automate the translation from one HLDO to another.

4. Providing a mechanism to use Python scripts to convert from one or more source HLDOs to one or more target HLDOs, without the business analyst who is writing the script needing to know anything about the complex database structures or web-services interfaces of the source or target system.

5. Providing support for sending text or Email messages as the result of a Python DTO script being executed. Also providing support for sending files by FTP or as Email attachments.

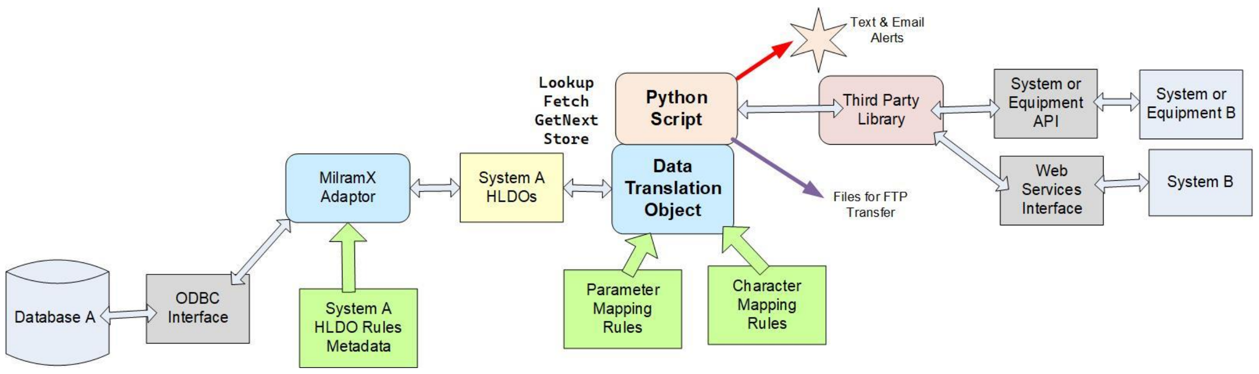
In this way, we are typically able to reduce the development time per data object to less than two hours each plus the time to develop the interface adaptor rules, and any custom adaptor code, for each system.

The HLDO rules and the Mapping rules can be imported into MilramX in the form of Excel spread-sheets, through the system administrator's interface, and then converted to XML metadata files for rapid access by the data transfer objects.

The interface adaptor rules can be separately developed by people who are familiar with each system being linked through MilramX, whereas the data transfer scripts and rules can be implemented by a business analyst, who is familiar with each organizations business practices.

In addition, KnarrTek provides pre-built DTOs and interface adaptors for the BellHawk operations tracking system. These adaptors can use either an ODBC interface when on a common LAN with MilramX or BellHawk's web-services interface when BellHawk is remote.

External Interfaces



Instead of using the MilramX adaptor mechanism, MilramX DTOs can also use third party libraries to interface to external systems. This can be especially beneficial when all the work has been done by someone else to reduce the complexities of interacting with systems which have complicated interface protocols.

There are now many excellent Python support libraries for such interfaces, which can easily be integrated into a MilramX based information exchange application.

Using this approach still retains all the advantages of a real-time AI, intelligent agent approach, but again can minimize the time and cost to develop these real-time decision support applications.

For More Information

For more information about MilramX and KnarrTek's software, services and expertise to implement real-time decision support systems, please see www.KnarrTek.com or send an Email to Sales@KnarrTek.com or call (774)415-7878.