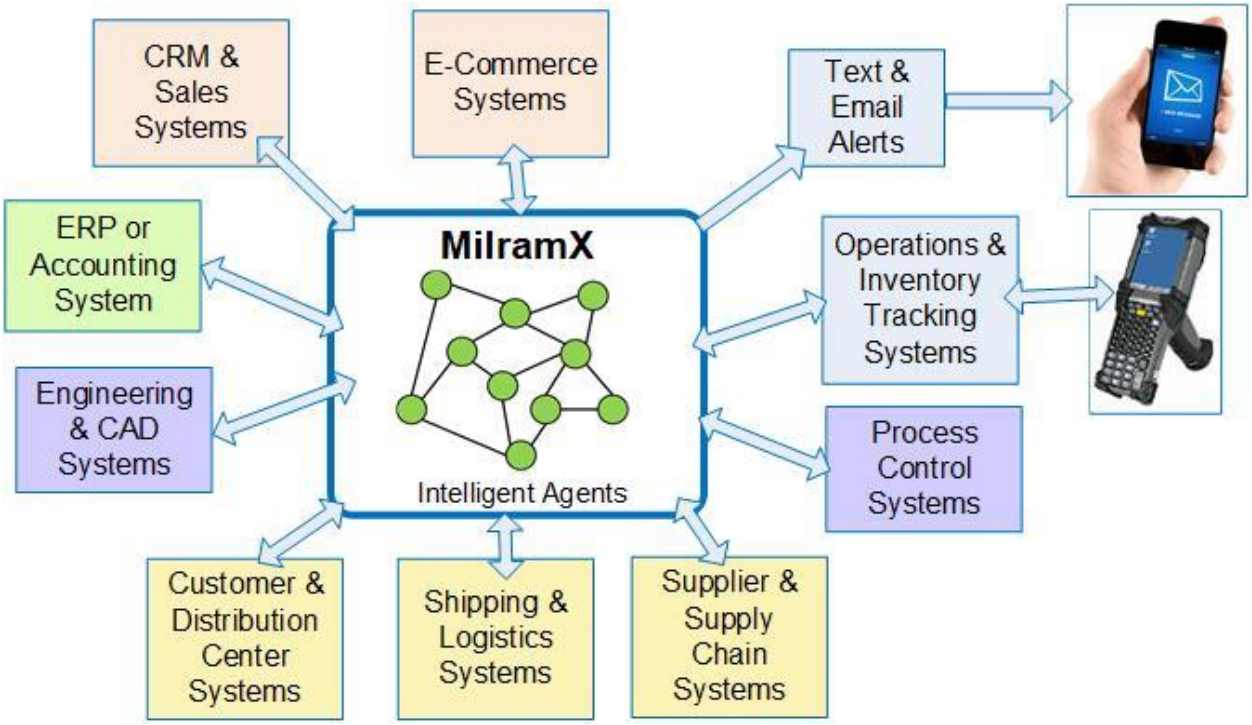
MilramX™ Overview

Introduction



MilramX is a software platform whose goal is to make it easier, faster, and less costly to develop Industrial Real-Time Decision Support Systems (DSSs).

The purpose of a real-time industrial DSS is to provide people within an industrial enterprise with the information they need to do their jobs, when they need it, in a format that is easy to use.

This can be as simple as transferring data between two systems, used by different groups of people, to avoid the need for duplicate data entry. Or it can be monitoring the level of inventory in stock and automatically reordering replacement materials. It can also be as complex as facilitating the interactions between hundreds of people within an organization at many different locations.

At the heart of MilramX are intelligent agents, called Data Transfer Objects (DTOs) which are code objects that monitor data updates from a wide variety of systems and then update other systems and/or send Email or text messages to people advising them that they need to take some action. These DTOs can also incorporate Artificial Intelligence algorithms, such "Neural Networks" (non-linear self-adapting statistical correlators) which learn the parameters of operational models to predict expected supplier delivery times or how long it will take to make a specific customer order.

While many Decision Support Systems start out as relatively simple applications, many grow over time to be more complex than most high-end ERP systems.

The primary purpose of MilramX is enable systems integrators to manage this complexity by dividing large complex decision support problems into a set of Intelligent Agent DTOs that can be developed and tested independently and then incrementally integrated into the DSS as it grows in size and scope.

DTOs are code objects (like subroutines) which are independently scheduled to run to perform tasks when needed, rather than being organized as subroutines in a strict top-down hierarchy which is done in conventional software for systems such as ERP (Enterprise Resource Planning) systems. This enables many DTOs to run asynchronously, carrying out tasks on behalf of people, which these people would otherwise have to do themselves.

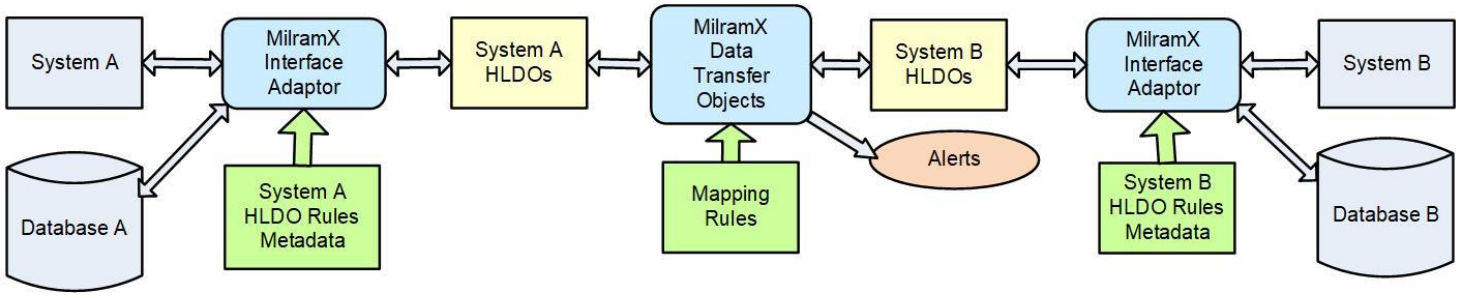
Conceptually, in a MilramX DSS, we have an army of intelligent agent DTOs working tirelessly round-the-clock doing skilled grunt-work so that people can concentrate on doing their jobs and using their experience and knowledge to make good decisions without having to spend a lot of time finding the data they need. A MilramX DSS can also alert people to issues they would have otherwise missed because they were not looking at the right data at the right time.

Because of the complexity of many decision support systems and the fact that these systems evolve over time, as the needs of their enterprise change, it is not realistically possible to write a detailed specification for the decision support overall system. And, even if this were possible, the specification would be obsolete by the time it was completed and approved.

With MilramX we use an Agile implementation process, where the DTOs are developed incrementally as the Decision Support System (DSS) is incrementally deployed. The DSS then becomes a living entity that can be readily changed as the needs of the enterprise change over time.

While some DSSs only have a few DTOs many have hundreds, and may grow to have thousands of DTO intelligent agents. To this end, MilramX incorporates many mechanisms to make it easy to develop, manage, schedule and monitor these DTOs, as will be described in subsequent sections of this overview.

HLDOs and Adaptors



If DTOs had to interact directly with the databases or web-services interfaces of the various systems which they need to interact with then these DTOs would become very complex as they would have to deal with a wide variety of data structures and data formats, which may be very different for each of the systems they interact with.

To this end, MilramX uses the concept of Interface Adaptors which translate between the complexity of a database or web-services interface and High Level Data Objects (HLDOs)

HLDOs are essentially named JSON strings containing parameter name:value pairs. For example, a Customer HLDO may be represented as

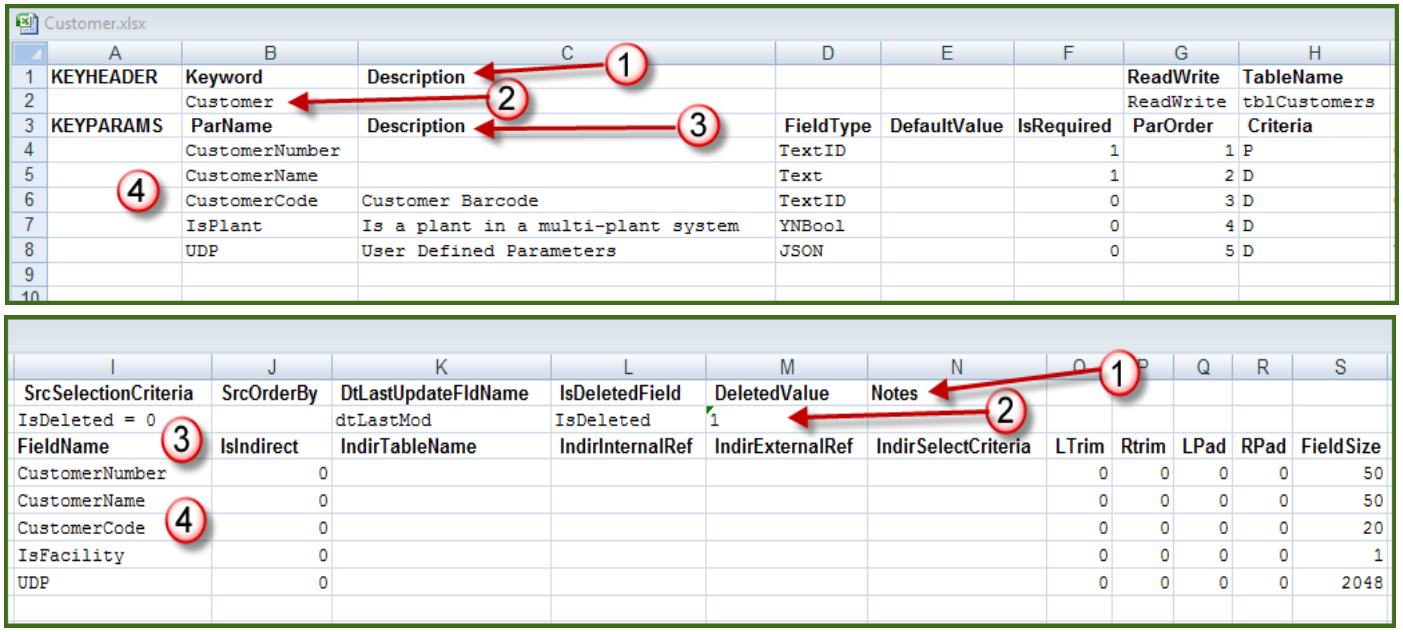
```
{"CustomerNumber":"ABC123", "CustomerName":"ABC Company"}
```

As these are all strings, they are applicable to a wide range of systems, which may use very different encoding schemes for their data.

A single set of interface code, such as for an interface to a SQL Database, can be used to read and write many different data objects, provided that the interface code knows how to translate between each HLDO and the fields in different database tables, including indirections and other complexities.

To this end, users of MilramX are able to define sets of metadata, which describe how to translate between table in a database and HLDOs. This Metadata can be imported into MilramX in the form of Excel spreadsheets and then translated into an XML metadata file that is used by the Adaptor code to translate between objects in the database and their corresponding HLDO representation.

Here is an example of such an Excel spreadsheet is shown below (in two sections: columns A-H and columns I -S):



Customer.xlsx

	A	B	C	D	E	F	G	H
1	KEYHEADER	Keyword	Description				ReadWrite	TableName
2		Customer					ReadWrite	tblCustomers
3	KEYPARAMS	ParName	Description	FieldType	DefaultValue	IsRequired	ParOrder	Criteria
4		CustomerNumber		TextID		1	1	P
5		CustomerName		Text		1	2	D
6		CustomerCode	Customer Barcode	TextID		0	3	D
7		IsPlant	Is a plant in a multi-plant system	YNBool		0	4	D
8		UDP	User Defined Parameters	JSON		0	5	D
9								
10								

I	J	K	L	M	N	O	P	Q	R	S
SrcSelectionCriteria	SrcOrderBy	DtLastUpdateFldName	IsDeletedField	DeletedValue	Notes					
IsDeleted = 0		dtLastMod	IsDeleted	1						
FieldName	IsIndirect	IndirTableName	IndirInternalRef	IndirExternalRef	IndirSelectCriteria	LTrim	Rtrim	LPad	RPad	FieldSize
CustomerNumber	0					0	0	0	0	50
CustomerName	0					0	0	0	0	50
CustomerCode	0					0	0	0	0	20
IsFacility	0					0	0	0	0	1
UDP	0					0	0	0	0	2048

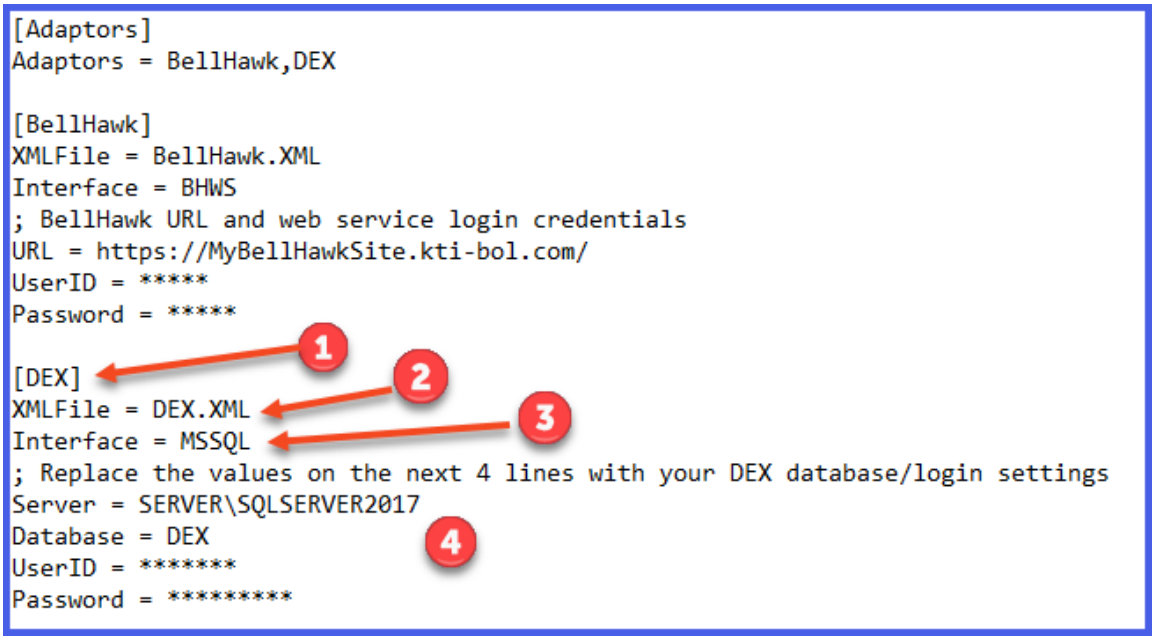
The spreadsheet consists of (from top to bottom):

1. A `KEYHEADER` row (1), which contains headings for Keyword related items
2. A row containing the Keyword (2) related items
3. A `KEYPARAMS` row (3), which contains headings for Parameter related items

4. Rows (4) containing data related to parameters.

Details of the format of this spreadsheet are explained in a separate MilramX user manual, as are the differences when the spreadsheet is used to define the relationship of each HLDO to a web-services interface.

Groups of DTOs are integrated into Transfer Processes which share a common set of adaptor definitions through the initialization file read by the Transfer Process when it first starts running.



This initialization file relates the name of each adaptor (1) to the metadata file name (2), containing the HLDO definitions and how they relate to the data source, and the Interface (3) to be used to read and write the data. This is then followed by the information (4) about how to access the database or web-services interface being used by the adaptor.

This enables a DTO to use simple subroutine calls to:

1. Fetch the latest updates for a named (Keyword) HLDO from a named adaptor.
2. Get the next HLDO from the set of updates just fetched
3. Lookup related HLDO data using a named adaptor
4. Assign data to one or more target HLDOs
5. Store the target HLDOs using a named adaptor.

This isolates the DTO from any of the details of the source or target systems and even allows changing an adaptor, such as from a local database adaptor to a remote web-services interface without modifying the DTO code.

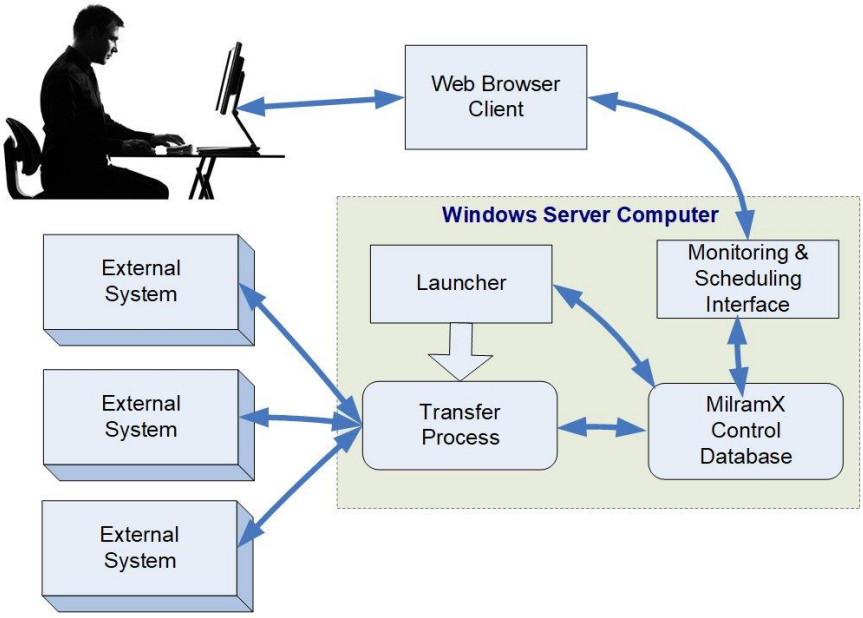
Within the DTO, the assignment of parameter values can directly reference the HLDO parameter name thus abstracting the actual database or web-services interface field names (which can be incomprehensible to a business analyst reading the DTO code).

```
BH_Customer("CustomerNumber") = DEX_Customer("CustomerNumber")

BH_Customer("CustomerName") = DEX_Customer("CustomerName")
```

This does not preclude the full use of the .Net programming environment to manipulate the HLDO values within the DTO.

Transfer Processes



DTOs are grouped into Transfer Processes which contain DTOs dedicated to a common purpose, such as the DTOs in the BHDEX Transfer process which exchange data between a local BHDEX database and the database of a remote BellHawk system. These transfer processes contain the necessary interfaces and are scheduled and run by a Launcher process.

The Launcher process takes its DTO scheduling information, which may be to run a single DTO or a chain of DTOs, periodically or at a specific time, from the MilramX control database. The Launcher then, at the appropriate time, launches the Transfer Process containing the DTO, with the DTO name or name of a string of DTOs to run, as its argument. The Transfer Process then carries out the designated transfer.

The reason for using a separate Launcher process is that a Transfer Process can become hung if there is a problem with the Interface or simply consumes a large quantity of resources when running that need to be returned to the operating system. When it finishes running the Transfer Process kills itself thereby returning all of the resources it used to the operating system. The Launcher also monitors the Transfer Process and can kill this process if it runs for longer than a specified time.

This enables MilramX to keep scheduling DTOs 24x7 even though a DTO or its adaptor may hang or not perform correctly.

The scheduling data for the DTOs is setup in the MilramX control database using a web-browser interface. This same interface can be used to monitor transfers remotely.

MilramX can also send Email or Text message alerts to a designated IT person if there are problems with the data transfers.

Software Development Environment

Development can be performed on a Windows Server or Windows Workstation computer using the Community Version of the Visual Studio IDE and a SQL Server Express database.

DTOs can be developed in VB.Net or C#.net using the Visual Studio IDE. All DTOs provided by KnarrTek, such as the BHDEX DTOs, are written in VB.Net as this is more familiar to industrial engineers than C#.net.

The MilramX SDK (Software Development Kit) consists of:

- A MilramX website to be installed using IIS
- A backup copy of the control database
- A Launcher program. Typically for development this is run manually as a user program.
- A set of DLLs (Dynamic Link Libraries) to be linked with the DTO code to create a transfer process.
- A Visual Studio project for a sample Transfer Function

If the BHDEX version of MilramX is ordered then a pre-built transfer function and a backup of the BHDEX database, along with the backup of a preloaded Control database are supplied.

Optionally, the BHDEX code development project, with the source code for the BHDEX DTOs is available for use as a starting point for DTO development.

Commentary

MilramX is still evolving with the upcoming addition of a Management Console to enable managers to be able to view and manage operations and inventory across multiple warehouses and manufacturing plants.

The core scheduling algorithms in MilramX are also being tuned to take advantage of the new multi-thread processors, such as the 64 core processors from Intel, now becoming available. Also, MilramX is structured so that the processing can be split over multiple CPUs as well as networks of distributed computers. This will enable hundreds or thousands of DTOs to be run in parallel, if needed.

