DLMS: Ten Years of AI for Vehicle Assembly Process Planning

Nestor Rychtyckyj

Ford Motor Company
Manufacturing Quality and Business Systems, Vehicle Operations General Office
Room B154, PO Box 1586, Dearborn, MI 48121
email: nrychtyc@ford.com

Abstract

Since its presentation at the inaugural 1989 IAAI Conference (O'Brien et al. 1989), Ford's Direct Labor Management System (DLMS) has evolved from a prototype being tested at a single assembly plant to a fully-deployed application that is being utilized at Ford's assembly plants throughout the world. DLMS is Ford's automated solution to managing the automobile manufacturing process system at our vehicle assembly plants. This paper will describe our experiences and the lessons that have been learned in building and adapting an AI system to the rapidly-evolving world of automotive vehicle assembly process planning. We will cover issues such as knowledge base development and maintenance, knowledge representation, porting the system to different platforms and keeping the system viable and up-to-date through various organizational and business practice changes. We will also discuss how DLMS has become an integral part of Ford's assembly process planning business.

Problem Description

For a manufacturing company like Ford Motor Company, the assembly process planning activity is the critical link between the development and design of a product and it's final assembly and delivery to the customer. In a typical year Ford manufactures and sells over six and a half million cars and trucks all over the world. A major initiative was undertaken at Ford Vehicle Operations to improve the quality and effectiveness of the assembly process planning activity. The central theme to this project is the development of a knowledge-based system that will support the creation and manipulation of planning data in all stages of the assembly planning process, from the central office budgeting and cost estimating down to the work allocation and line balancing at the plant floor level. The result of this initiative is the Direct Labor Management System.

The development of the Direct Labor Management System (DLMS) began in 1989 at Ford's Body & Assembly Division. DLMS was designed to be an integral part of the Manufacturing Process Planning System that was being developed for the assembly plants in North America. The objectives of DLMS included standardizing the process sheet writing, creating work allocation sheets for the plant floor and estimating labor time accurately. The process sheet is the primary vehicle for conveying vehicle assembly information from the central engineering functions to the assembly plants. It contains specific information about work instructions and describes the parts and tools required for the build process. The work that is required to build the vehicle according to the process sheet instructions must then be allocated among the available personnel. Work allocation requires a precise means of measuring the labor time that is needed for any particular task.

The requirement for a system to automate process planning in automobile assembly at Ford Motor Company was very evident since the early 1980's. Previously, process sheets were written in free-form English and then sent to the assembly plants for implementation. The quality and correctness of process sheets differed greatly based upon which engineer had written a particular sheet. There was no standardization between process sheets. Industrial engineers at the assembly plants would be forced to implement work instructions based on various styles of process sheets. The process sheets could not describe the amount of labor required and the assembly plants were not able to accurately plan for labor requirements. Work usage instructions were written manually and the time required to accomplish a particular job would have to be measured manually. These manual "stopwatch" time studies suffered from several major disadvantages. A time study consisted of an industrial engineer watching an assembly line worker doing their job and measuring how long each job would take. These measurements would vary from worker to worker, so that multiple time studies were required for each particular job. Since there may be hundreds or even thousands of jobs in an assembly plant, the time studies were very expensive and time-consuming. Time studies also have a very adverse effect on worker morale and are a source of resentment among the assembly personnel. Since labor is a very significant portion of the cost of producing an automobile, there was a very strong incentive to develop a system that could both standardize the process sheet and create a tool for automatically generating work instructions and times from these process sheets. The first attempts to create DLMS were done utilizing standard third generation programming languages (COBOL) and existing IBM mainframe databases (IMS). The sheer complexity of the knowledge required to accurately generate reliable work instructions could not be represented in either a database or in a program. A database could easily store the amount of data required, but the relationships between the various components in the database could not be adequately represented. A program could be written that could explicitly list all of the inputs and desired outputs, but this program would quickly become obsolete and be impossible to maintain.

The solution to this problem was to develop a knowledge-based system that utilizes a semantic network knowledge representation scheme. DLMS utilizes techniques from natural language processing, description logics and classification-based reasoning to generate detailed plant floor assembly instructions from high-level process descriptions. This system also provides detailed estimates of the labor content that is required from these process descriptions. Techniques such as machine translation and evolutionary computation are being integrated into DLMS to support knowledge base maintenance and to deploy DLMS to Ford’s assembly plants that do not use English as their main language.

The DLMS application remains a viable and integral part of vehicle assembly process planning. The writing of process sheets has been standardized through the use of Standard Language. The output of the DLMS system consists of work allocation instructions along with their associated MODAPTS codes that are converted into labor time for each operation. The following sections will describe the DLMS system in more detail and discuss the development and use of Standard Language. Other issues associated with AI systems development, such as knowledge base maintenance and the integration of DLMS with external databases and systems, will also be covered.

Application Description

The Direct Labor Management System (DLMS) is an implemented system utilized by Ford Motor Company's Vehicle Operations division to manage the use of labor on the assembly lines throughout Ford's vehicle assembly plants. DLMS was designed to improve the assembly process planning activity at Ford by achieving standardization within the vehicle process build description and to provide a tool for accurately estimating the labor time required to perform the actual vehicle assembly. In addition, DLMS provides the framework for allocating the required work among various operators at the plant and builds a foundation for automated machine translation of the process descriptions into foreign languages.

The standard process planning document known as a process sheet is the primary vehicle for conveying the assembly information from the initial process planning activity to the assembly plant. A process sheet contains the detailed instructions needed to build a portion of a vehicle. A single vehicle may require thousands of process sheets to describe its assembly. The process sheet is written by an engineer utilizing a restricted subset of English known as SLANG (Standard LANGuage). Standard Language allows an engineer to write clear and concise assembly instructions that are machine readable.

Figure 1 shows a portion of a process sheet written in Standard Language. This process sheet is written by an engineer at the Vehicle Operations General Office; it is then sent to the DLMS system to be "validated" before it can be released to the assembly plants. Validation includes the following: checking the process sheet for errors, generating the sequence of steps that a worker at the assembly plant must perform in order to accomplish this task and calculating the length of time that this task will require. The DLMS system interprets these instructions and generates a list of detailed actions that are required to implement these instructions at the assembly plant level. These work instructions, known as “allocatable elements”, are associated with MODAPTS (MODular Arrangement of Predetermined Time Standards) codes that are used to calculate the time required to perform these actions.

MODAPTS codes are widely utilized as a means of measuring the body movements that are required to perform a physical action and have been accepted as a valid work measurement system. (IES 1988). For example, the MODAPTS code for moving a small object with only a hand is M2; utilizing the arm gives a code of M3. The MODAPTS codes are then combined to describe a entire sequence of actions. MODAPTS codes are then converted into an equivalent time required to perform that action. Figure 2 shows the output generated by the DLMS system including a description of each action with its associated MODAPTS code.

The allocatable elements generated by DLMS are used by engineering personnel at the assembly plant to allocate the required work among the available personnel. DLMS is a powerful tool because it provides timely information about the amount of direct labor that is required to assemble each vehicle, as well as pointing out inefficiencies in the assembly process.

The DLMS system consists of five main subsystems:

Process Sheet Written in Standard Language

TITLE: ASSEMBLE IMMERSION HEATER TO ENGINE
10 OBTAIN ENGINE BLOCK HEATER ASSEMBLY FROM STOCK
20 LOOSEN HEATER ASSEMBLY TURNSCREW USING POWER TOOL
30 APPLY GREASE TO RUBBER O-RING AND CORE OPENING
40 INSERT HEATER ASSEMBLY INTO RIGHT REAR CORE PLUG HOSE
50 ALIGN SCREW HEAD TO TOP OF HEATER
TOOL 20 1 P AAPTCA TSEQ RT ANGLE NUTRUNNER
TOOL 30 1 C COMM TSEQ GREASE BRUSH

Figure 1.

Resulting Work Instructions Generated by DLMS For Line 20

LOOSEN HEATER ASSEMBLY TURNSCREW USING POWER TOOL
GRASP POWER TOOL (RT ANGLE NUTRUNNER) <01M4G1>
POSITION POWER TOOL (RT ANGLE NUTRUNNER) <01M4P2>
ACTIVATE POWER TOOL (RT ANGLE NUTRUNNER) <01M1P0>
REMOVE POWER TOOL (RT ANGLE NUTRUNNER) <01M4P0>
RELEASE POWER TOOL (RT ANGLE NUTRUNNER) <01M4P0>

Figure 2.

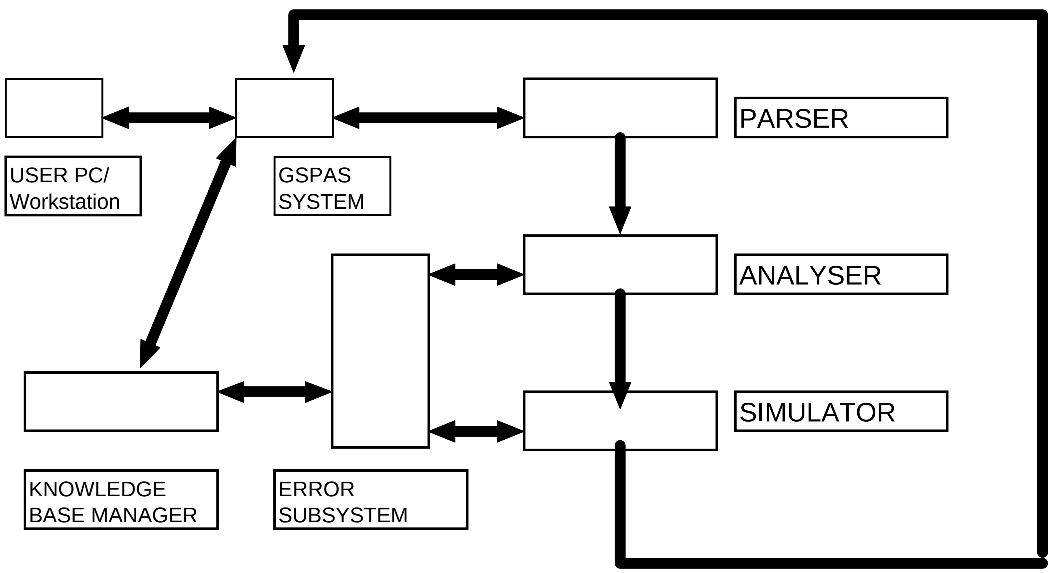

Figure 3: DLMS System Architecture

Adverb	**Verb**	**Noun Phrase**	**Initial Location**	**Intermed Location**	**Final Location**	**Faste-ner**	**Tool**
	Obtain	Fuel Filler Door	From Vehicle				
Auto	Load	Sub-Assembly			To Station		
	Insert	Bolt	Into slot				Using Tool

	Verify	That Bracket is in Place					

Figure 4. Template Describing A Standard Language Sentence

parser, analyzer, simulator, knowledge base manager and the error checker. The input into DLMS is a process sheet; it is initially parsed to break down the sentence into its lexical components which includes the verb, subject, modifiers, prepositional phrases and other parts of speech. Since Standard Language is a restricted subset of English, the parser has a very high rate of success in properly parsing the input from the process sheets. The parser utilizes the Augmented Transition Network (ATN) method of parsing (Charniak 1987). Any process element that is not parsed successfully will then be flagged by one of the error rules that will (hopefully) suggest to the user how to correct this element. The analyzer will then use the components of the parsed element to search the knowledge base (or taxonomy) for relevant information describing that item. For example, if the input element contained the term "HAMMER", the taxonomy will be searched for the term “HAMMER”. When it is found the system will learn all of the attributes that "HAMMER" has: (it is a Tool, its size is medium, it can be used with one hand, etc.) The system performs this analysis on all of the components of the input element in order to select what work instructions are required. The work instructions are then found in the taxonomy based on all of the available input and are passed on to the simulator. The simulator uses the information found in the taxonomy to generate the allocatable elements and MODAPTS codes that describe the input element. These work instructions are then sent to the user. The knowledge base manager is used to maintain the knowledge base; this maintenance may be performed by the user community or by the system developers.

All of the associated knowledge about Standard Language, tools, parts and everything else associated with the automobile assembly process is contained in the DLMS knowledge base or taxonomy. This knowledge base structure is derived from the KL-ONE family of semantic network structures and is the integral component in the success of DLMS. DLMS also contains a rulebase of over 350 rules that are used to drive the validation process and perform error-checking on the Standard Language input. Figure 3 displays the DLMS system architecture.

The organization of the knowledge base is based on the KL-ONE model. The root of the semantic network is a concept known as THING which encompasses everything within the DLMS world. The children of the root concept describe various major classes of knowledge and include such things as TOOLS, PARTS and OPERATIONS. Each concept contains attributes or slots that describe that object. The values of these attributes are inherited from the concept's parents. Ranges of valid values can be given for any particular attribute. Any attempt to put an invalid value in that attribute will trigger an error. All of the information dealing with the organization and structure of the taxonomy is also contained in the taxonomy itself. There are four types of links that describe the relationship between any two concepts: subsumes, specializes, immediately-subsumes and immediately-specializes. The subsumption relation describes a link between a parent concept and all of its children, including descendants of its children. The "immediately-subsumes" relation describes only the concepts that are direct children of the parent concept. The "specializes" and "immediately specializes" relations are inverses of the subsumption relation. A concept "immediately specializes" its direct parent concepts and "specializes" all of the concepts that are ancestors of its parents. These relationships are stored as attributes of any given concept and can be utilized as a tool to trace any concept through the entire taxonomy.

Uses of AI Technology

The DLMS system utilizes several different AI techniques including Description Logics, Rule-based Processing and Machine Translation. The heart of the DLMS system is the knowledge base that utilizes a semantic network model to represent all of the automobile assembly planning information. The use of a semantic network as part of knowledge representation system is also known as *Description Logics.* A Description Logic implementation known as CLASSIC has been successfully used at AT&T to develop telecommunication equipment configurators (McGuiness and Patel-Schneider 1998). Semantic networks have also been integrated into Object-Oriented Analysis (OOA) (Mylopoulos 1999). The goal of Object-Oriented Analysis is to combine ideas from object-oriented programming with semantic network modeling and knowledge representation into a powerful modeling framework. The DLMS implementation of Description Logic is based on the KL-ONE knowledge representation language.

The KL-ONE knowledge representation system (Brachman 1985) was first developed at Bolt, Baranek and Newman in the late 1970's as an outgrowth of semantic net formalisms. KL-ONE was selected for use on the DLMS project because of its adaptability for many diverse applications as well as the power of the KL-ONE classification algorithm. KL-ONE is derived from research done on semantic networks. The principal unit of information is the "concept". Each concept has a set of components or attributes that is true for each member concepts being closer to the root of the tree and the more specific concepts being the leaves of the tree. A concept in a KL-ONE knowledge base inherits attributes from the nodes that subsume it. The power of the KL-ONE system lies in the classification scheme. The system will place a new concept into its appropriate place in the taxonomy by utilizing the subsumption relation on the concept's attributes (Rychtyckyj 1994).

A strictly rule-base approach was also considered, but the complexity and future maintainability of a system containing explicit knowledge about a dynamic domain such as automobile assembly ruled this approach out. This maintainability issue was illustrated in the development of the R1 (also known as XCON) system that was utilized at Digital Equipment Corporation to assist in the configuration of DEC VAX computer systems. This equally dynamic environment showed that in any given year more than 50% of the rules were modified (Sowa 1987). Rules were later added to DLMS as part of the error checker and to control the execution of the system as this type of knowledge changes much less frequently.

A requirement for the DLMS knowledge base included the ability to make frequent and complex changes without affecting other components of the knowledge base. This required that the objects in the taxonomy be stored in classes that were analogous to the real world of automobile assembly planning. This approach led to a semantic network representation of the automobile assembly world where classes and subclasses corresponded to their appropriate equivalents in the real world. This type of semantic network representation was very similar to the KL-ONE representation language. It was decided to model the Ford automobile manufacturing knowledge base utilizing KL-ONE in order test the feasibility of this approach. This prototype proved very successful and the basic KL-ONE model proved to be both robust and flexible as the knowledge base evolved over the years. Changes were made for processing and memory efficiency (i.e. the use of a hash table to store the list of concepts), but the KL-ONE logical design has been successful in terms of our problem domain.

Since its implementation in 1990 the DLMS knowledge base has been frequently modified to keep pace with the rapidly changing automobile and truck assembly process. These changes have included the implementation of DLMS for plants outside of North of the set denoted by that concept. The main form of relation between concepts is called "subsumption". Subsumption is the property by which concept A subsumes concept B if, and only if, the set denoted by concept A includes the set denoted by concept B. The KL-ONE knowledge base as used in DLMS can be described as a network of concepts with the general America, the assembly of entirely new types of vehicles including electric and alternative fuel vehicles and the improvements in the actual assembly process. Since the complexity of the knowledge base has increased, we have become concerned about the efficiency of the subsumption and classification algorithms. One possible solution to this is to develop a tool that will assist in the re-engineering of the network to reduce complexity and increase efficiency. Currently we are utilizing an Evolutionary Computational technique, known as Cultural Algorithms (Reynolds and Chung 1996), to re-engineer the semantic network with the goal of reducing the network complexity in two ways (Rychtyckyj and Reynolds 1998). The first approach is to reduce the number of attributes that have to be compared during the classification algorithm. The second approach is to begin the classification search at a node that is lower in the network to reduce the number of nodes that have to be visited.

We are also utilizing Machine Translation software for the translation of our system output into the home languages of the countries where our plants are located in. For this approach we are working with Systran Software Inc. to customize their software to work with Standard Language and our application. One part of this involves building a lexicon of the specific Ford terminology and the required translations in each target language. Another requirement of the project is to modify the Systran software to accept the sentence structure that is utilized within Standard Language.

Application Use and Payoff

As mentioned previously, DLMS has been in use continuously within Ford Motor Company since 1990. Currently the system supports hundreds of users that are located in various Ford locations around the world including North and South America, Europe and Asia. The users are primarily engineers that are writing processes that will be used to build vehicles at Ford assembly plants. Another group of users are the engineers that are located at these assembly plants. A third group of system users are located at the central office locations and are concerned with financial planning for the labor authorizations for each assembly plant.

One of the most important tangible benefits from the utilization of DLMS has been the acceptance and use of Standard Language to describe assembly instructions throughout our engineering community. This has dramatically increased productivity by reducing ambiguity and confusion between our General Office engineers and the people at our assembly plants. Standard LANGuage (SLANG) was developed as a controlled language that would have the flexibility to describe all of the instructions required for the vehicle assembly process. SLANG needed to be both unambiguous and precise enough so that each sentence could only generate one unique set of work allocation instructions. A controlled language, such as SLANG, also needs some type of mechanism that will check the written text so ensure that it complies with the rules of that language. In DLMS this is accomplished through a parser that checks each sentence for correctness and compliance to Standard Language guidelines.

Standard Language requires that each sentence conform to a structure that can be read by the system and provides sufficient information for the AI system to generate the correct work allocation instructions for that command. In addition, each word must be found in the DLMS knowledge base and be used correctly within the Standard Language sentence. The rules for writing Standard Language can be best described by defining a template that the written text must follow. Figure 4 shows a template with examples of typical Standard Language sentences. The verb and noun phrase, which usually describes a part, are required; the other parts of the sentence are optional and used as required. The last sentence in Figure 4 ("Verify that bracket is in Place") is a special type of inspection element that can utilize a much more free-form text syntax than a regular Standard Language expression. Each sentence in Standard Language must contain a verb that describes the main action performed by the assembly operator. The Standard Language verbs have been defined by the engineering community to represent a single precise action and are not interchangeable. For example, in Standard Language the verbs "SECURE" and "TIGHTEN" describe different operations and cannot be used interchangeably. All of the lexical terms within Standard Language are defined within the DLMS knowledge base.

Standard Language provides significant benefits in reducing the ambiguity and inconsistency that is present within free-form written text. Learning to write in Standard Language incurs training costs and a learning curve for the process writers, but the benefits of Standard Language more than make up for this. DLMS has provided other benefits to Ford Motor Company, such as automatic generation of work instructions with associated times, accurate estimates of direct and indirect labor times and the ability to plan for mix/volume changes and line balancing. The work instructions that are created by DLMS are utilized by the engineers at the assembly process to allocate the work among the available personnel at the plant. The system provides a method to determine how much work each person is required to perform. This will identify any potential overwork or under-utilization for any particular operator and provide an opportunity to re-allocate the work. DLMS also distinguishes between direct and indirect labor. Direct labor describes work that is directly related to vehicle assembly. Other actions, such as walking to obtain parts or tools, are described as indirect labor. The plant engineers use DLMS to identify areas with a high proportion of indirect labor and modify that particular work area to reduce the indirect labor.

Plant engineers can also utilize the output from DLMS to simulate different scenarios on how the mix and volume of a particular vehicle can be assembled. This is extremely useful when a new vehicle is being launched at an assembly plant. The amount of labor that is needed to build the vehicle can be calculated to produce accurate manpower estimates and tool requirements. DLMS is a tool that gives Ford the ability to manage the assembly process planning work from development to production.

Application Development and Deployment

The original DLMS prototype was developed at Ford in conjunction with Inference Corporation in 1989. It was piloted at one of our assembly plants and included knowledge about one phase of the assembly process. After this approach was validated, the DLMS system was expanded to include the entire assembly process and was deployed at other North American assembly plants. During this time period our emphasis was on building up the knowledge base and making those modifications that were necessary for the system to be accepted by our plants. This knowledge transfer consisted of working closely with assembly engineers, making periodic visits to the assembly plants and conducting monthly video conferences with the users of the system.

DLMS was originally developed using Common Lisp and the Automated Reasoning Tool (ART) from Inference Corporation. LISP is an extremely powerful symbolic programming language that includes facilities for garbage collection, symbol manipulation, rapid prototyping and object-oriented programming. ART is a LISP-based expert system shell that utilizes a forward-chaining inference engine to perform pattern-matching and rule firing. DLMS was initially deployed on the

Texas Instruments Explorer platform which was a stand-alone Lisp machine that included the UNIX operating system. Communications between DLMS and the mainframe IMS database was handled using a screen emulator interface. After the initial DLMS deployment, the TI Explorer platform was discontinued and both support and maintenance for these machines became problematic. In order to ensure the future viability of DLMS the system was ported to the Hewlett Packard UNIX platform. The communications interface was rewritten utilizing the Brixton communications software through an interface with LISP. The DLMS development team usually consisted of no more than three developers at any time.

Following a major re-organization that put all the assembly plants around the world into one organization, it was decided replace the legacy mainframe Manufacturing Process Planning System(MPPS) with a new client-server application. The new Global Study Process Allocation System (GSPAS) utilizes a distributed Oracle database that needs to be accessed from DLMS. The ART software tool was not being upgraded and would not function with the latest versions of Oracle. This necessitated the conversion of DLMS from ART to another expert system shell that would preserve the functionality of the system and provide a platform for interfacing with the database. We selected the LispWorks/KnowledgeWorks tool from Harlequin Inc. due to its similarity to ART and continued support. The graphical Knowledge Base Manager facility was ported from ART to the LispWorks Common Lisp Interface Manager (CLIM) software which allows for the development of a graphical user interface from LISP. The ART rulebase was rewritten into KnowledgeWorks and DLMS was successfully deployed as part of the GSPAS system.

With the expansion of DLMS to our European assembly plants our focus shifted on expanding our knowledge base to model the assembly process in our European plants. This included modifying Standard Language and adding additional tools and parts for different vehicles. This change also produced a requirement that we translate the DLMS output into other languages.

Currently there are two versions of DLMS being utilized within Ford Vehicle Operations. The MPPS version of DLMS still utilizes ART and LISP while the GSPAS version utilizes KnowledgeWorks and LISP. Both versions run on the HP UNIX platform and share a common knowledge base. Communication to the GSPAS system is handled through the RPC protocol and the Oracle database is accessed directly from the LISP code in DLMS.

Maintenance

As mentioned previously, the DLMS taxonomy or knowledge base contains all of the relevant information that describes the vehicle assembly process at Ford Motor Company. This includes all of the lexical classes included in Standard Language such as verbs, nouns, prepositions, conjunctions and other parts of speech, various tools and parts utilized at the assembly plants, and descriptions of operations that are performed to build the vehicle. Currently the DLMS taxonomy contains over 9000 such concepts.

The DLMS Knowledge Base is maintained through the use of two different tools: the Knowledge Base Manager (KBM) and the Knowledge Base Update facility (KBU). The Knowledge Base Manager is a graphical tool that is used by the system developers to make important changes to the knowledge base that will affect the actual output generated by the system. Since this output will have a major impact on the assembly process any such change must be approved by a committee representing all of the interested parties. All changes made to the knowledge base are logged by the system to keep a record of the system's modification history.

The Knowledge Base Update (KBU) is an automated update facility that was used by system users to make minor modifications to the knowledge base. A minor modification is a change that will not impact the output produced by the system. Examples of minor modifications include the addition of new words into the taxonomy. The KBU facility allowed users to incorporate these changes directly into the taxonomy without any kind of system developer intervention. All changes made through the KBU facility were also logged for future reference.

With the requirement that the Standard Language output be translated from English into the home languages of our assembly plants we discovered that many errors had been introduced into our system through the KBU. These consisted of simple grammatical errors such as misspellings or giving a word the incorrect part of speech, but they created serious difficulties for our translation software. Other non-technical terms that described tools or equipment at one assembly plant were not known to workers in other countries and also adversely impacted our translation. These problems have forced us to remove the Knowledge Base Utility from production and force all additions and modifications to the knowledge base to be approved by the user committee and then sent to the developers.

There have been several utilities developed that are used to validate the knowledge base and prevent any errors from being inadvertently introduced into the system. These include an automated facility that scans through the knowledge base and creates sample test cases that cover various operations within the system. These test cases are then executed and compared against a previous baseline to determine if the results have changed. There is also a suite of test cases that are manually updated to cover problems and modifications that have occurred in the system. The regression tests are also run against a similar baseline of expected results. We are also utilizing Evolutionary Computation to develop a method of automatically re-engineering the knowledge base to reduce complexity and improve efficiency of our classification algorithms.

Conclusions

DLMS has proven to be a successful implementation of AI technology that has delivered tangible benefits to Ford Motor Company. These include the following: standard and accurate process sheets through the use of Standard Language, automatic generation of work instructions with associated times, accurate estimates of direct vs. indirect labor times and the ability to plan for mix/volume changes and line balancing. Through the 10 years of work on DLMS we have validated the use of AI as a viable technology in a dynamic business environment. The use of Description Logics to model our assembly process planning environment has paid off with its flexibility and expressiveness. DLMS has provided Ford with a competitive advantage and has justified the use of AI as a tool for building and delivering systems that solve difficult business problems.

References

Brachman, R., Schmolze, J. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171-216.

Charniak, E., Riesbeck, C., McDermott, D., Meehan, J., 1987. *Artificial Intelligence Programming*, Lawrence Erlbaum Associates.

Industrial Engineering Services. 1988. *Modapts Study Notes for Certified Practitioner Training.*

McGuiness, D. and Patel-Schneider, P. 1998. Usability Issues in Knowledge Representations Systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 608-614. Menlo Park, CA: AAAI Press.

Mylopoulos, J., Chung, L., Yu, E. 1999. From Object-Oriented to Goal-Oriented Requirements Analysis. *Communications of the ACM* 42(1): 31-37.

O'Brien, J., Brice, H., Hatfield, S., Johnson, W., Woodhead, R. 1989. The Ford Motor Company Direct Labor Management System. In *Innovative Applications of Artificial Intelligence*, 331-346. MIT Press.

Reynolds, R.G. and Chung, C. 1996. A Self-adaptive Approach to Representation Shifts in Cultural Algorithms. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computing*, 94-99. Nagoya Japan: IEEE Press.

Rychtyckyj, N. 1994. Classification in DLMS Utilizing a KL-ONE Representation Language. In *Proceedings of the Sixth International Conference on Tools with Artificial Intelligence*, 339-345. IEEE Computer Science Press.

Rychtyckyj, N. 1996. DLMS: An Evaluation of KL-ONE in the Automobile Industry. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning*, 588-596. Morgan Kaufmann Publishers.

Rychtyckyj, N. and Reynolds, R. G. 1998. Learning to Re-Engineer Semantic Networks Using Cultural Algorithms. In *Evolutionary Programming VII*, 181-190. Springer-Verlag.

Soloway, E., Bechant, J., Jensen, K. 1987. Assessing the Maintainability of XCON in RIME: Coping with the Problems of a Very Large Rule-Base. In *Validating and Verifying Knowledge-Based Systems*, 294-299. IEEE Press.