AI Magazine Volume 26 Number 3 (2005) (© AAAI)

The General Motors Variation-Reduction Adviser

Deployment Issues for an AI Application

Alexander P. Morgan, John A. Cafeo, Kurt Godden, Ronald M. Lesperance, Andrea M. Simon, Deborah L. McGuinness, and James L. Benedict

■ The General Motors Variation-Reduction Adviser is a knowledge system built on case-based reasoning principles that is currently in use in eighteen General Motors asssembly centers. This article reviews the overall characteristics of the system and then focuses on various AI elements critical to support its deployment to a production system. A key AI enabler is ontology-guided search using domain-specific ontologies.

The Variation-Reduction Adviser (VRA) is a knowledge system for automotive assembly plants whose goal is to support quality improvement activities for assembly line processes (Cafeo et al. 2001, Morgan et al. 2001). The primary use of VRA is to improve communication in the plants and between plants to assist with problem solving necessary to keep the line producing the highest quality products. Our original prototype was tested by a "dimensional management" team working on "variation reduction" problems in a plant. Currently, other teams including "paint," "maintenance," and "general assembly" are testing it at various plant locations, so its range of application includes a whole cluster of related assembly-plant domains. While its original name reflected the specific focus on dimensional analysis for variation reduction, we have kept this name and broadened its interpretation following the principle of *kaizen* (kaizen is Japanese for "continuous improvement." It is fundamental to many aspects of Japanese quality management), that all improvements in process can be viewed as "variation reduction."

VRA was originally conceived as a case-based reasoning (CBR) system (Leake 1996) and retains case-based features. Its failure as a pure CBR system for social reasons is one of the interesting aspects of this application. That this failure induced us to introduce an ontology-guided search (OGS) functionality not originally planned is another interesting aspect. In this article, we focus on the AI perspective of our business task, pointing out the problems being addressed, some challenges encountered in the field, our solution strategy, and an evaluation of the value added by the system.

There are, of course, recognized relationships between various aspects of knowledge management and AI. See, for example, the survey by Smith and Farquhar (2000). AI has been used for planning in manufacturing; for example, DLMS (Rychtyckyj 1999) and the Stamping Adviser (Leake et al. 1999). VRA is more closely related to problem-solving systems; for example, Ford's eBPR (Kwiecien et al. 2001), Schlumberger's Eureka/InTouch (McDermott, O'Dell, and Hubert 2000), and Xerox PARC's Eureka (Bobrow and Whalen 2002). All of these have common elements with VRA of best practices, peer-to-peer sharing, and diagnosis, as well as some commonality in their choice of AI tools. The primary differences include VRA's focus on manufacturing (including its community-of-practice-specific diagnostic ontologies) and the fact that its "best practice" functionality is peer moderated rather than "managed." We lack the space here to compare these various systems at length.

The VRA prototype is currently in production use in 18 GM assembly centers. The original system is in English, but there is a Spanish version in use in 2 Mexican plants, and a German version is being tested.

Task Description

VRA addresses two primary issues with respect to plant production. First, there are many people working on tasks across multiple shifts, and these people must communicate about their progress and problems. The second is that plants need to maintain a record of updates to equipment and work done. VRA fulfills both these needs, which are relatively short term in nature. The combination of these two types of entries, when viewed from a longer perspective, enables VRA to also function as a lessons-learned system for assembly plants, providing a "memory" for solutions and a repository from which best practices can be extracted.

Application Description

The VRA architecture (see figure 1) includes viewing and authoring subsystems, with a variety of domain-friendly features, support functions, a database of entries, and search and analysis functions. Also included are database and ontology maintenance functions. VRA is organized around entries. Each entry has some attribute values (entered from pull-down lists) and also a block of free text. Figure 2 shows an image of the opening screen of VRA, showing entries and some of the search functions visible on the left. Graphical attachments are optional, but useful. See the paper by Morgan et al. (2003) for additional details. The application is currently being converted from Microsoft Visual Basic/Access to a web-based version with more powerful database and search support. We have been assisted in this by members of the PARC scientific staff, who joined the project for the year 2003 to assist with in-plant social-technical analyses and with this web-based transition.

The key AI elements of VRA are its case-based features and the elements of the domain-specific ontologies and ontology-guided search. These are discussed further in the following two sections.

Uses of AI Technology: CBR

VRA was originally conceived as a classic feature-vector-based diagnostic CBR system. It was quickly realized, however, that a strict feature-based CBR model would not work because of the complexity of the problem-solving process. This issue is described in the next subsection. The system based on the model in the next subsection became version 0 (VRA-0) of our system, described later. In response to user feedback, we developed VRA-1, which is VRA-0 weakly linked to a communication log. After VRA-1 was further tested, a new version, VRA-2, was created. In the paper by Morgan et al. (2003), the evolution from VRA-0 to VRA-2 is detailed. Here we only sketch this evolution and note the main AI features.

A Case Structure for a Complex Diagnostic Environment

Consider a diagnostic environment in which, for each case, a small subset of a large set of symptoms can arise. Some of these symptoms are the results of tests. These tests are not performed in any fixed order, but at the discretion of the technicians. No particular subset of tests is always performed. A case consists of symptoms, results of tests, results of inspections (a kind of test), faults, repair actions (for example, replace a part), and outcomes. This formulation fits our domain and also that of the National Semiconductor case structure described in chapter three of Watson (2002).

A fixed-length feature vector cannot capture a case because: (1) the attribute values are naturally grouped (by symptom, test, and so on), so that there are repetitions of values for the same attribute that must be properly associated with each other; (2) cases do not have a natural fixed length; (3) there is a time sequence for these groupings of attribute values that has physical significance and that changes from case to case. A case needs to be a reasonable summary of what happened as symptoms, tests, inspections, and results occurred, a comprehensible record that a person can read and understand. Thus, for VRA-0, we devised the following structure: a case is defined to be a sequence of observations. Observations are classified into a finite number of types. Each type is represented by a templated sentence. These sentences capture symptoms, results of tests, actions, resolutions, and a few other types. A vector of attribute values represents each type of observation, where the values are the fillers of the slots of the templated sentence. Thus, a case is a sequence of observations of various types, and the types occur in no particular order, although they are taken from a finite set of

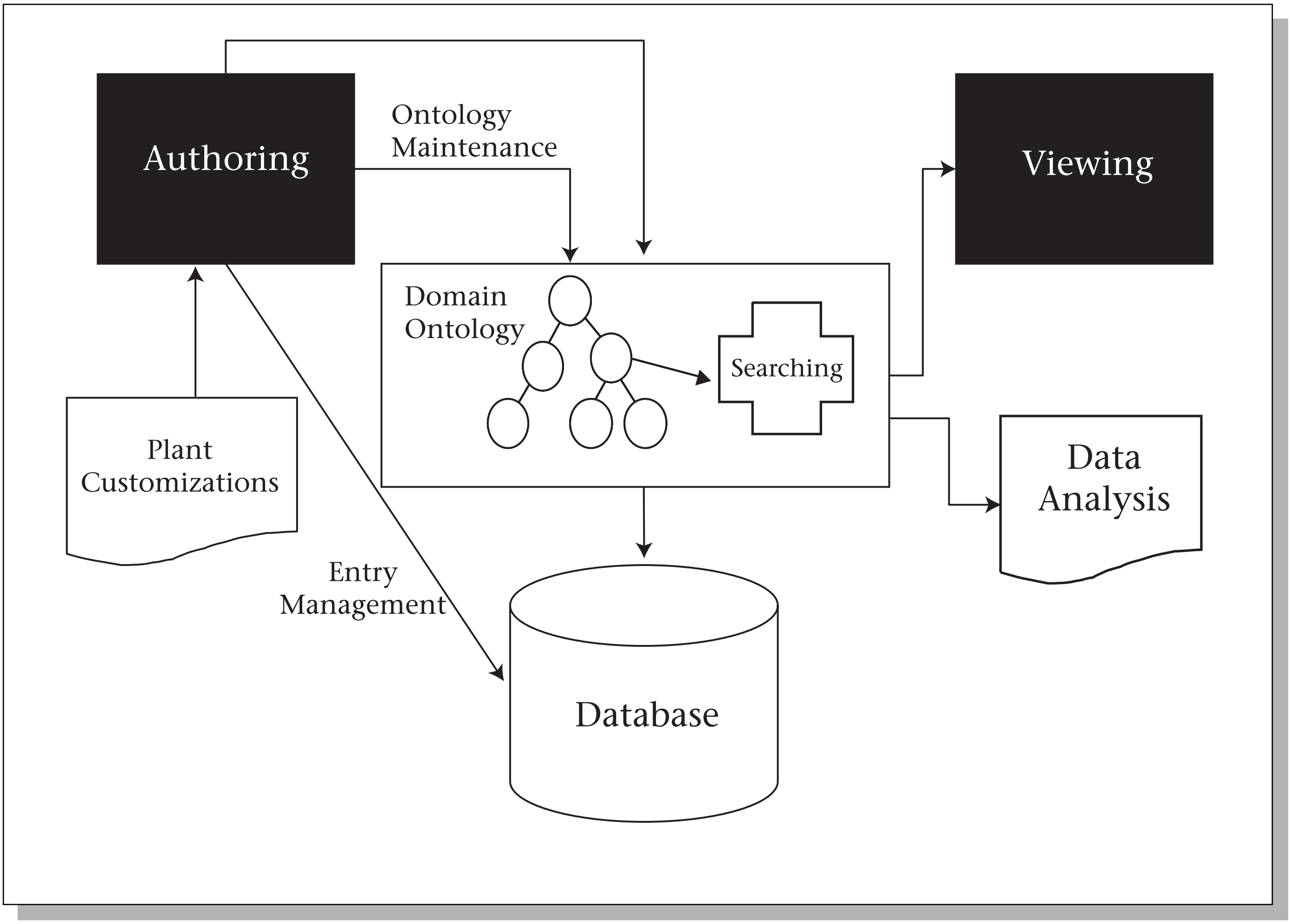


Figure 1. The VRA Achitecture.

types. This formal structure seems to be detailed enough to capture the important aspects of the diagnostic process, and it is vastly more structured than free text. Similarity between cases is built up from similarity between values of attributes and similarity between observations. See Morgan et al. (2003) for details.

A Knowledge Structure Emphasizing Communication Over Problem Solving

After using VRA-0, users asked us to provide them with a log environment, in which a few categorizing attributes might be selected through pull-down menus, and then the user would be free to enter content as a block of text. This log system began as a supplement to the CBR system (creating VRA-1). However, it quickly became a focal point of the implemented system. In fact, the log flourished while the CBR system languished. The users found significant benefit in having a distributed online mechanism for capturing semistructured input of daily events. They could not make much use of the CBR system for recovering previously solved problems, however, because they would not author enough cases to "boot strap" the archive.

The next version of VRA, VRA-2, thus focused on the communications and log aspects of the system. One way of describing this evolution is moving from the context of CBR with a focus on structural similarity to that of computer-supported cooperative work (CSCW) with a focus on communication (see Ackerman [2003]) and simple natural language input. This evolution could be viewed as a movement away from structured AI as a primary solution strategy. However, VRA-2 has seen a technology pull for a different kind of AI, in requests for a smart search capability. That is, the original desire to recover solutions to problems, when the problems reoccur, remains. Users want to be able to find previous cases similar to a cur-

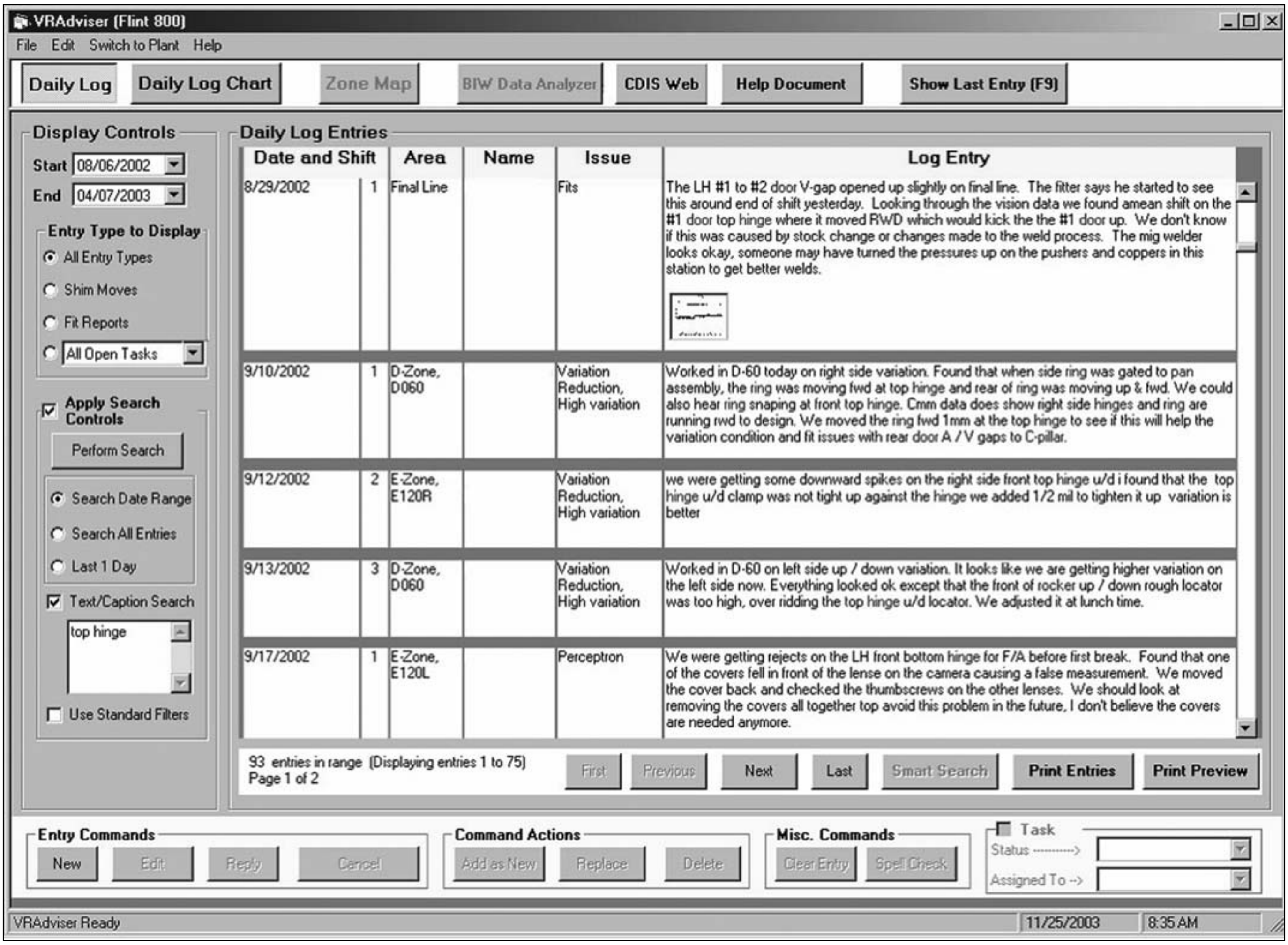


Figure 2. Opening Screen of VRA.

rent problem, even though they are unwilling to contribute more case-authoring effort than making entries in a written log book. Originally, the CBR structure facilitated the retrieval of past similar cases. VRA-2 does not have the symptom information captured in a structured way, but this CBR recall function may be salvaged via OGS. Our solution uses ontologies (see McGuinness [2002]) as a way of encoding context and symptoms. This is exploited when we process queries in a query expansion style (see McGuinness [1999]) to yield a functionally smarter search capability.

The core of VRA-2 observation is a free-text block, in which any number of sentences might be written. Attached to this free text are classifying attributes, whose values are chosen by the user from pull-down menus (see table 1). Here, some structure is restored, but users cannot enter symptom descriptions from the pull-down menus. This kind of interface was the one required to maintain a satisfied user base. Although we would have liked an interface that allowed users to input more structure, that kind of interface was not operational in our plant settings. Thus, much of the content in our database is available only as unstructured text. The clarity of knowledge capture and the structured similarity search of VRA-0 are lost in VRA-2.

The sociotechnical interplay here is interesting. Although there were some technical challenges, the core reasons for evolving from VRA-0 to VRA-2 were social: users did not think they had time to author cases, and it was not feasible to create a dedicated group of case authors. More details concerning the system evolution and the social and technological issues influencing the evolution are in Morgan et al. (2003). Users, however, were enthusiastic about using VRA as a communication tool, as it was recognized that this immediately helped daily work. Since the elements of cases were being captured in partially structured log entries, we decided that this database of log entries might still function as a lessons-learned archive (the original purpose of the CBR system), if a sufficiently "smart" search engine could be devised. The next section describes VRA needs for ontology-guided search and our solution path.

Uses of AI Technology: Ontology-Guided Search

VRA achieved striking early acceptance while functioning essentially only as a communication tool. Its primary everyday impact was to facilitate communication across shifts of staff, and it had the added value of being available from multiple workstations, thereby replacing the single written logbook. While it has had significant impact providing improved communication, it is also a problem-solving tool. Previous entries in the database that have relevance to new issues can be found. It can also be searched to provide automatic report generation about problems and work items over particular time periods, by plant, by zone, and by worker or work area.

The current VRA allows several versions of simple string-based search. However, the user community specifically requested the ability to locate references to concepts and log entries that are related to their search terms, either for search queries that fail to retrieve any exact string matches or for queries that do return results but have additional closely related concepts that may also be relevant for the user. For these reasons, we constructed an OGS engine to infer structure and interrelationships on the free text without requiring the user to take on the additional burden of more complicated data entry. While an alternative might be to capture user entries in a controlled, semistructured language, based on previous GM work on controlled languages (Godden 1998), we believe that this would place too great a burden on users even if VRA had a built-in controlled language checker. Another alternative would be to depend upon markup information generated either by automatic tools or humans and then search on the meta information. However, GM deployments have found that users are unwilling to do manual markup. Additionally, automatic markup tools might provide some assistance both in generating background ontologies and in providing automatic markup. While Clear Forest and other similar tools have useful entity identification and extraction capabilities, they are primarily used for text analysis and mining and do not contribute substantially to our initial work in ontology creation and use. We are, however, actively evaluating such tools for follow-on research regarding the identification of causes and corrections of plant issues described in VRA user logs.

Date	Date of entry
Shift	Shift (first, second, third)
User	Names of users, customized for each location
Model	Model produced in plant (043, 051)
Zone	Plant zone in question (zone A, zone B)
Station	Station in zone (station A010, station B024)
Issue	Problem issue (parts quality, tooling)
Issue Detail	Details associated with above issue (part is bent, locator block)
Task Status	Status of task if entry is a task assignment. Task status options: open, in process, done
Task assigned to	User Name

Table 1. Attribute List.

Prior work using data that was contracted to be marked up using controlled vocabularies (using either manual or automatic techniques) was found to be inconsistent and inadequate for dependable searches (McGuinness 2000). Thus, while our work uses metatagging information if available, it does not count on this meta information. An anonymous reviewer adds "Schlumberger worked on automatic metadata collection in 2001–2002 and eventually concluded that the available systems did not eliminate the need for a manually generated taxonomy."

Our approach uses ontologies we built from the starting points available from within the company and driven by the needs of our application. We did not take an approach that utilizes automatically generated taxonomies. As has been pointed out in other literature (for example, in Delphi [2002]), automatically generated taxonomies can require large data sets for training as well as having control and accuracy issues, and while they may have benefits of scaling and certain kinds of efficiency, the trade-offs were not seen to be of benefit to our effort.

The supporting domain ontologies facilitate intelligent search of unstructured text. Seven interrelated ontologies comprising approximately 200 concepts have already been seeded for the first OGS prototype, including: (1) process elements—tooling, robots, operators, transfer mechanisms, welding, anything used to make a vehicle that is not a part of the vehicle; (2) process issues—such as robot failures; and (3) parts, subassemblies, and part features. Parts are individually inventoried items that make up the vehicle, for example, the left front door handle. Subassemblies are specific to the manufacturing process. Part features include commonly referred to items such as the door ring that are abstractions of various parts and subassemblies. Additional initial ontologies include: (4) single part issues—relate to only one vehicle component, such as a ding in a fender; (5) multiple part issues—relate to two or more parts, especially misalignments, unsatisfactory gaps, malformations of joints between parts; (6) data analysis—results of analysis of measurement data generated by optical and mechanical gages; and (7) plant locations—zones and stations organized topologically or functionally.

These initial ontologies include common terms and morphological variants used in the plants. The ontologies contain information found in log entries, synonyms and common misspellings, as well as a canonical form of each concept. For each ontology, we capture subclass relationships between concepts (for example, "Hood" is a subclass of "Panel") as well as "part-of" relationships (for example, "Tailgate" is part-of a truck "Box"). Additionally we capture various semantic properties of the concepts such as front versus rear position as well as indications of the source of the concept, the person who entered the concept into the ontology, and so on.

The initial ontology built for enhanced retrieval focuses on subclass, synonym, and containment relationships along with meta information for ontology evolution. We have done some additional design work on using more sophisticated ontologies with expanded property information including more value restrictions, cardinality, enumerated filler sets, and so on. At present, the ontologies are being maintained in the Protégé-2000 environment and deployed into VRA in RDF format. We may convert to OWL in the future if user feedback indicates the need for greater expressive power.

The initial search algorithm for VRA uses both the subclass and part-of relations, but this could expand in future versions as usage analysis is performed. Synonyms in the target text are normalized to a canonical form during the search before comparison is made with the ontology. After exploiting this simple similarity-based retrieval for terms from background ontologies, we will evaluate how well the retrieval is doing, and we do not anticipate the need for any full or partial parsing of the natural language text. We expect results similar to those found in PChip (McGuinness 1999) and FindUR (McGuinness 2000). In those applications, we found improved recall with little degradation of precision. Thus, without OGS, simple textual searches typically missed relevant information because the documents (in this case, the log entries) were short and contained few words to search on. When ontology-enhanced search was used, queries were expanded to include more words to search for, and thus, relevant documents could be found. Since the documents being searched were in a limited domain, there were few problems with multiple senses of words introducing problems that hurt precision. In our database, case entries are similar—the textual fields do not contain long descriptions, and the content is limited to plant assembly information. In the full range of FindUR deployments, query expansion was done along a range of complexity. The simplest deployments used subclass relationships only, and more sophisticated search interfaces leveraged domain and range information, value restrictions, cardinality, disjoint class information, enumerated sets, roles, and subrole hierarchies. When interface requirements were such that they demanded more expressive and precise query manipulation, the additional ontological information was leveraged effectively. When, however, interfaces were required that simply used straight text input, the simpler ontologies were used as background information. We are exploring a similar deployment strategy here.

Evaluation criteria for search includes impact on precision and recall using individual ontologies; review of structure modification requests (for example, it is not clear in advance how important term relationships will be, compared to raw occurrences of terms); review of user interface concerns regarding user-suggested updates, as cited later; and review of requests to evaluate whether patterns emerge, conflicts arise, ontologies become stable, and so on.

Application Use and Payoff

VRA has been deployed in one plant for about five years and in fifteen plants for more than one year, with new installations in an additional two plants. In each plant where it is deployed, VRA is used daily. For example, at the GM truck plant in Silao, Mexico, the Dimensional-Engineering Team begins its daily morning meeting with a review of the previous day's entries. About 10 entries are created per day per shift in the plants where it is installed.

A formal business case was created to quantify the benefits and payoff of VRA, and we sketch the elements of this business case here. The business case presents evidence that VRA is a mechanism for cost avoidance, a more systemic concept than cost savings. Scenarios are constructed about "events" that generate cost. Formulas estimate how using VRA reduces these costs. The frequency of the events over a time period (like a year) is estimated. The result is a dollars/year estimate of cost avoidance generated by using VRA.

For VRA we had three scenarios: (1) wasted time in connecting, (2) continuous-improvement problem solving, and (3) crisis problem solving.

Wasted Time in Connecting. In this scenario, we envision an exchange between two team members in which there is wasted time, say, through "telephone tag" or by losing notes that have to be recovered or by forgetting to respond to a request or other such "slips" that can occur when everybody is busy and doing a number of things at the same time.

Continuous-Improvement Problem Solving. Typically, this scenario includes two types of activities: (1) reducing process variation and (2) resolving small issues not likely to require rework (adjustments to a vehicle before it can be released from the factory).

Crisis Problem Solving. The events for this scenario are "breakdowns" that cause definite warranty or rework until they are solved. These problems generally get a lot of attention when they occur. Solving these, definitely and directly, improves the productivity of the plant and the overall quality of the vehicle output. Additionally, since warranty claims quantitatively decrease customer loyalty, fixing this problem also addresses customer loyalty.

The first two scenarios have to do with moving the work process to a less wasteful state. The third scenario has to do with returning the current state of the work process to its normal operating conditions.

In the first two scenarios, jobs are done quicker and less time is wasted. This time savings is converted to a dollar figure by multiplying by a wages-per-hour estimate. Here we could have left the savings in hours rather than converting to dollars. In a company where contracts fix most wage costs, it may not be realistic to convert time savings to dollars, as if wages could be "saved." However, this device of converting time to money might be accepted as a metaphor for the value of saved time, without interpreting it literally.

In the third scenario, the value of avoiding rework, warranty costs, and lost sales is converted to dollars through the use of economic models. Additionally, lowered warranty usage translates to higher customer satisfaction and a higher percentage of return customers. Establishing a rigid analytical justification for the assumptions in such models and for values of their parameters is difficult, and we made do with "best guess" estimates in combination with some quantitative market research numbers that were available. Even with the variability of the evaluation parameters, using conservative scenarios showed considerable cost avoidance.

Application Development and Deployment

The development process began at the GM Research and Development Center. It was noticed that complex dimensional-management problems were being solved daily in the assembly plants without any systematic record being kept of this problem-solving activity. This suggested a CBR system, and a first prototype was constructed at the R&D Center. Its evolution under user feedback was described earlier. Four researchers working approximately 20 hours per week each, with the cooperation of two or three plant engineers over a year. yielded essentially the final prototype, using Microsoft Visual Basic and Access. Since then, there have been many evolutionary changes and a process to convert the code to a web-based system.

Currently, a software supplier is working with us on a production-hardened version of the web-based code. The cost of basic development was the salaries of the involved parties. As the project matured, suppliers have been involved with completing and hardening the code, which has involved further cost. Most of the difficulties encountered in this project have to do with the human-computer interface and in fitting the system into the plant workflow and sharing patterns. The following is a list of some of the practical lessons learned from our experience designing, developing, and deploying VRA.

1. The interface for input and retrieval in our plant settings needed to appear simple and natural. Thus, a natural language input and output format was required.

2. While structured case information may be seen to have future value from retrieval and reporting perspectives, this was not viewed to have enough benefit to offset the perceived burden of authoring case information in a structured format.

3. The exact form of the user interfaces and their supporting structures cannot be worked out in advance. Rather, the user community must be given the opportunity to try out prototypes and have them modified based on experience. This is consistent with the grassroots development process noted in Morgan et al. (2002).

4. Improved communication is received with more immediate enthusiasm than providing a problem-solving tool, whose usefulness takes time to establish.

While we hoped the incentive of being able to reuse the solutions to previously solved problems would be enough to motivate the users to author structured cases, we found this was not so. They would use the system only by means of natural language input. Thus, our only options were to have authors or editors separate from the users (impossible in our setting) or to provide a mechanism that provides some access to the structure and content implicit in the free-text fields. We believe that our ontology-based approach to smart search is an appropriate reaction to the environment we find in our plants and offers a place for AI technology to provide value and impact in industrially deployed plant communication and retrieval systems.

Maintenance

Once an application is dependent upon a background set of knowledge, it becomes important to have an evolution environment for obtaining, checking, and maintaining the knowledge. For example, a new interface to support OGS will allow a user to make a suggestion to add a term to a particular ontology. We are currently investigating the requirements for such an interface. The suggestion log would then be submitted to an internal ontology owner for approval and incorporation into the next version.

Both academic and industrial work has been done on ontology evolution environments that this project can draw on. In their paper "Industrial Strength Ontology Management," Das, Wu, and McGuinness (2001) provided a list of ontology management requirements that we endorse and include in our evolution plan. The list includes scalability, availability, reliability, and performance; ease of use by domain-literate people; extensible and flexible knowledge representation; distributed multiuser collaboration; security management; difference and merging; XML interfaces; internationalization, including support for multiple languages; and versioning.

Over time, as analysis is done on the size, usage, and updating requirements of the ontologies, we will create an ontology evolution environment that addresses the concerns listed above that are most important to the GM deployments. We anticipate ease of use, availability, and multiuser collaboration to be the most important initial concerns. However, difference and merging, versioning, extensibility, and internationalization will become more important as VRA has a longer life and is deployed in more varied locations. As already noted, the VRA application is available in English, Spanish, and German. Maintaining versions in different languages has obvious ontology implications.

VRA will continue in the near term to be guided by the research group, but our plan is for the plant data managers (one per plant) to handle the day-to-day management of each plant system, while a few selected managers will control the ontology maintenance process. Each GM vehicle model is manufactured for a period of years, so the knowledge refresh process can have an evolutionary flavor. The model changeover process will use VRA as a diagnostic aid, and this will prime the knowledge base as new vehicles are produced.

Summary and Discussion

Working closely with the GM asssembly centers, we have deployed an AI-based knowledge system, the Variation-Reduction Adviser, which has accomplished measurable benefits for the GM vehicle-assembly process. As a result, the system is being deployed to all assembly centers, and it is in daily use in all the centers in which it is currently deployed.

The system underwent a considerable degree of restructuring based on user feedback. This feedback focused on the perceived burden of authoring structured cases. An ontology-guided search mechanism has been designed to allow free-text case authoring while maintaining the use of the case base as a solved-problems archive.

The success of the system is due to its ability to address everyday needs for communication in ways superior to previous processes. This capability is our explanation for the significant user pull for VRA and is the main reason we believe for its success.

The knowledge management nature of VRA is more in the class of lightweight and grassroots diagnostic systems, such as Xerox PARC's Eureka, rather than more managed systems (as noted in the beginning of this article). Commercially available technology for automatically capturing metadata (for example, Clear Forest, Stratify, or Interwoven's Meta-Tagger) typically have their greatest success under conditions different from this application, although they were considered. Protégé was our choice for an ontology management tool, preferable to taxonomy-focused systems (Wordmap, Inxight), since we have more than simple taxonomies to manage (note our use of "part-of" as well as "isa" relations).

Corporate policy does not allow us to specify the ontology, the details of its implementation, the algorithm for OGS, or other material judged to offer GM a competitive advantage. However, the essential features are outlined, especially in the section on uses of AI technology so that the essence of our approach is clearly revealed.

AI was critical to the success of this deployed application. Both the CBR inspiration and the functionality of OGS were essential to frame and drive the system toward its eventual user acceptance and its suitability to its dual functions in communication and problem solving. It is now being considered as a model at GM for shops where problem-solving teams must collaborate, share, and remember, both within and across communities of practice.

Our practical lessons learned from this application were listed in the section on application development and deployment, but the fundamental lesson—applicable to any business process—is the grassroots development process. Users must be listened to aggressively and the system changed to fit their work practices. Knowledge as communication is received, understood, and managed much more intuitively than knowledge as gems. Best practices are valuable, but connecting with peers is essential.

Acknowledgements

We appreciate support from the GM Flint Assembly Center: Dave Coumes, Dave Holland, Gene Morrison, Dan Rach, and Nathan Grzymkowski. We also acknowledge support from GM Dimensional Engineering: Glenn Pacer, Gary Telling Tony Grix, and Richard Grimes. We acknowledge the PARC scientists who joined us in 2003: Jack Whalen, Erik Vinkhuyzen, Robert Moore, Robert Cheslow, and Danny Bobrow. This article is a revised version of a Deloyed Application Award winning paper presented at the 2004 Innovative Applications of Artificial Intelligence Conference. Finally, we would like to thank the anonymous reviewers for several useful suggestions and the *AI Magazine* editors for their help in preparing this article for publication.

References

Ackerman, M. S. 2003. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction* 15(2–3): 179–203.

Bobrow, D., and Whalen, J. 2002. Community Knowledge Sharing in Practice: The Eureka Story. *Reflections* 4(2): 47–59.

Cafeo, J. A.; Gibbons, D. I.; Lesperance, R. M.; Morgan, A. P.; Sengir, G. H.; Simon, A. M. 2001. Capturing Lessons Learned for Variation Reduction in an Automotive Assembly Plant. In *Proceedings of the Fourteenth International Florida AI Research Society Conference*, 89–92. Menlo Park, CA: AAAI Press.

Das, A.; Wu, W.; McGuinness, D. 2002. Industrial Strength Ontology Management. In *The Emerging Semantic Web*, ed. I. Cruz, S. Decker, J. Euzenat, and D. McGuinness. Amsterdam, The Netherlands: IOS Press.

Delphi Group. 2002. *Taxonomy and Content Classification: Market Milestone Report*. Boston, MA: Delphi Group.

Godden, K. 1998. Controlling the Business Environment for Controlled Language. Paper presented at the Second International Workshop on Controlled Language Applications. Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, May 21–22.

Kwiecien, S.; Kramek, B.; Aiello, S.; Moola, M.; and Swarup, S. 2001. Best Practice Replication: The Evolution of KM at Ford Motor Company. *Knowledge Management* 5(3) (October).

Leake, D. ed. 1996. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. Menlo Park, CA: AAAI Press.

Leake, D.; Birnbaum, L.; Hammond, K.; Marlow, C.; and Yang, H. 1999. Integrating Information Resources: A Case Study of Engineering Design Support. In *Proceedings of the Third International Conference on Case-Based Reasoning* (ICCBR 1999). Lecture Notes in Computer Science, Volume 1650, 482–496. Berlin: Springer-Verlag.

McDermott, R.; O'Dell, C.; Hubert, C. 2000. *Building and Sustaining Communities of Practice: Final Report*. Houston, TX: American Productivity & Quality Center (APQC) (www.apqc.org/portal/apqc/site/store).

McGuinness, D. 1999. Ontology-enhanced Search for Primary Care Medical Literature. Paper presented at the International Medical Informatics Association Working Group 6—Medical Concept Representation and Natural Language Processing Conference, Phoenix, Arizona, December 16–19.

McGuinness, D. 2000. Conceptual Modeling for Distributed Ontology Environments. In *Proceedings of the Eighth International Conference on Conceptual Structures: Logical, Linguistic, and Computational Issues* (ICCS 2000), Lecture Notes in Computer Science, Volume 1867. Berlin: Springer-Verlag.

McGuinness, D. 2002. Ontologies Come of Age. In *Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential*, ed. D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster. Cambridge, MA: The MIT Press.

Morgan, A. P.; Cafeo, J. A.; Gibbons, D. I.; Lesperance, R. M.; Sengir, G. H.; Simon, A. M. 2001. CBR for Dimensional Management in a Manufacturing Plant. In *Proceedings of the Fourth International Conference on Case-Based Reasoning* (ICCBR 2001). Lecture Notes in Computer Science, Volume 2080, 597–610. Berlin: Springer-Verlag

Morgan, A. P.; Cafeo, J. A.; Gibbons, D. I.; Lesperance, R. M.; Sengir, G. H.; Simon, A. M. 2002. The General Motors Variation-Reduction Adviser: An Example of Grass Roots Knowledge Management Development. *International Journal of Intelligent Systems in Accounting, Finance, and Management* 11(4): 269–276.

Morgan, A. P.; Cafeo, J. A.; Gibbons, D. I.; Lesperance,

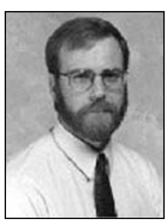
R. M.; Sengir, G. H.; Simon, A. M. 2003. The General Motors Variation-Reduction Adviser: Evolution of a CBR System. In *Proceedings of the Fifth International Conference on Case-Based Reasoning* (ICCBR 2003). Lecture Notes in Computer Science, Volume 2689, 306–318. Berlin: Springer-Verlag.

Rychtyckyj, N. 1999. DLMS: Ten Years of AI for Vehicle Assembly Process Planning. In *Proceedings of the Eleventh Confrence on Innovative Applications of Artificial Intelligence*, 821–828. Menlo Park, CA: AAAI Press.

Smith, R., Farquhar, A. 2000. The Road Ahead for Knowledge Management: An AI Perspective. *AI Magazine* 21(4) (Winter): 17–40.

Watson, I. 2002. *Applying Knowledge Management: Techniques for Building Corporate Memories*. San Francisco: Morgan Kaufman Publishers.

Alexander Morgan has been a research scientist at General Motors for more than 25 years. His main areas of research include the numerical solution of systems of polynomial equations and the development of practical knowledge systems. Most recently, he has been involved in projects involving data mining, text analysis, and information extraction for health care, quality, and warranty databases. He has a Ph.D. in mathematics from Yale University. His e-mail address is alexander.p.morgan@gm.com.



John Cafeo is a lab group manager at the General Motors Research and Development Center. He obtained a BSME (1983), an MSME (1984), and a Ph.D. (1992) all from Penn State. His master's work was in the area of acoustical modal analysis, and his doctorate work was the design and development of a laser-based vibration measurement system. John is a member of SAE and SEM. He is currently an associate technical editor for *Experimental Techniques* magazine and a member of the ASME Codes and Standards Committee (PTC-60) for verification and validation in computational solid mechanics. His e-mail address is john.cafeo@gm.com.

Kurt Godden is a technical fellow at GM R&D, working on the semantic web and ontologies. He has held positions from academe to industry, including two major corporations and a startup. Most of his 25-year career has dealt with language technologies, including natural language query systems, machine translation, and speech understanding. He has a U.S. patent pending for an *n*-best recognition algorithm. Kurt obtained his Ph.D. in computational linguistics from the University of Kansas and an M.S. in management of technology from Rensselaer Polytechnic Institute. He may be reached at kurt.godden@gm.com.

Ron Lesperance is a staff research engineer in the Manufacturing Systems Research Laboratory at the General Motors R&D Center. His research interests include computer vision, case-based reasoning, and ontology-guided search. He received his Ph.D. in electrical and computer engineering from the University of Michigan. He can be contacted at the GM R&D Center, Mail Code 480-106-359, 30500 Mound Road., Warren, MI, 48090-9055; ronald.m.lesperance@gm.com

Andrea Simon has a B.S. in computer science and psychology from Wayne State University in Detroit, Michigan, and a master's degree in computer science from the same institution. She is a senior project scientist at General Motors Research Laboratories where her current area of interest is in the customization and adoption of knowledge-sharing techniques and practices to large corporations. She can be reached via e-mail at andrea.m.simon@ gm.com.

Deborah McGuinness is codirector and senior research scientist of the Knowledge Systems, Artificial Intelligence Laboratory (KSL) at Stanford University and CEO of McGuinness Associates Consulting. Deborah's research focuses on the semantic web, ontologies and ontology evolution environments, knowledge representation and reasoning, explanation, and search. Her consulting focuses on helping companies utilize artificial intelligence research in applications such as smart search, ontology management, information integration and sharing, online commerce, and configuration. Her e-mail address is dlm@mcguinnessassociates.com.

James Benedict is the COO of McGuinness Associates Consulting. James has a master's degree in international management with more than 20 years of work experience in various industries providing management of corporate information technology, accounting systems, and business planning. James helps clients plan, develop, deploy, maintain and evaluate the business impact of applications of artificial intelligence and semantic web technology. His e-mail address is jbenedict@mcguinnessassociates.com.