Proceedings of IMECE2005
2005 ASME International Mechanical Engineering Congress and Exposition
November 5-11, 2005, Orlando, Florida USA

IMECE2005-79885

DISTRIBUTED ADAPTIVE OPTIMIZATION IN VIRTUAL BODY ASSEMBLY

RICHARD J. GERTH
Center for Automotive Research
1000 Victors Way, Ann Arbor, MI 48108, USA
rgerth@cargroup.org

SVEN BRUECKNER
Altarum Institute
3520 Green Court, Ann Arbor, MI 48105, USA
sven.brueckner@altarum.org

ABSTRACT

The Digital Body Development System (DBDS) is 4 year project to shorten the time to launch vehicles by improving the launch problem solving process. The project is based on two concepts: virtual functional build and an intelligent agent based decision support system. This paper presents a novel architecture for the decision support system that streamlines the launch process through the integration of a virtual assembly simulation, problem identification, and solution generation and evaluation. Following the virtual functional build process, the architecture deploys a number of multi-agent systems to provide system functionality, such as problem knowledge retrieval, solution generation, modification, and evaluation. The architecture has been implemented and will be explained on a simple 2-D door model and a casebase of 100 cases. Results show that the DBDS can find the correct solution much faster than a random search and can automatically implement and evaluate the solution.

INTRODUCTION

The car body is one of the most important vehicle systems. In terms of vehicle model launch it can be considered the most important vehicle system as

1. it is often the bottleneck during launch,
2. it is often the most costly system (powertrain costs can be spread across multiple programs),
3. platform development and capital investment costs limit the car manufacturer's ability to introduce new models (model must run for a certain length of time to recoup investment), and
4. it is the first system the customer sees when first considering a vehicle for purchase.

While many efforts have been focused in improving product design, relatively few efforts have been focused on improving vehicle launch, specifically as it pertains to the body. Increasing body quality during launch results in increased customer satisfaction. A recent study of 14 vehicles found a strong correlation between body gaps and flush around the doors and customer satisfaction in fit and finish as measured by J D Power Initial Quality Survey [1].

Thus, the DBDS focuses primarily on the vehicle launch process, which includes die tryout and assembly system validation (see Fig. 1).

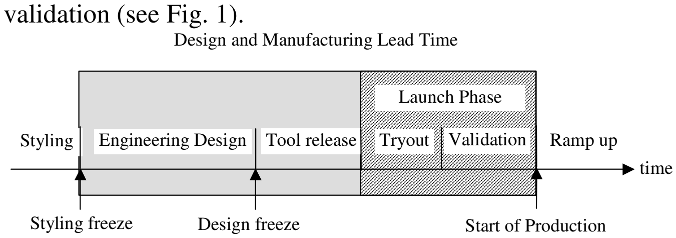

Figure 1. Auto Body Development Cycle

Detailed engineering design of individual parts and components begins following design freeze. This typically includes a finite element analysis (FEA) of the nominal design to examine stresses, vibration, crash testing, etc., as well as a tolerance analysis to determine how the components will fit. This latter analysis often involves identifying designs that are sensitive to variation and making the design more robust by changing and redesigning parts to reduce geometric effects. Once the individual part design is set, it is released for "tooling" (tool release – i.e. the process of constructing the stamping dies), and the functional build process begins.

Functional build is a critical process in launching a vehicle, whereby individual prototype parts are stamped and then sent to a central location to be assembled into a prototype vehicle body [2]. Since production tooling is often not yet available, the body is fastened with screws and rivets; hence it is called a "screwbody." The screwbody is examined by experienced experts who must decide whether gaps and interference conditions between individual parts are sufficient to warrant changing the dies, the welding tooling, clamp locations, etc. If it is decided that a change is warranted, then the dies may have to be returned to the supplier to be changed. If a change is not warranted, then the specifications may be changed to match the part shape. This usually involves a uni- or bi-directional opening of the part tolerances. The process is then repeated after the changes have been implemented. It is not uncommon to have three or more functional build evaluation bodies during a vehicle launch, which is costly and time consuming.

The next evolution of functional build is virtual functional build (VFB). A key enabling technology is optical measurement technology. There are a variety of technologies ranging from laser scanners, such as Perceptron's ScanWorks [3], to white light systems, such as Cognitens' Opticell [4], and holographic systems, such as Coherix' ShaPix [5]. These technologies provide precise part representations in virtual form. Stamped parts exhibit significant physical differences from their CAD nominal values due to factors in the tooling and forming processes, which is why the optical data is so critical. Rather than sending physical parts to a central location to be assembled, suppliers optically measure their parts and send the virtual part representation to a central web site. Then the virtual parts are assembled, and the problem areas are identified. The major advantage of virtual assembly is one is freed from logistical requirements of having all parts sent to a central location at a scheduled time. Coordinating the timing and shipment of up to 30 different suppliers and hundreds of different parts is extremely difficult. Also, the screwbody process itself takes generally 4 to 6 weeks to complete. If critical parts are delayed, then the build can be further delayed. VFB can be completed in a far shorter amount of time. Assembling the virtual part representations is much faster than physically placing parts in fixtures and riveting or screwing the parts together. Furthermore, any part that has not yet been manufactured may be replaced by its CAD nominal as a best guess for what the part will look like. Virtual functional build saves the time and cost of assembling a physical prototype, and allows users to create many more virtual prototypes than physical prototypes. This is particularly important during the iterative die tryout process. The capability to quickly evaluate the effect of a die change on the body assembly is a functional evaluation of the part, as opposed to a pure specification based evaluation.

VFB is in its infancy, and there are many process issues that must be addressed, such as purpose of the VFB, information requirements prior to measurement, fixturing requirements, etc. For example, is the purpose of the measurement to check the part dimensions relative to their CAD nominal dimensions, or is the purpose to determine whether the part will cause a problem during assembly? The answer to this question may affect how one fixtures the part. If the purpose is to compare the part relative to part nominal, then one might want to fixture the part in as free a state as possible while still ensuring repeatable results. A comparison of the virtual part with the CAD nominal file would show where dies might need to be adjusted to achieve a better part. If the purpose is to determine assemblability, then one might want to fixture the part as it would be in the assembly tooling (i.e., completely overconstrained). Displaying parts in body position would then show gap and interference conditions as one would expect to see during assembly.

However, VFB as described above is not able to predict the dimensional quality of the assembly. Even in the previous example, where the virtual parts are placed in body position, one would not be able to predict the dimensions of the resulting assembly, because simple visualization cannot account for the springback that occurs after welding and the tooling clamps are released. This requires the integration of tolerance analysis and FEA simulation. Most tolerance analysis models are Monte Carlo simulation based and assume rigid parts, i.e., the assembly process does not affect the dimensional quality of the parts, which is not true of body assembly. Tolerance analysis models begin with a nominal representation of the parts and assembly tooling, apply manufacturing variation to the part and tooling features (from design specifications or actual manufacturing data), simulate the assembly process in the appropriate sequence, and output the desired measurements. The output is typically a distribution and a sensitivity analysis for each measurement. Integration of FEA models allows the tolerance simulation to take elastic deformation of the parts induced by spot welding into account. The software typically does not account for plastic deformation, and hence heat distortion effects from welding are not modeled. Conceptually, the parts are assembled in the software. Weld points are identified and the parts are forced into full contact at those points. These points are held as boundary conditions. Then the FEA program minimizes the stress in the assembly by changing the shape of the part according to the boundary conditions.

Several groups have developed a joint FEA-dimensional variation simulation engine: General Motors (GM) has developed one for internal use; Dessault Systems released such an engine in their Catia V5 product [6]; and UGS PLM has incorporated this functionality in their VisVSA V5.1 product [7]. Future versions of the software will be able to predict the amount of residual stress in a functional build assembly.

With these new optical measurement and simulation tools it is possible to virtually assemble and predict dimensional quality including variation. Thus, engineers will have a tool to understand dimensional problems with actual body parts during launch. Despite the large opportunity to improve timing and reduce cost through these two technologies, many of these gains will be difficult to achieve due to the following:

1. System complexity: Designers will need to make decisions in concert. Any decision made on one part could have an impact on other adjoining parts. For example, a change on a rear reinforcement rail to ensure it will assemble with a rocker panel can also impact how the rail fits with the wheel housing.
2. Excessive Engineering Change Orders (ECOs): There are too many ECOs due to a lack of understanding by product designers of what design features and changes to design features will have a true impact on the assembly of individual parts, as well as the function of the assembly itself. A better understanding of the impact of design changes on manufactured assemblies and their variation should lead to a significant reduction in ECOs, a common disruption to a smooth and timely vehicle launch.
3. Experiential decision making: Current decision making processes in functional build (whether physical or virtual) are based on the experience and memory of individuals who have participated in previous programs. This experience requires years of hands-on practice with die making, welding, and hand assembly of panels; a knowledge base that is rare and

becoming rarer as a large portion of the work force reaches retirement. The quality and speed of decision-making can be drastically improved through data or a quantitative understanding of cause and effect relationships in the system.

4. Communication and coordination of supply chain: Too much time and cost is wasted on tooling buyoff and part validation due to unpredicted manufacturing variation, poor communication between suppliers and the customer, and lack of available information.
5. Distributed knowledge: Effective solutions to problems and their cost and time impact on the program is generally distributed in the supply base. It is difficult to identify the suppliers with the pertinent knowledge and evaluate the tradeoffs between competing knowledge.

The DBDS will overcome many of these problems by helping engineers identify and evaluate solution alternatives based on proven historical cases. The DBDS closes the design loop during the manufacturing validation phase, using functional build concepts. The DBDS builds upon the knowledge and experience gained during vehicle launch programs and applies them to simulated assembly models based on actual scanned parts.

THE DBDS

The Digital Body Development System (DBDS) is depicted in the blue box in the lower half of Fig. 2 and consists of 3 major subsystems:

1. Data Preparation and Repository Module (DPRM)
2. Virtual Assembly and Simulation Engine (VASE).
3. Solution Generation and Evaluation Module (SGEM)

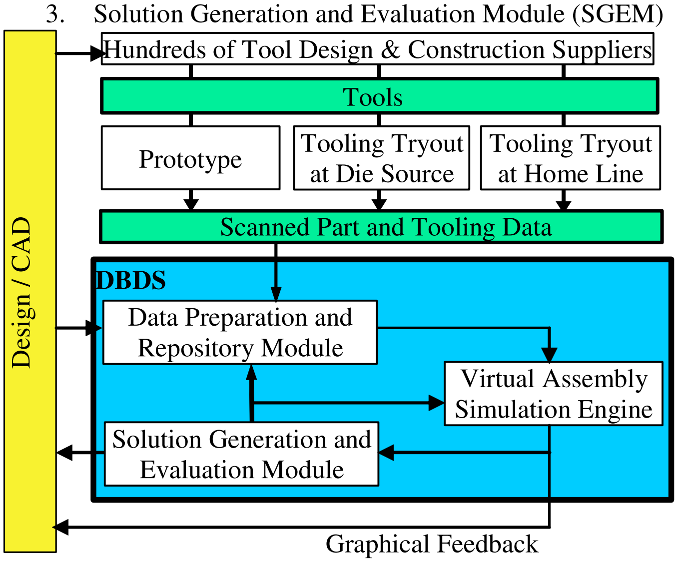

Figure 2. Schematic of Digital Body Development System

The system begins by collecting information in the Data Preparation and Repository Module (DPRM). The module acts as a central collection facility for all data in the system. It checks for data consistency and formatting before sending the data onto VASE or allowing other parts of the system to access the data. It also houses the database of historical problem-solution cases, which is used by the SGEM. In addition, the DPRM contains a set of project management and communication tools to aid in coordinating the product validation and launch process among the various suppliers and the OEM.

The VASE then simulates the assembly function and generates dimensional and residual stress distributions for specified measurements across the vehicle body. UGS PLM VisVSA V5.1 is the VASE used in the DBDS. Generally there will be several hundred assembly measurements (simulation outputs) per vehicle. These simulation results are then sent to the Solution Generation and Evaluation Module (SGEM).

The SGEM groups problems that have similar characteristics, such as measurement location, measurement type and direction, and common parts involved in the measurements, into problem areas. The problem areas are then ranked according to the problem magnitudes as well as the relationship between problems. Examples of problem relationships are problems that have the same part supplier, or passed through the same assembly steps. These relationships are similar to hypotheses of root causes.

The purpose of the grouping and ranking is to

1. reduce the problem space and identify the critical problems that will drive the solution generation and evaluation module, and
2. begin to introduce information that relates problems to root causes.

Using a database of past solutions structured according to problem relevance criteria, an agent based case retrieval network (CRN) is used to identify the best solution to the given problem areas. These solutions are sent to the VASE and automatically implemented in the model to evaluate their functional effectiveness with regards to dimensional quality and residual stress. The system continues to iterate on various solutions using local change rules to modify solutions until it finds solutions that satisfy the design requirements.

The DBDS does not explicitly determine the root cause of the system. Instead, the root causes are implicitly embedded in the problem area groupings, the relationships between problem areas, and the relevance edges in the CRN. The SGEM looks for the solutions that best map to the causal structure. In essence, one is looking for the best solution that maps into the problem characteristics under the assumption that problems with the same underlying causal structure have the same root cause, or at least, can be resolved by the same solution.

The DBDS is intended to be used iteratively throughout the vehicle program. For example, every time a die is created or modified, such as during prototype, die tryout at the die source, and die tryout on the home line, the system would be invoked (see Fig. 2). Scanned part images would be sent to the DPRM and converted to a format suitable for use by the VASE. Simulating the scanned part files, instead of the nominal CAD files would provide information on the effectiveness of die changes and need or lack of need for further changes. The DBDS would determine whether additional changes are necessary, what they should be, and their expected outcome on the assembly as determined from simulation.

ADAPTIVE HEURISTIC SEARCH

The DBDS treats the generation of solutions to problems identified in the current design as a search problem in the high-dimensional space of possible modifications to the design guided by a fitness function. Any point in this abstract search space is a set of parameterized changes to the current design. Computing the fitness of such a set of changes requires the application of these changes to the design, and the simulation and analysis of the resulting new design comparing it with the current design.

In [8] Brueckner and Parunak present an experimental application of their agent-based Adaptive Parameter Search Environment (APSE), which performs a heuristic parallel search across an abstract space of input parameters to an arbitrary simulation model guided by a fitness function defined over metrics reported during the execution of the model. The DBDS is an application and extension of APSE in which sets of design changes are treated as input parameters to the virtual assembly of a car body and in which the search is guided by the design intent of the functional build process.

The Solution Generation and Evaluation (SGE) module of the DBDS hosts an APSE search agent population, whose task it is to explore the space of possible changes to the base design for improvements that reduce or remove the problems observed in its execution. Thus, the changes to the base design are input parameters to a black-box simulation and a predefined fitness function measures the degree to which the now modified design meets the design intent.

The APSE search agents collaboratively explore the space of potential solutions (model parameters) and evaluate them through successive simulation runs. Using a Particle Swarm Optimization (PSO) algorithm [9] combined with probabilistic local hill climbing, the agents coordinate their activity so that computing resources (simulation runs) are focused on exploring the most promising regions of the search space.

Given the complexity and massiveness of the search space that the DBDS must explore in a given optimization run, the heuristic of the APSE search agents was enhanced. While search agents in APSE are guided only by the fitness of the currently known solution candidates (points in the abstract search space), the DBDS provides two additional sources of guidance for the distributed search (see Fig. 3). The first source of solution candidates is the human design team. At any point during the search process, human experts may look at the problem symptoms and the solutions the DBDS has explored so far and suggest another solution to the system. Solutions may also be suggested by the solver, a multi-agent system that seeks to match the problem symptoms to the descriptor of solution cases recorded in a case base. The retrieval is guided by the problem symptoms observed in the execution of the current design and by the fitness of solutions that have already been evaluated by the search agents.

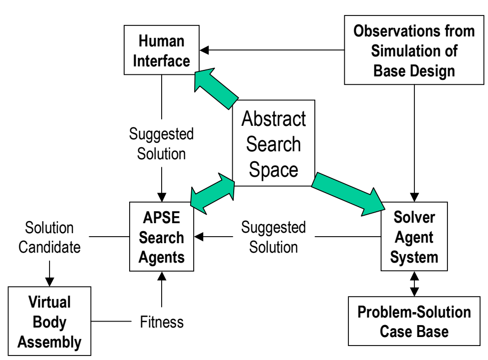

Figure 3. The DBDS Performs a Parallel Heuristic Search with Human and Case-Based Guidance

These two additional sources of creativity were incorporated into the search process by enhancing the APSE search agents' behavior. In APSE, an agent explores the search space through a series of short-range moves that are guided by hill-climbing and PSO heuristics. In the DBDS, a search agent monitors the performance of its short-range movement heuristic (rate of improvement over time) and may decide to abandon its current region in search space through a long-range jump beyond the local correlation distance of the fitness function. The destination of the jump is a solution candidate provided by the human design team or the case-based solver. Figure 4 illustrates the emerging agent trajectory in an abstract search space.

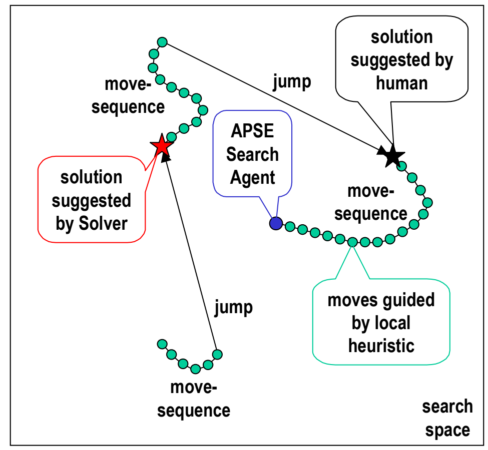

Figure 4. Agents move and jump through the search space guided by local heuristic, human input, and case knowledge

The distinction between a local improvement heuristic and a global jump to externally suggested solution candidates is sufficiently general that other solution approaches can be implemented. Just as the DBDS currently implements a case-based approach to the solution of problems with the base design, other (e.g., rule-based, model-based, etc.) approaches could be implemented independently and feed into the decision process of the search agents.

SWARMING CASE RETRIEVAL

Today's car body development process heavily depends on human expert knowledge and experience. The DBDS is a decision support system that has the ability to discover new solutions on its own through a heuristic search and evaluation in simulation, while at the same time utilizing and capturing human creativity and expertise to move from experience-based to data-driven design.

The SGE module of the DBDS includes a dynamic solver that analyzes problems with the base design as they manifest themselves in observable symptoms during the virtual assembly and that suggests solutions to these problems drawn from a set of problem-solution cases. The solver is integrated with the heuristic search process by suggesting solution candidates to the APSE search agents for their next long-range jumps and by modifying the case retrieval process based on the fitness of the solutions that have already been explored (Fig. 5).

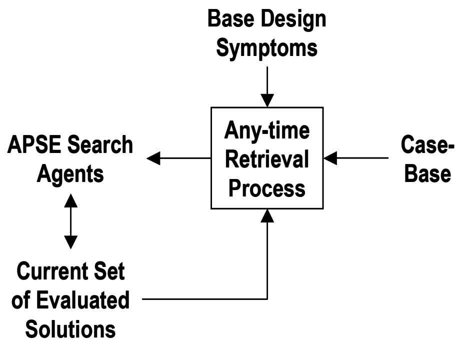

Figure 5. The dynamic solver modifies the solution candidates that it suggests to the search agents based on the progress of the exploration of the search space

The ongoing asynchronous interaction with the search agents and the continuous addition of fitness evaluations of new solution candidates requires a dynamic update of the case retrieval. This led to an agent-based any-time approach that continuously integrates changes in the external circumstances without having to restart its reasoning process from scratch.
The following details of the operation of the solver top down. First, the adaptive any-time process that manipulates the description of the current problem symptoms to provide a high-quality retrieval of high-performance solutions will be presented. This is followed by a description of the specific internal mechanics of the fine-grained agent system that drives the adaptive modification of the current problem description.

<u>Linking Emergent Clustering and Spreading Activation Case Retrieval</u>

The virtual assembly of the base design by the VASE module results in a large set of uniquely identified measurement points on the assembled car body that are either within or outside specified tolerances. Just as a fever, a cough and a runny nose are possible symptoms of an underlying viral infection, so are patterns of deviations at pre-defined measurement points on a (virtually) assembled car body symptoms of specific underlying problems (root causes) with the design.

The dynamic solver seeks to match the currently observed symptomatic patterns to those of problems encountered in the past, whose solution is recorded in the case base. The case base is organized into a simplified Case Retrieval Network (CRN) [10], which represents basic components of the problem description and the associated solution as individual nodes in a spreading activation network. The nodes representing problem components are called Information Entity (IE) nodes and a solution is stored in a so-called case node. All IE nodes that describe the problem solved in a specific solution case are linked to the respective case node through weighted relevance edges. The retrieval process first places an activation onto individual IE nodes depending on their match to the current problem symptoms and then propagates the activation through the relevance edges to the case nodes. The relative activation of the individual case nodes provides an ordering of the recorded solutions with respect to their relevance to the current problem.

The goal is to abstract away from the specific locations and count of measurement points provided by the simulation by identifying symptomatic regions on the virtual car body that may be expressions of the same underlying problem. For instance, if a door is set slightly off-center into its frame, one may find several disconnected regions along the frame in which pre-defined measurements are out of tolerance (e.g., gaps, interferences). To that end, the solver executes a fine-grained multi-agent system that continuously rearranges measurement points into clusters that form components of the problem signature (Fig. 6). The currently emerging problem signature is matched against past problems' signatures in the case base to provide a relevance measure of the available solutions. This relevance measure guides the selection of the next solution candidate upon request of an APSE search agent. A case is then selected probabilistically, based on its current normalized relevance.

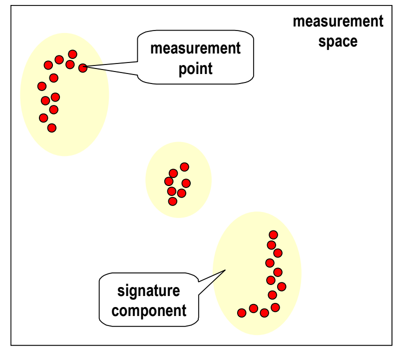

Figure 6. Clustering of Measurement Points into Signature Components

The quality of the case retrieval process is high if there is only one case (or very few cases) with a significant probability to be selected. Otherwise, a case may as well be randomly selected from the entire case base. The current retrieval quality is determined from the Case Selection Entropy (CSE) metric, which is the Shannon (Information) Entropy [11] of the case selection probabilities. The current CSE, resulting from the interaction of the current arrangement of measurement points with the Case Retrieval Network, may modify the behavior of the agents in the next clustering cycle. Similar entropy measures defined over the current preferences of an autonomous decision maker (here case selection) have been used before [12, 13] to estimate the current information these preferences actually convey and to subsequently adapt the decision process if necessary.

Figure 7 illustrates the tight feedback loop (black) between the ongoing clustering of measurement points and the current case relevance ordering provided by the CRN. Through this feedback, the identified problem regions are modified to match past experience recorded in the case base more closely while maintaining a close tie with the actual problems observed in the simulation.

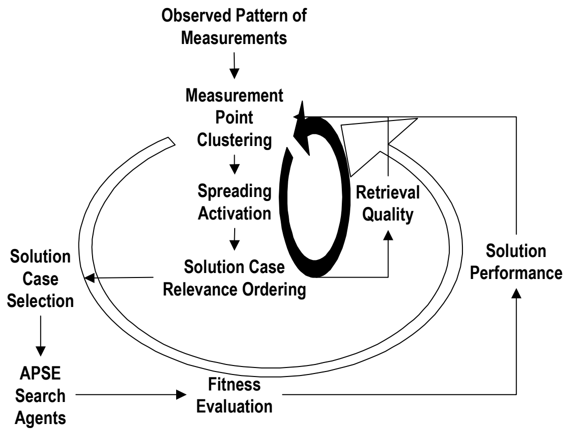

Figure 7. Adaptive Case Retrieval Guided by Retrieval Quality and Solution Performance.

The clustering process is also influenced on a larger time scale by the observed performance of solutions that have been explored by the APSE search agents (white loop in Fig. 7). If a solution case is adopted by a search agent in a long-range jump, the DBDS evaluates the fitness of the changed car body design in terms of the reduction in problems compared to the base design and the estimated cost in implementing these changes. The fitness of all solution candidates proposed by the solver is fed back through the Case Retrieval Network (activating case nodes and spreading to IE nodes) to attract the clustering mechanism away from or towards specific arrangements.

Emergent Clustering

The output of the simulation is a cloud of values for predefined measurement points. Each point is associated with geometric coordinates on the car body, but it also carries additional context values, such as part features with which it is associated, assembly process steps that came in contact with the part, or the supplier providing the part. Thus, a measurement point is located in a high-dimensional space that combines the geometric and context dimensions. Through the additional context, points that are related in the process but not necessarily in geometry can be associated to the same signature component.

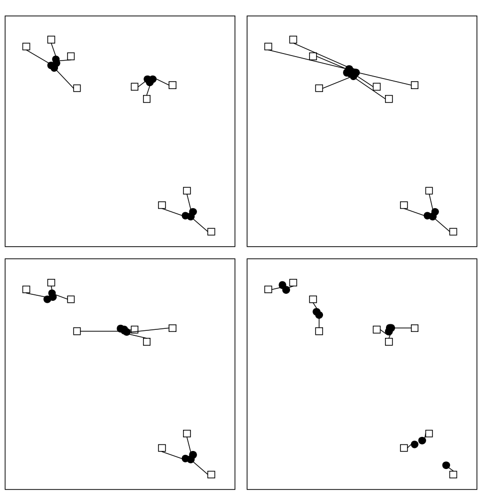

Figure 8. Possible Cluster Arrangements (black) for the same Original Measurement Points (white).

The goal is to start from the original locations of the measurement points and rearrange the points into arbitrary clusters while trying to keep each point close to its original location. As Fig. 8 illustrates, there are a number of possible arrangements that meet these qualitative objectives, as there is no prior assumption on the particular number or size of clusters. The emergent clustering algorithm is designed to potentially visit all these arrangements (with varying probability), and the feedback from the Case Selection Entropy metric and the currently known solution fitness push the clustering system out of unfavorable configurations.

Emergent any-time clustering is one of the prime examples of emerging functionality through stigmergic coordination in large-scale fine-grained multi-agent systems. Nest sorting [14], is an instance of emergent clustering observed in social insect systems. In this case, independent agents (ants) pick up or drop off passive objects with a dynamically computed probability. This behavior has been replicated in collective robotics (see for instance [15]). An alternative approach to clustering is to give the initiative to the objects themselves, which then reason about their current local arrangement and move about in space. Parunak, et al. successfully applied this approach to create large-scale, self-organizing document bases [16] and the approach was applied here as well.

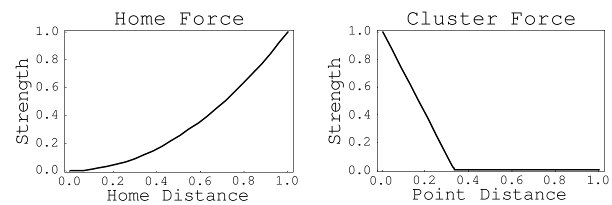

Figure 9. Forces represent agent objectives in clustering

In the emergent adaptive clustering algorithm, each point is assigned an agent, which moves through the space of geometric locations and additional context. The sum of two dynamic force vectors, representing the two objectives in the rearrangement, determines the trajectory of an agent. The first force vector ("Home Force" in Fig. 9) attracts the agent back to the original location of the measurement point. This force increases with distance. The second force vector is the sum of individual component vectors ("Cluster Force" in Fig. 9), which each attract the agent to the location of another nearby agent. The strength of this force decreases with distance. The rates in which the forces change for changing distances are dynamic parameters of the system.

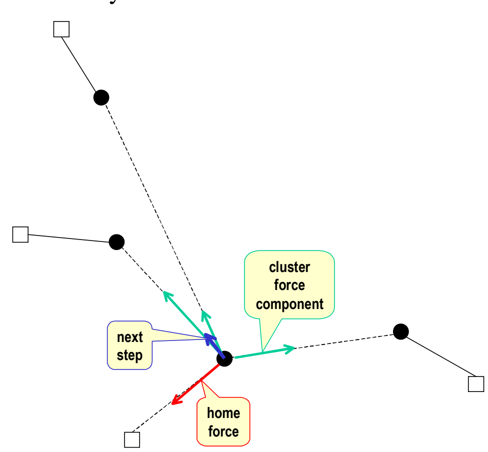

Figure 10. Iterative Local Force Vector Calculation

In each cycle, each agent calculates the home force and the cluster force vector from the position of the agents in the previous cycle. The vector sum of these two forces determines the direction into which the agent relocates in this step. The length of the step is the length of the combined vector, but limited to a relatively small step-length value (Fig. 10).

If the force calculation algorithm in the agent were deterministic and used only constant scaling parameters, then the system would quickly stabilize on one arrangement that minimizes the "tension" among the objectives. To avoid unstable minima and to explore a variety of nearby cluster configurations a small random component is added to the individual relocation calculation.

Qualitatively different cluster configurations are obtained through the feedback of the current retrieval quality and the solution performance, encoded in the Case Selection Entropy (CSE) and the fitness of solution cases.

The CSE metric offers a global evaluation of the value of the current point arrangement for the high-quality (non-random) retrieval of a solution from the case base, but it does not provide any guidance on how the arrangement should be changed to achieve a higher retrieval quality. Since higher CSE values correspond to low retrieval quality, exploration of new configurations over the exploitation of current clusters are encouraged by increasing the impact of the random component in the agents' trajectory calculations.

The fitness of solution cases that have been explored by the APSE search agents can be translated into directional guidance for the clustering agents. Before each cycle of the emergent clustering algorithm, the fitness of all cases (zero if not yet explored) is propagated backwards through the CRN to the IE nodes that represent regions of high point concentration (clusters) recorded with these past cases. Solution cases that led to an improvement in the design communicate a positive activation to their IE's while those that actually made the problem worse send a negative activation.

The positive or negative activation of IE's in the Case Retrieval Network translates to additional attractive or repulsive force components that steer points towards or away from regions in measurement space. Lenz, et al. applied a similar back-propagation approach in CRN's to guide the interactive diagnosis of failures in computer hardware [17].

EXAMPLE – 2D DOOR STUDY

The DBDS will be explained based on a simple 2D door model that was created for development purposes. The door model was then used to test the signature generation process for historical cases in the case base and the effectiveness of the retrieval process for the appropriate solution to the current problem.

<u>VisVSA Model</u>

The first step is to create a VisVSA simulation model of the assembly (see Fig. 11). There are 18 Functional Requirements (FRs) the design must satisfy: 18 gap measurements around the front and rear doors.

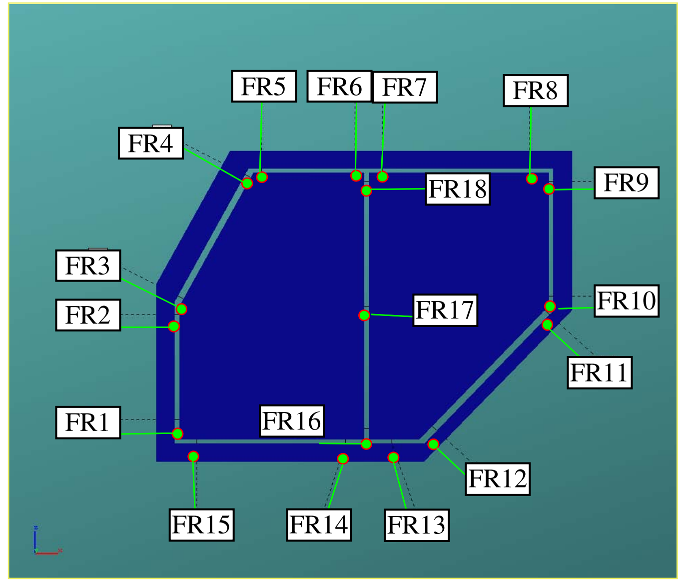

Figure 11. 2D Door Model

The 2D door model consists of a body side, a front door and a hanging fixture, a rear door and a hanging fixture, associated move statements that locate the doors to the bodyside, and 18 measurement statements. The body and doors have datum or locating features and the features associated with the FRs. These dimensional features can vary in the X and Z direction. Each variation is a dimensional parameter (DP) that is changed in each Monte Carlo simulation run. Thus, there are 2*18 = 36 DP on the doors and body side.

The assembly sequence is to first assemble the front door to the body side using the front door hanging fixture, and then to move the rear door to the body using the rear door hanging fixture. Fig. 12 shows the front door located by a 4-way locating hole in the front and a 2 way locating slot in the rear. The hanging fixture aligns these locators with the corresponding locating features on the body. There are similar locating features on the rear door. To simulate the variation associated with the hanging process, each of the 6 fixture DPs and the 3 DPs on each part were varied. Thus, with the 36 DPs from the parts and the additional 2*6 + 3*3 = 21 DPs from the hanging process there are a total 57 DPs in the model.

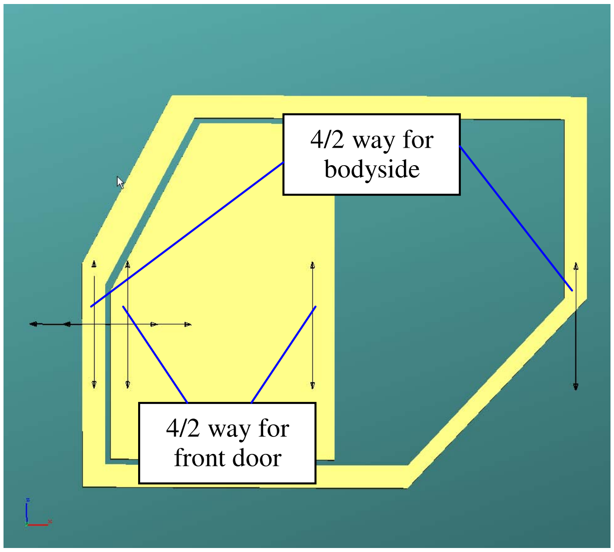

Figure 12. Front Door and Body Locating Features.

VisVSA is a Monte Carlo based simulation tool. Each of the 57 DPs is allowed to vary randomly according to a normal distribution, which centered about the nominal DP value and which variation is a function of the specified DP tolerance. Each simulation run, a random number is generated for each DP. The model then executes the assembly sequence and records the 18 FR values. This process is repeated several thousand times to generate a distribution of FR values.

Another output of the model is a sensitivity analysis. Each DP value is allowed to vary one-at-a-time between its high and low specification value, and the corresponding FR values are recorded. From these experiments it is possible to compute the sensitivity of each FR to each DP. This results in eighteen 57-dimensional vectors.

The Case Base

The case base consisted of 100 randomly created problem-solution cases. Each case was generated by randomly selecting a random number of DP features in the model and randomly varying their nominal coordinates. This ‘problem’ model was executed to provide the pattern of deviations from the functional requirements as measured in the FR features. Each FR deviation from the problem case was then enhanced with contextual information: the 57-dimensional vector of sensitivity values of the particular FR. The emergent clustering mechanism was then applied to the combined FR pattern deviations and contextual information to create a characteristic signature for the problem. The associated solution to the problem was simply the inverse of the randomly generated model modifications (a.k.a. a naïve solution). The problem signature and the solution constituted a single case in the case base. The process was then repeated 100 times to generate 100 problem-solution cases.

The 100 randomly deviated models produced 100 distinctive patterns of deviated FR values. Even though each case was associated with the same contextual information (sensitivity vectors), the emergent clustering mechanism created 100 (partially) distinct signatures – that is, for each problem pattern, the point agents of the case cluster solver converged on different cluster locations. Therefore one can conclude that the clustering mechanism preserves the salient information about the unique problem symptoms while reducing the overall dimensionality of the problem description. Figure 13 plots the center of gravity for each case signature in the case base, illustrating the diversity in the signature patterns which is required for a meaningful retrieval process.

Problem Case 1 Evaluation Exercise

The functioning of the DBDS will be explained in detail, using Case 1 as an example. Case 1 is presented to the DBDS as a base design. Case 1 has error in it and is also a case in the case base. Thus, a successful evaluation exercise will run Case 1, create a characteristic signature, identify Case 1 in the case base as the most appropriate case, retrieve its associated historical solution, apply it to the current problem and determine that the problem has been resolved (deviation from functional specification is sufficiently reduced).

The problem recorded in case 1 expressed itself as a pattern of deviations, d_i, from the expected values of the output measurements defined in the model. The error level of a design is the length of the vector spanned by the deviations in the output measurements (see eq. (1)) which, in the case of the base design, computes to 2.45 (see Table 1). It is the formal goal of the DBDS to minimize this error and, thus, meet the functional requirements.

$$\sqrt{\sum d_i^2} \tag{1}$$

Table 1. Base and Modified Design Deviations

	Base Design		Modified Design	
	d_i	d_i^2	d_i	d_i^2
FR1	-0.62	0.3790	0.00	0.0000
FR2	-0.57	0.3213	-0.02	0.0004
FR3	-0.49	0.2393	-0.02	0.0004
FR4	-0.37	0.1353	0.01	0.0001
FR5	0.07	0.0047	0.01	0.0001
FR6	0.12	0.0144	0.00	0.0000
FR7	-0.65	0.4284	0.01	0.0001
FR8	0.23	0.0545	0.00	0.0000
FR9	-0.96	0.9174	0.00	0.0000
FR10	-0.22	0.0492	0.00	0.0000
FR11	-0.33	0.1111	-0.02	0.0004
FR12	0.68	0.4587	-0.01	0.0000
FR13	0.61	0.3661	-0.01	0.0000
FR14	-0.11	0.0128	0.00	0.0000
FR15	-0.01	0.0002	0.01	0.0001
FR16	0.03	0.0012	0.03	0.0008
FR17	0.72	0.5125	0.01	0.0000
FR18	1.41	1.9833	0.00	0.0000
error =		**2.4473**		**0.0496**

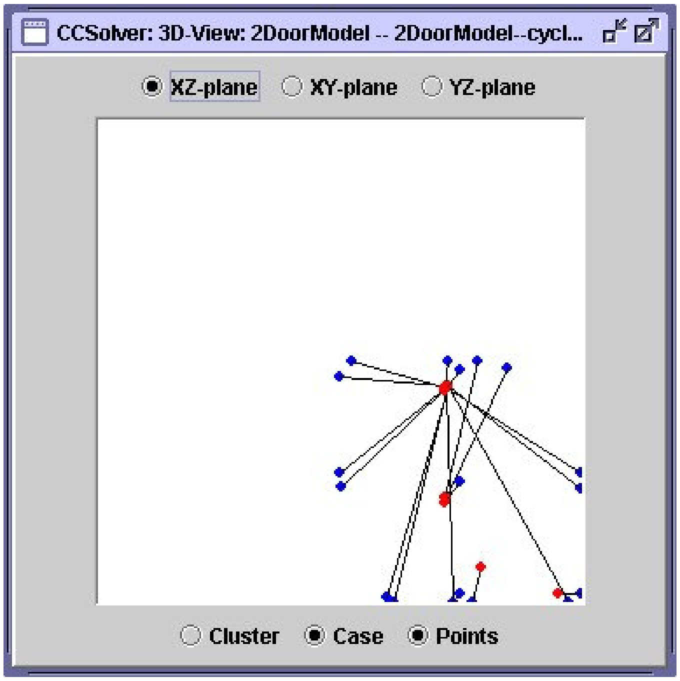

Figure 13. Emerging Signature (red clusters).

The case cluster solver is shown in Fig. 13. The pattern of output measurement deviations from the base design is handed to the case cluster solver (blue points), which uses them to instantiate point agents in the emergent clustering process that extracts the problem signature. The red points are the clusters that emerge in the process. These locations will be treated as the problem signature. This signature is matched against signatures of historical cases in the case base, and Case 1 offers the best match (rank 1, 20% selection probability).

Extracting the historical solution of Case 1 from the case base at the request of a search agent creates a new solution candidate. This candidate proposes a modified design, which is automatically implemented and evaluated by the DBDS. The resulting pattern of deviations (see Table 1) results in an error level less than 0.05, which is within the level of noise generated by the Monte-Carlo simulation of the assembly. Thus, Case 1 solves the problem posed by the base design.

Experimental Results

Ten experiments were conducted to test the effectiveness of the case retrieval process. For each experiment, one case in the case base was declared the "base design" of a DBDS run. Thus, the design problem represented in the base design was also in the case base as a historical case. The purpose of these experiments was to prove that the DBDS would be able to

1. Retrieve the historical case most appropriate to the current problem as it had been encountered before, and to do so with likelihood significantly higher than random
2. Implement the solution associated with that case and demonstrate that the problem is solved.

Table 2 shows the result of the retrieval process of the case cluster solver for the ten experiments. The first column indicates which of the randomly deviated models was used as the base design, that is, which case in the case base should be the best match to the given problem. For simplicity, the first ten problem instances were selected for the testing exercise.

Table 2. Experimental Results of 10 DBDS runs.

Experiment	**Rank**	**Selection Probability**
1	**1**	20%
2	2	4%
3	**1**	45%
4	**1**	5%
5	2	9%
6	**1**	7%
7	3	3%
8	**1**	21%
9	3	4%
10	**1**	4%

The second column shows the rank of the problem similarity metric that was assigned to the case that matched the current problem. If the SGEM worked perfectly each experiment would have a rank of "1" in this column. The third column shows the actual retrieval probability assigned to each experimental case. A high probability combined with a low rank number indicates the case clearly stands out among the cases in the case base. A low probability combined with a high rank number indicates several (many) cases would be considered a good match.

For example, in the first experiment the matching case in the case base was the first case with a selection probability of 20%. The other 99 cases all had selection probabilities less than 20%, and the sum of all their probabilities was 80%. This means that average selection probability for the other cases was less than 1% (80%/99), and this particular case was very different from the other cases. By contrast in experiment 2, the matching case was the second case with a selection probability of 4%. Assuming the selection probability for the first case is only somewhat higher than that, say 5%, the average remaining cases have an average selection probability of about 1% (91%/98). Thus, there are many cases in the case base similar to the one used in experiment 2. Despite these similarities the DBDS was able to identify the correct case within 2 selections. In general, the DBDS found the correct solution on the first try in 60% of the experiments, on the second try in 20% of the experiments, and on the third try in the remaining 20% of the experiments.

CONCLUSION

Car body development is the most costly step in the launch of a new vehicle and even small improvements of this process may yield high gains for the automotive industry. This paper presents the Digital Body Development System (DBDS) – a decision support system for the car body development team – which is an extension of the agent-based Adaptive Parameter Search Environment (APSE) presented in [8]. The DBDS is based on a modular architecture, which makes the required activities of the evaluation of the fitness of solution candidates (simulation, cost estimate) transparent for the APSE search agents exploring the space of changes to the current design of the car body.

The primary extension of APSE, besides its application to a highly complex domain, is the integration of external guidance into the local search heuristic of the agents. The DBDS enhances the decision process of the individual agent. The enhanced agent tracks the performance of the local improvement process (moves) and decides whether to abandon its current region (jump) in favor of solution candidates suggested either by the human design team or a novel adaptive case-based solver.

The case-based solver is a complex adaptive system that interacts with the APSE search agent population, providing it with solution candidates that may address currently observed design problems and adjusting its recommendations based on the fitness of the solutions that have been explored already. The solver links a fine-grained agent system that continuously modifies the description of the current problem with a Case Retrieval Network that records solutions to past problems. The retrieval of solutions is refined by the agents' modification of the problem description, driven by the currently estimated quality of the case retrieval and the performance of selected cases.

The case based solver was implemented and integrated with VisVSA, a Monte Carlo based assembly simulation engine. The system was then tested on a simple 2-D door model. A case base consisting of 100 randomly generated problem cases was used to represent the history of cases from past vehicle launches. Ten problems from the case base were selected as the base problem currently encountered, and the DBDS successfully matched all cases and solved each problem within three or fewer attempts.

The DBDS is the focus of an ongoing NIST/ATP-supported Joint Venture of 14 organizations. A patent application has been filed for the case based solver presented here.

ACKNOWLEDGMENTS

This work is supported by the DBDS Research Consortium, sponsored in part by a NIST/ATP grant (Project Number 00-00-5521). The views and conclusions in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Institute of Standards and Technology or the US Government.

The authors gratefully acknowledge the support of the participating organizations: Altarum Institute, American Tooling Center, Atlas Tool Inc., Autodie International, the Center for Automotive Research, Cognitens Inc., ComauPico, UGS PLM, Ford Motor Co., General Motors Corp., Perceptron Inc., Riviera Tool Co., Sekely Industries, and Thunder Bay Pattern Works.

REFERENCES

[1] Gerth, Richard J., Baron, J., Brahmst, E., Geddes, S. (2002) Comparative Dimensional Quality of Doors: a Benchmarking Study, Int. Body Engineering Conf., Society of Automotive Engineers, Paris, France, July 8-12.

[2] Auto/Steel Partnership (2000) Event-Based Functional Build: An Integrated Approach to Body Development - Complete Version, Auto/Steel Partnership, Southfield, Michigan, http://www.a-sp.org/publications.htm.

[3] ScanWorks , Perceptron Incorporated, Plymouth, Michigan, www.perceptron.com

[4] Opticell , Cognitens Incorporated, Wixom, Michigan, www.cognitens.com

[5] ShaPix , Coherix, Ann Arbor, Michigan, www.coherix.com

[6] Catia V5, Dassault Systems Incorporated, www.3ds.com

[7] VisVSA V5.1, UGS PLM, Southfield, Michigan, www.ugs.com.

[8] S. Brueckner and H.V.D. Parunak. Resource-Aware Exploration of the Emergent Dynamics of Simulated Systems. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems. Melbourne, Australia, 2003.

[9] J. Kennedy, R. C. Eberhart, and Y. Shi. Swarm Intelligence. San Francisco, Morgan Kaufmann, 2001.

[10] Mario Lenz, Hans-Dieter Burkhard, and Sven Brueckner. Applying Case Retrieval Nets to Diagnostic Tasks in Technical Domains. I. Smith and B. Faltings (eds.). Advances in Case-Based Reasoning. Springer Verlag, LNAI 1168. 1996.

[11] C. E. Shannon and W. Weaver. The Mathematical Theory of Communication. Urbana, IL, University of Illinois, 1949.

[12] S. Brueckner, H.V.D. Parunak. Information-Driven Phase Changes in Multi-Agent Coordination. In Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003), 950-951. Melbourne, Australia, 2003.

[13] H. V. D. Parunak, S. Brueckner, R. Matthews, and J. Sauter. How to Calm Hyperactive Agents. In Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS 2003), 1092-1093. Melbourne, Australia, 2003.

[14] E. Bonabeau, M. Dorigo, and G. Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. New York, Oxford University Press, 1999.

[15] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chretien. The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots. In J. A. Meyer and S. W. Wilson, Editors, From Animals to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behavior, pages 356-365. MIT Press, Cambridge, MA, 1991.

[16] H. V. D. Parunak, P. Weinstein, S. Brueckner, J. Sauter. Hybrid Stigmergic Mechanisms for Information Extraction. In Proceedings of the Second International Workshop on Mathematics and Algorithms of Social Insects (MASI 2003). Atlanta, USA, 2003.

[17] Mario Lenz, Hans-Dieter Burkhard, and Sven Brueckner. Applying Case Retrieval Nets to Diagnostic Tasks in Technical Domains. I. Smith and B. Faltings (eds.). Advances in Case-Based Reasoning. Springer Verlag, LNAI 1168. 1996.