Applying Distributed Adaptive Optimization to Digital Car Body Development

Sven A. Brueckner[1], Richard Gerth[2]

[1]Altarum Institute
3520 Green Court, Ann Arbor, MI 48105, USA
sven.brueckner@altarum.org
[2]Center for Automotive Research
1000 Victors Way, Ann Arbor, MI 48104, USA
rgerth@cargroup.org

Abstract. Companies in today's automotive industry are under immense competitive pressure to reduce the length of their product development cycle from initial concept to begin of high-volume manufacturing. A very costly and immensely knowledge-intensive step in this process is the creation of tools and dies required to manufacture a car body of a specified design. This paper presents a novel architecture for a decision support system that streamlines the development process through the integration of a virtual assembly simulation, problem identification, and solution generation and evaluation. Following the virtual functional build process, our architecture deploys a number of multi-agent systems to provide system functionality, such as problem knowledge retrieval or solution generation and evaluation.

1. Introduction

Today's fierce competition in the automotive industry pressures companies to find ways to drastically reduce time-to-market while increasing the quality of new vehicles. A key element in the launch process is the body development and manufacturing validation. This step is often a bottleneck and the most costly and thus limiting aspect of the vehicle launch preparation. The high cost in terms of duration and money is due to the intense human involvement in the process as current practice relies heavily on human knowledge and experience with very limited means of evaluating proposed solutions other than actual physical implementation.

Two recent technological advances provide essential building blocks that allow us to move from experience-based to data-driven body development. First, we now have the ability to efficiently create, transfer and store high-resolution digital scans of 3-dimensional parts; and secondly, the integration of Finite Element Analysis (FEA) and dimensional models enables us to predict residual stresses in a functional build assembly. Thus, at this point, suppliers can produce parts using prototype tools and dies and submit scans of these parts to the OEM. The scans of the parts are then assembled in simulation and the resulting sub-assembly or complete car body may be compared with the design intent.

To close the loop in the virtual functional build process, we develop a decision support system (DBDS – Digital Body Development System) that analyzes the virtual product, identifies symptoms of underlying problems in the current design, and proposes and evaluates alternative solutions to the human design team based on past experience and heuristic search. The launch team then has the option of proposing additional solution alternatives or choosing a solution for implementation.

DBDS treats the generation of solutions to problems identified in the current design as a search problem in the high-dimensional space of 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.

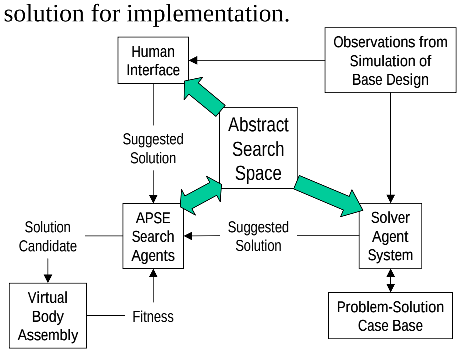

Fig. 1. DBDS performs a parallel heuristic search with human and case-based guidance

In [3] we present an experimental application of our 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. 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.

Given the complexity and massiveness of the search space that DBDS must explore in a given optimization run, we enhance the heuristic of the APSE Search agents to include prior experience and domain knowledge accessible in a problem-solution case base and we enable the human design team to suggest alternative solutions to the search process (Figure 1). We implement a Solver, a multi-agent system that interacts indirectly with the APSE Search agents and that seeks to retrieve solution points (sets of design changes) from 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 been evaluated already by the Search agents.

The remainder of this paper is structured as follows. Section 2 discusses the current practice of auto body development in more detail. In Section 3 we present the DBDS component architecture, which closes the loop in the virtual functional build. Section 4 discusses the adaptation of the APSE Search agents to the decision support problem and Section 5 outlines our swarming approach to solution retrieval. We conclude in Section 6.

2. Current Practice in Body Development

Detailed engineering design of individual parts and components begins after "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 (see Figure 2), whereby individual prototype parts are stamped and then sent to a central location to be assembled into a prototype vehicle body [1]. 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. However, each evaluation is based on a different generation of tooling, so very little information is gleaned on the effects of process variation to the integrity of the entire body.

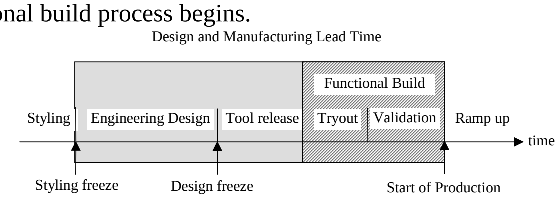

Fig. 2. Vehicle Launch Process.

Two technological trends aimed at improving the dimensional integrity and performance (NVH – or noise, vibration and harshness) of the body are 1) the integration of Finite Element Analysis (FEA) and dimensional models and 2) the scanning of fabricated parts and assemblies for comparison of actual builds with the design intent.

The integration of FEA and dimensional models is significant in that it allows the prediction of residual stress in a functional build assembly. Conceptually, the parts are assembled in the software. Any interference or gap conditions will be accommodated by the assumption that sheet metal is a compliant part. 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.

The recent progress of combining FEA and dimensional models significantly advances the science for understanding the complex interactions between sheet metal parts and the joining processes (usually spot welding). The effects of interference and gap conditions between two mating parts are evaluated based on the amount of part "compliance" that can be expected. Compliance (the bending of parts as they are joined together) can be predicted using FEA, which is also used to predict residual stress in the assembly. The dimensional model can expand that understanding over the expected variation of the fabricating and assembly processes. Together, these two tools can quantify manufacturing capability (fabrication and assembly) and produce a distribution of residual stress as well as dimensional measures in the body.

3. Simulation-Based Decision Support

The Digital Body Development System (DBDS) provides continuous support for the vehicle launch team along the entire iterative functional build process. A single iteration starts at a base design, which comprises scans for all parts that have been produced at this point and CAD-nominals for the remaining parts. The base design also specifies the assembly process as it is currently planned.

In a first step, the base design is "executed" by simulating the virtual assembly of the parts and pre-defined measurements are taken on the resulting product. The design process is completed, if the results meet the design intent. Otherwise, DBDS interprets the output of the measurements as symptoms of underlying problems and generates and evaluates solution alternatives (changes to the base design). It may also invite human engineers to suggest additional solutions. Eventually, the launch team will settle on a solution and implement the corresponding design changes (i.e., change tools and dies, make and scan new parts) to arrive at a new base design.

In today's practice of body development, a large team of experts with diverse background and experience analyze the current design as it manifests itself in the screwbody. Based on their domain knowledge and past experience, individual experts suggest solution alternatives and then discuss their potential merit until the team agrees on a solution. The whole solution generation and selection process is dominated by human knowledge and experience and solutions are chosen or discarded mainly based on hypotheses rather than evidence. Alternatively, DBDS explores the space of possible solutions to the currently observed problems and evaluates each solution alternative by simulating the design that results as the changes proposed by the solution are applied to the current base design. Thus, solutions that DBDS suggests to the launch team are based on evidence provided by the simulation rather than hypotheses.

The simulation-based improvement of a given base design using a heuristic search and evaluation process may be applied to domains other than car body development. To facilitate such a transfer later on, we specify a generic module architecture that makes the specifics of the domain transparent to the optimization process.

The DBDS decision support system has seven modules (Figure 3). The User Interface (UI) module manages the interaction of the system with the human design team. The Solution Generation and Evaluation (SGE) module proposes alternative solutions to solve problems with the base design and evaluates them for their quality and cost. The Change Cost Estimation (CCE) module estimates the cost of actually implementing a particular solution as a change to the base design. The Solution Implementation (SI) module translates a proposed set of changes to the base design into a valid modified design that can be simulated by the VASE. The Virtual Assembly and Simulation Engine (VASE) simulates the "execution" of a given design by virtually assembling parts according to a process description. The Data Preparation and Repository Module (DPRM) manages the large amounts of data generated and used by the DBDS. The Process Controller (PC) module integrates the other modules and manages the data and process flow among them.

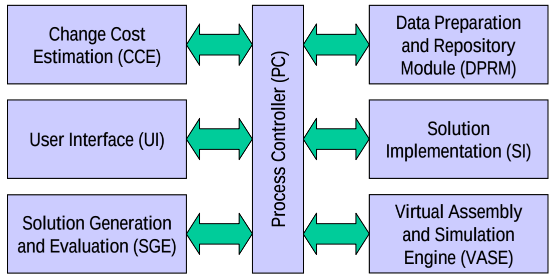

Fig. 3. The Generic DBDS Module Architecture.

Figure 4 illustrates the high-level (black) and low-level (white) process loops facilitated by the PC. At the high level, the user triggers the improvement of a base design, which requires its execution (VASE) and analysis and optimization (SGE). The optimization process generates alternative solutions, which are evaluated (lower-level loops) for their performance (SI, VASE) and cost (CCE).

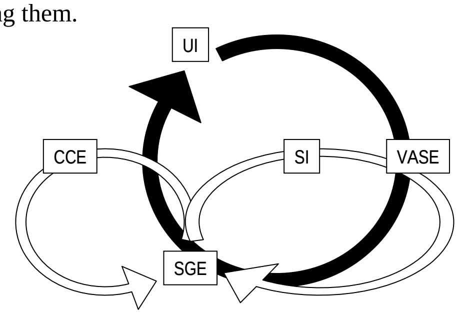

Fig. 4. Inner (white) and Outer (black) DBDS Loops.

4. Heuristic Search for Design Changes

In [3] we present APSE – a multi-agent system that performs a distributed heuristic search through the space of input parameters of a black-box simulation model to find a configuration that maximizes a fitness function defined over observed metrics on the simulation. 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 [6] 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.

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, we treat changes to the base design as input parameters to a black-box simulation and define a fitness function for the search process that measures the degree to which the now modified design meets the design intent.

DBDS is an enhancement of the APSE architecture. While Search agents in APSE are guided only by the fitness of the currently known solution candidates (points in the abstract search space), DBDS provides two additional sources of guidance for the distributed search (see Figure 1). 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 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 (see Section 5).

We integrate these two additional sources of creativity 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 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 5 illustrates the emerging agent trajectory in an abstract search space.

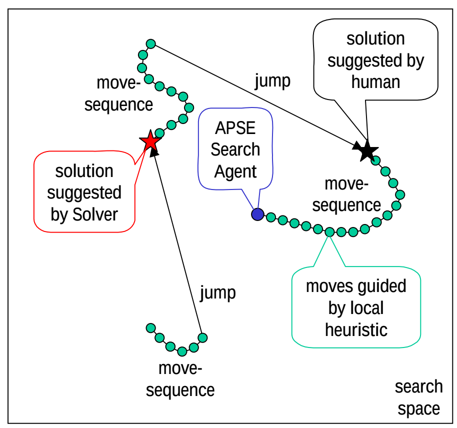

Fig. 5. Agents move and jump through the search space guided by local heuristic, human input, and case knowledge.

The generic distinction between a local improvement heuristic and global jumps to externally suggested solution candidates is open to other solution approaches. Just as 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.

5. Swarming Case Retrieval

Today's car body development process heavily depends on human expert knowledge and experience. With DBDS we create 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 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. We integrate the Solver 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 (Figure 6).

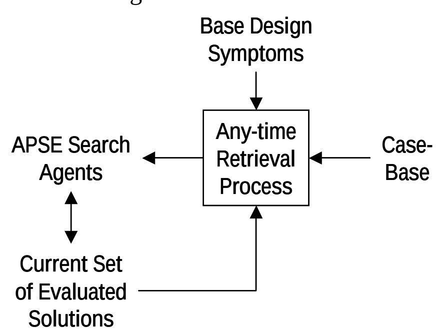

Fig. 6. The dynamic Solver modifies the solution candidates that it suggests to the Search agents based on the progress of the exploration of the search space.

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. Thus, we chose an agent-based any-time approach that continuously integrates changes in the external circumstances without having to restart its reasoning process from scratch.

In the following we discuss details of the operation of the Solver top down. First, we present the adaptive any-time process that manipulates the description of the current problem symptoms to provide a high-quality retrieval of high-performance solutions. Then, we specify the internal mechanics of the fine-grained agent system that drives the adaptive modification of the current problem description.

5.1 Linking Emergent Clustering and Spreading Activation Case Retrieval

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.

Our 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. We organize our case base into a simplified Case Retrieval Network (CRN) [7], 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.

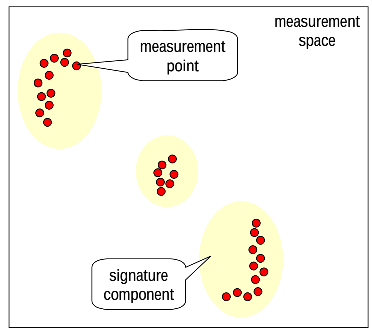

Fig. 7. Clustering of Measurement Points into Signature Components.

Our 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, we may find several disconnected regions along the frame in which our 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 (Figure 7). 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. We select a case probabilistically, based on its current normalized relevance.

The quality of the case retrieval process is high, if there are only one or very few cases with a significant probability to be selected. Otherwise, we may as well select a case randomly from the entire case base. We measure the current retrieval quality with 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. We have used similar entropy measures defined over the current preferences of an autonomous decision maker (here case selection) in previous projects [4, 9] to estimate the current information these preferences actually convey and to subsequently adapt the decision process if necessary.

Figure 8 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.

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 Figure 8). If a solution case is adopted by a Search agent in a long-range jump, 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 to specific arrangements.

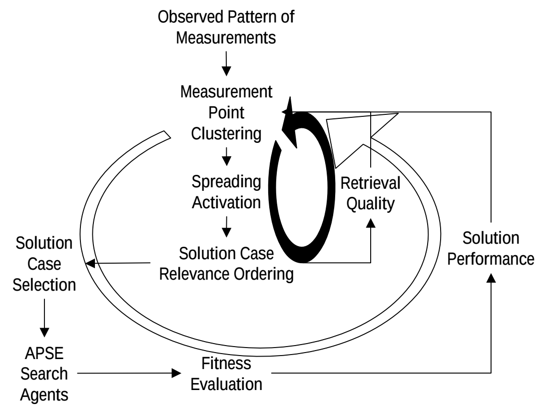

Fig. 8. Adaptive Case Retrieval Guided by Retrieval Quality and Solution Performance.

5.2 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, we may associate points that are related in the process but not necessarily in geometry to the same signature component.

Starting from the original locations of the measurement points, we seek to rearrange the points into arbitrary clusters while trying to keep each point close to its original location. As Figure 9 illustrates, there are a number of possible arrangements that meet these qualitative objectives, because we do not assume a particular number or size of clusters. We design our emergent clustering algorithm to potentially visit all these arrangements (with varying probability) and we use the feedback of the Case Selection Entropy metric and the currently known solution fitness to push the clustering system out of unfavorable configurations.

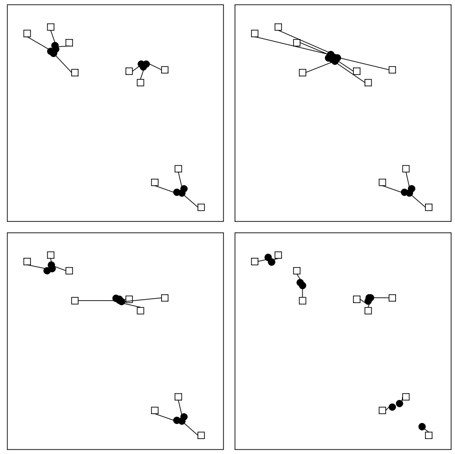

Fig. 9. Possible Cluster Arrangements (black) for the same Original Measurement Points (white).

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 [2], 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 [5]). 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. We successfully applied this approach to create large-scale, self-organizing document bases [10] and we follow the approach in this application too.

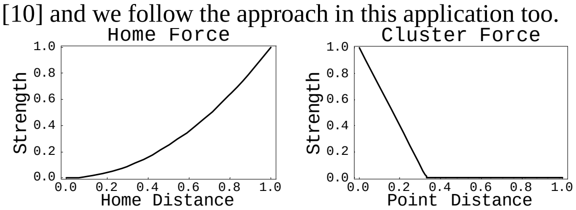

Fig. 10. Forces represent agent objectives in clustering.

In the emergent adaptive clustering algorithm, we assign each point 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 Figure 10) 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 Figure 10), 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.

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 (Figure 11).

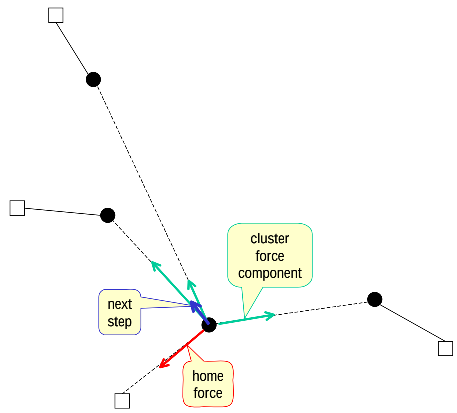

Fig. 11. Iterative Local Force Vector Calculation.

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, we add a small random component to the individual relocation calculation.

We achieve qualitatively different cluster configurations 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 (see Section 5.1).

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, we need to encourage exploration of new configurations over the exploitation of current clusters 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, we propagate the fitness of all cases (zero if not yet explored) 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. We have used a similar back-propagation approach in CRN's to guide the interactive diagnosis of failures in computer hardware [8].

6. 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 [3]. 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.

Tab. 1. DBDS Joint Venture Partners.

Altarum Institute
American Tooling Center
Atlas Tool, Inc.
Autodie International
Center for Automotive Research
CogniTens Inc.
ComauPICO
UGS
Ford Motor Company
General Motors Corporation
Perceptron, Inc.
Riviera Tool Company
Sekely Industries
Thunder Bay Pattern Works

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. DBDS enhances the decision process of the individual agent, who now 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 DBDS is the focus of an ongoing NIST/ATP-supported Joint Venture of more than a dozen automotive, software development and research companies and organizations (see Table 1). The architecture and algorithms reported in this paper are currently being implemented and tested and quantitative results from our first prototype will be forthcoming soon.

7. Acknowledgements

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.

8. References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

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

[7] Mario Lenz, Hans-Dieter Burkhard: Case Retrieval Nets: Basic Ideas and Extensions. in: G. Görz, S. Hölldobler (Eds.): KI-96: Advances in Artificial Intelligence, Springer Verlag, LNAI 1137, 1996.

[8] 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.

[9] 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.

[10] 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.

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