A Multiagent Approach to Evaluating Innovative Component Selection

Carrie Rebhuhn, Brady Gilchrist, Sarah Oman, Irem Tumer, Rob Stone, and Kagan Tumer
Oregon State University, USA

Though there is a clear correlation between profitability and innovativeness, the steps that lead to designing a "creative" product are elusive. In order to learn from past design successes, the impact of design choices on the innovativeness of an entire product must be identified. This problem of quantifying the impact of design choices on a final product is analogous to the problem of 'credit assignment' in a multiagent system. We use recent advances in multiagent credit assignment to propagate a product's innovativeness back to its components. To validate our approach we analyze products from the Design Repository, which contains thousands of products that have been decomposed into functions and components. We demonstrate the benefits of our approach by assessing and propagating innovation evaluations of a set of products down to the component level. We then illustrate the usefulness of the gathered component-level innovation scores by illustrating a product redesign.

1 Introduction

Innovation can revolutionize the way we approach and think about modern designs. Innovative designs can corner or even create markets, and can lead to higher profits for the companies that develop them. Companies that can develop more innovative solutions can have a definite advantage over companies that focus on more incremental approaches to design. But how can design engineers learn to innovate? This is an active field of research, and promoting *creativity* in the early stages of design has shown promise in promoting *innovativeness* in the final product [9]. In this work, we refer to *creativity* as the quality of the ideation process that leads to *innovation*.

There have been numerous studies linking design diversity with the use of solution-generation tools [2, 9]. These tools use design information from a Design Repository to offer component suggestions to fulfill a desired function. But often there is a large selection of components available, offering no indication of whether a component solution has been used in innovative designs or commonplace ones. The designer is left to sort through *all* of the solutions to find the creative solutions. This is not a problem while the Design Repository holds a manageable number of design breakdowns, but because the Design Repository will grow over time it is essential to develop metrics that will allow designers to manage such suggestions.

To increase the potential for creating a market-changing product, we want to focus on evaluating *innovation* in stored design choices. The problem with using innovation as a metric is that it has a complex definition and no common mathematical form. Market success, product rareness, quality, company prevalence, usability, time on the market, and many other factors can impact the perceived innovativeness of a product. Although this quality has a varying definition and no mathematical formula, people can still identify innovation in products. Experts in design and marketing have created lists of identified innovative designs, published in *Time Magazine*, the IDSA IDEA Award, and *Popular Science*.

Innovation within a product can be identified by experts or by popular opinion, but it is much more difficult to quantify the amount that particular design decisions led to the product's innovativeness. Accurately quantifying this parameter for each design decision would allow us to sort design suggestions in such a way that designers would be able to quickly identify and leverage innovative design solutions. Previous work has focused on propagating innovativeness down to the component level through comparative analysis of standard and expert-identified innovative products [9]. In this approach, an 'innovation' parameter was calculated based on the rareness of the component in a population of designs and this was weighted in order to calculate the creativity of a product.

The set of decisions made by the designer and the resulting innovativeness of the product are correlated in some complex manner. But how can we evaluate the impact of each design decision? The field of multiagent systems faces a similar problem, referred to as *credit assignment*. In a cooperative multiagent system, each agent contributes in some way to the performance of all the agents. If there is a multiagent system coordinating traffic, it is easy to measure the performance of the entire system through total throughput of traffic. However it is difficult to quantify the impact of the individual decisions of each traffic-controlling agent on the total traffic throughput. Similarly, it may be easy to identify the innovativeness of a product, but difficult to quantify the impact of each design decision.

One way to handle this credit assignment is by using the *difference reward*. This reward evaluates the contribution of each agent by comparing the performance of the system to the performance of a hypothetical system in which the agent did not contribute. Because the difference reward can more specifically encapsulate the effects of an agent than a system evaluation, this reward has shown increases in learning speed, convergence, and solution quality over using the system reward. This accurate identification of impact is exactly what we need in order to identify the innovative impact of specific components.

The contributions of this work are to:

1. Form a method for the propagation of product-level innovation scores to individual design decisions based on techniques present in multiagent learning.
2. Present a new decomposition of innovation scores based on the component frequency of appearance within the dataset, which we use as the basis for our difference reward.
3. Present results from using this technique with real design data and three separate sets of human-generated innovation scores.
4. Demonstrate the potential use of this new component-level innovation data in a proposed redesign of a product from the Design Repository.

The rest of the paper is structured as follows: in Section 2 we provide an overview of the design repository, previous techniques used to calculate innovation, multiagent systems, and difference rewards. In Section 3 we frame design as a multiagent system, suggest a reward structure for use in this system, and propose two methods of innovation score propagation to the component level. In Section 4 we show results from testing our multiagent learning using several product-level innovation evaluation methods. In 5 we discuss trends in innovation scores and give an example of a creative redesign based on our innovation score propagation. Finally, in Section 6 we discuss conclusions from this research and avenues for future work.

2 Background

This work draws on principles from design engineering as well as multiagent learning. On the design side, in Section 2.1 we explain the general structure and benefits of the Design Repository housed at Oregon State University, and in Section 2.2 we will introduce *novelty*, a metric developed by Shah et. al. [10] for quantification of the uniqueness of a design. On the multiagent side, in Section 2.3 we identify difference rewards as a learning-based method for innovation score propagation.

2.1 Storing Innovation Information

The Design Repository at Oregon State University contains a wealth of information about modern products [4]. It currently contains over 180 products with more than 6000 artifacts. The purpose of the repository is to store previous design information. Products are disassembled into their individual components. Information about each component is recorded such as function, mass, dimensions, materials, manufacturing processes, failure modes. Functional models are created for every product, identifying the way that materials, signals, and energy change as they flow through the product. The functions assigned to the components are done so using the functional basis [7]. This allows for a common language for functionality that is universally understood.

The output from the repository is a morphological matrix containing all possible solutions for a given function (example: "Export") and flow (example: "Electricity"). A morphological matrix is a matrix of component solutions to given functions. This matrix can be used by the designer to select components and assemble them into a complete concept. Within the repository, in addition to the component the frequency that a component solves the given function is also provided. However, the repository does not yet have a way to capture the innovativeness of products and their components. With inclusion of innovative scores, the creative potential of concepts can be determined earlier on in the design process. In order to promote innovation-oriented designs, in this work we develop a mechanism to propagate innovation scores down to the component level.

2.2 Novelty Calculations and Creativity

The attempt to ascribe metrics to creativity and innovation is not a new study, and is a source of recent research in the design community [9, 5**?**]. Quality, quantity, novelty, and variety were identified by Shah et al. [10] to evaluate a set of concepts. In particular, the metric of novelty has been implemented in recent work by Oman et al. [9] along with a weighting system in order to identify the creativity score of particular functions. The metric for novelty we use in this paper is similar to the one used in Oman et al., and is given by the equation:

$$S_{Nj} = \frac{T_j - R_j}{T_j} \tag{1}$$

where S_{Nj} represents the novelty, T_j is the number of designs in the set being examined and R_j is the number of times the component solution for function j is seen in the set. This novelty metric has been traditionally applied only over sets of products with a similar type or goal, however in our work we use it over a set of products of varying types and goals. This may mean that the elements performing functions in one device may not feasibly perform the same functions in another device, but a user may mitigate this potential problem by obtaining lower level functional basis descriptions from the Design Repository.

Previous work has focused on comparison of the functions of 'common' products to 'innovative' products [9]. In this way, functions unique to innovative products along with their frequency of appearance within the design set could be used to characterize their general impact of the innovation within products. In this work, we take this comparison idea and draw on the reinforcement learning concept of a *difference reward*, which has been shown in many domains to effectively quantify the impact that an agent has on a multiagent system [1].

2.3 Multiagent Learning and Difference Rewards

Multiagent reinforcement learning is a field of artificial intelligence that involves the coordination of distributed learners. Control of a complex system may be modeled as a set of less complex interacting entities which make autonomous decisions. These entities are called 'agents', and they interact with their environment through a process of sensing the world state, taking actions available within that state, and receiving rewards. Reinforcement learning in a cooperative multiagent system focuses on finding a set of actions which most benefits the collective system. Learning agents adapt their strategies through repeatedly taking actions and getting a reward for these actions. The Design Repository provides a large database over which to train agents in a *supervised learning problem*, which is a problem where an agent is trained based on expert guidance [3]. In our case the expert guidance is provided by the design examples within the Design Repository.

An agent learns a *policy*, which holds information on an agent's perception of the value of taking a particular action. This policy is updated each time the agent receives a reward for an action it has taken. We perform Q-learning adapted for a stateless game [6], which adjusts the policy according to the rule:

$$V(a) \leftarrow V(a)_{old} + \alpha(R(a) - V(a)_{old}) \tag{2}$$

where $V(a)$ is the expected value of the action, α is the learning rate, and $R(a)$ is the reward received for taking action a. α is a number on the range [0,1] which impacts the learning speed of the agent. Increasing the α parameter may increase the value that an agent puts on more recent information, while decreasing α lowers learning speed but increases the chance of long-term convergence to an optimal policy. Agents develop policies in order to better use their knowledge about the value of specific actions to select their next action. Due to the fact that we cannot evaluate an arbitrary design we constrain our exploration in a different way; we force action selection to match the products we have and then reward accordingly. By forcing the agents to learn by example, we are using a process called *supervised learning*.

In developing the policy of the agent, the reward mechanism used to adjust the policy can be crucial to both convergence properties and the quality of the overall policy developed. Difference rewards have been used in a wide variety of large multiagent systems including the air traffic control, rover coordination, and network routing to promote coordination among agents [1, 11, 12]. This reward works by comparing the evaluation of the system to a system in which an agent is not there, or replaced by some counterfactual which refers to an alternative solution that the agent may have taken [1]. Mathematically this is expressed as:

$$D_i(z) = G(z) - G(z_{-i} + c) \tag{3}$$

where $G(z)$ is the full-system reward and $G(z_{-i} + c)$ is the reward for a hypothetical system in which agent i was removed and its impact was replaced by a counterfactual c. In this work we use difference rewards to compare products with particular creative components to products in which the creative components are replaced by a 'standard' component. This will be discussed in more detail in Section **??**.

3 Framing Product Design as a Multiagent Problem

To create a product, a designer must compose a set of components to fulfill a set of engineering constraints set by the customer. Functional requirements must be fulfilled by the components selected. Essentially, design may be broken down at a rudimentary level to function-satisfaction by selection of components; a process where an agent must select components to fulfill properties desired by the engineering requirements of the design. Our approach to product design is structured using three primary factors: products, functions, and components.

1. *Products*: Products represent a set of actions taken by a designer. Products feature a set of functions which they must perform, and a set of components which have been selected to satisfy these functions. We also assume that products have an associated innovation score. The creation of a product where the design process is modeled as a multiagent system represents the *joint action*, or set of all actions taken by agents, in a multiagent system.
2. *Functions*: The purpose of a product is to perform a set of functions. Because a product must perform a prescribed set of functions, this defines the requirements for the component-selection of the designer. In a multiagent system, the task of ensuring that these functions are satisfied in the creation of the product is given to the *agents* within the system.
3. *Components*: Components within a design represent design decisions, and can be used to perform one or multiple functions within a design. In a multiagent framework, the set of components represent actions available to agents trying to fill functions.

In a step toward an autonomous design strategy, we assign our agents the task of *component selection*, which is to select components to satisfy a particular design requirement (function). One agent is assigned to select a component to fulfill each type of function (i.e., to import electricity) and has an action space which includes all components which could possibly accomplish this task (i.e. a battery or a power cord).

We use two datasets that include real designs from the Design Repository to train our agents, using the function-component matrices of these designs to guide an agent to take an action while designing a product. We then evaluate the designs using the product-level rewards given by these datasets, and reward the actions using two separate methods, which will be explained in more detail in Section 3.1. Agents then performed a value update (Equation 2) with a learning rate of 0.5 using this reward. Policy values were initialized to the novelty score at the beginning, as we assume novelty does have some impact on the innovative impact of a component.

To use the difference reward decomposition as given by Equation 3, we need an equation for the system score of a device that we can decompose. In Section 3.1 we assume that we begin with a product-level evaluation of the creativity $G(d)$ which is given by our experimental dataset (explained further in Section 3.2) and then formulate a decomposition of $G(d)$. Then we use this formulation to develop an equation with which we can derive a difference objective. We then present two methods of application of difference rewards to our dataset. In Section 3.2 we present the formulation of our three datasets with which we train our agents.

3.1 Product-level Innovation Score Decomposition and Difference Reward Derivation

Novelty scores have been used in previous work along with a hand-tuned weighting in order to assess the innovation score of different components. From this fact, we postulate that $G(d) = f(S_N)$, where $G(d)$ is the total innovation score of a device, and $f(S_N)$ is some function of the novelty of all of the components within the device. As a starting point, we assume that the innovative impact of a component is proportional to its novelty score. This allows us to decompose the product-level innovation score down into component impacts I_i:

$$G(d) = \left(\sum_{i=1}^{n} \frac{S_{N_i}}{S_{N_{all}}} \right) G(d) = \sum_{i=1}^{n} I_i \tag{4}$$

where d is a design with n different functions, I is the impact of component i on the system score of the design, $G(d)$ is the full score for that design (recall that this is given by our data), S_{Nall} is the sum of all novelties of the components, and S_{Ni} is the novelty score for component i.

We assume that we know the global score. Therefore this decomposition does not add information, but instead allows us to derive a difference reward which represents what would happen if one component was taken out and replaced by another component using these impacts I_i. The difference reward for this system can be derived by using the $G(z)$ formulation in Equation 4 in Equation 3, with some replacement component represented by the counterfactual term c. Making this substitution and simplifying yield a derivation of the difference reward:

$$D_i(d) = \left(\frac{S_{N_i} - S_{N_c}}{S_{N_{all}}} \right) G(d) \tag{5}$$

where S_{N_c} refers to the novelty score of the component which has hypothetically replaced the original component in the design. In difference rewards, the counterfactual term c can be defined in a variety of ways. In our domain, we employed two counterfactual reward formulations.

The first counterfactual used the most common component solution (i.e. the lowest novelty score), and we refer to the difference reward formulated using this method as $C_i^{pair}(d)$. This was applied to reward the agents for all component solutions based on the design. This is given by the equation:

$$C_i^{pair}(d) = \left(\frac{S_{N_i} - S_{N_{least}}}{S_{N_{all}}} \right) G(d) \tag{6}$$

where $S_{N_{least}}$ refers to the novelty of the most commonly-used component that fulfills the agent's assigned function.

The second counterfactual used all other component solutions as comparison. This meant that the set reward, $C_i^{set}(d)$ was recalculated $k-1$ times, where k refers to the number of different components that an agent had in its action space. The agent performed a value update $k-1$ times with each difference reward. The equation for this innovation evaluation was therefore given by:

$$C_i^{set}(d) = \left(\frac{S_{N_i} - S_{N_{alt}}}{S_{N_{all}}}\right) G(d) \tag{7}$$

where $S_{N_{alt}}$ refers to the novelty of an alternative component seen in the set of products. This is applied over all alternatives.

The table of values resulting from these calculations represent the innovation score estimation of the function-component pairs found in the set. These values are the agent's estimation of the creative impact of the function-component pair on the product-level design score, and are used to present our findings in the two datasets in Section 4.

3.2 Training Data

The innovation score cannot at this time, from the data available in the Design Repository, be objectively calculated for a product. Though we cannot objectively calculate innovation, it is a quality that *humans* can readily assess. There are three different ways that we gather data on the innovation of different devices: expert innovation identifications found in consumer magazines, a survey of several college students, and a latent variable analysis using data from design engineering students. These are explained here in further detail.

Expert Data: Binary identifications of *innovative* products (innovative score=1) were taken from *Popular Science*, the IDSA IDEA Award, and *Time Magazine*'s 50 Best, and contrasted with hand-selected *standard* products (innovative score=0).

Survey Data: To make up for the binary nature and small size of the expert dataset, a survey was conducted involving 10 participants rated a series of 50 designs on innovativeness, with 5 levels of innovative scores. The average of this data was then taken, and this was used as a product-level score for the designs.

Latent Variable Data: We performed a latent variable analysis across 8 products. The entire dataset of 156 responses consisted of undergraduate engineering students who were taking a mechanical engineering design class. Each student responded to a set of questions which were targeted at identifying the impacts of three latent variables: innovation, product usability, and company profile.

4 Results

We divide the results generated with our multiagent system using the Design Repository Data by the external evaluation method used. The evaluation methods vary in products evaluated, type of innovation information, and demographic. Using the two difference reward formulations (D^{least} and $D^{average}$) and three datasets (expert, survey, and latent variable), we perform six different experiments exploring these parameters. Reward order impacts the results of reinforcement learning, so we perform experiments using a randomized order and take the average across 30 runs. In Sections 4.1-4.3 we identify trends in the data. In Section 5 we analyze the trends across all datasets.

4.1 Expert Evaluation Dataset

As a baseline for our evaluations we analyze the novelty. As shown in Figure 1, we plot the novelty of function-component pairs versus the average scores of products in which the function-component pair is found. Plotting the novelty against this average score solidifies a key hypothesis in the beginning of this work; *the frequency with which the component appears does not solely reflect the creative impact it has*. Novelty scores tend to be higher at the extremes of the average product-level score, with slightly lower values toward the center of the spectrum, but this correlation was weak in this data.

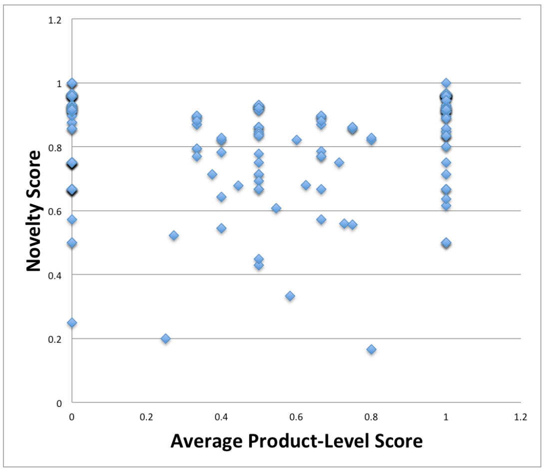

Figure 1: Novelty scores for the function-component pairs found within the expert evaluation dataset.

Figure 2 shows the results from our multiagent system method of evaluating the innovation at the component level. The data show an upward trend with two major outliers. The outlier shown at (0.750, 0.035) is

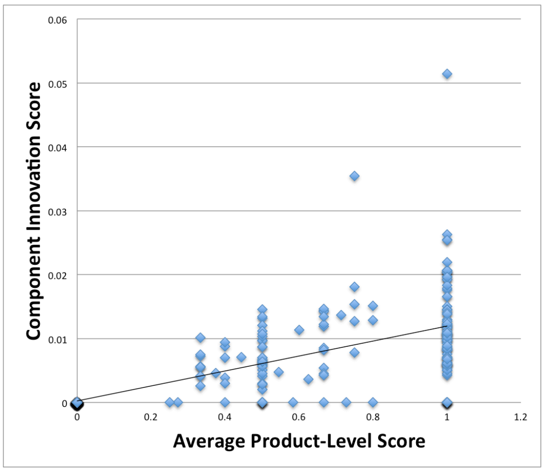

Figure 2: Agent-estimated values using the D^{least} reward on the expert evaluation dataset.

the *channel material, reservoir* function-component pair, while the outlier shown at (1.000, 0.051) is the *control magnitude material, screw* function-component pair. Both of these show particularly high estimations of contribution to the innovation score. Innovation scores increase with average product-level scores, but also increase in spread with the product-level average.

The interesting part of this data is not necessarily in the trends, which will tend to increase with the average product-level score because of the learning mechanism. The outliers give a better indication of innovate component usage. In Figure 2 there are two major outliers: the datapoints corresponding to *channel material, reservoir*, and *control magnitude material, screw*. While these components do not appear innovative in and of themselves, they may serve a particular function within their design in an innovative way.

Figure 3 shows much less correlation with the average product-level score than Figure 2. This is likely due to the fact that the $D^{average}$ difference reward has an average-novelty counterfactual, and therefore will tend to have an equal number of points which are positive and negative. Most of the positive data are collected at the higher end of the average product-level scores. The data show a slight upward trend with have a wide variation in values.

4.2 Survey Average Dataset

The novelty scores for the survey average dataset (Figure 4) have high novelty scores at the extreme values of average product-level score, accompanied by lower levels of novelty at the lower product-level scores. The data show the diversity of the scores found in this dataset, as there

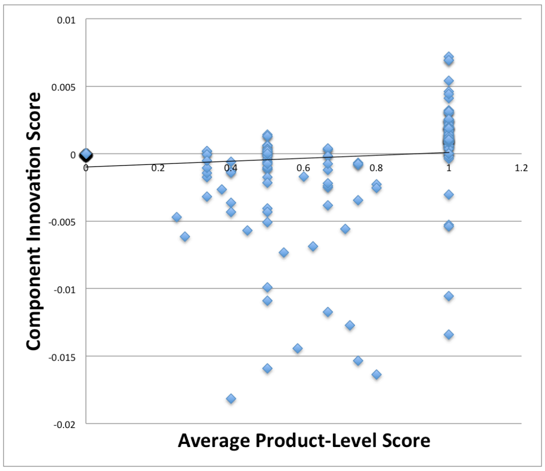

Figure 3: Agent-estimated values using the $D^{average}$ reward on the expert evaluation dataset.

is much less striation in the results as compared to the expert evaluation dataset. The novelty shows a general trend toward being high, but shows a dip on the range [0.05, 0.45].

The distribution of the data suggested that there may have been *more* data in the middle of the distribution, but it remains that, apart from a small number of outliers, the novelty of components both in highly-innovative and fairly plain products tend to be higher. Conversely, there seems to be a loose trend that the novelty of components at the lower-middle range of the average product-level score tend to be *lower*, indicating that products made from frequently-seen components may have a lower product-level score on average.

The innovation scores shown in Figure 5 show a clear trend; as product-level score increases, the component-level innovation both tends to increase and polarize. One particularly highly-estimated function-component pair at (0.600, 0.024) corresponds to the *convert energy, handle* function-component pair. At the lower levels of average product-level innovation, there tends to be a tight fit to the data. The variance in the data appears to increase with the average product-level score. At an average product-level score of 0.4, we see a dataset which disperses almost completely.

The innovation scores shown in Figure 6 once again demonstrate the fact that roughly half of the function-component scores will score negatively. A trend which correlates somewhat with the novelty scores for this dataset (Figure 4) is shown in the data; function-component pairs which appear at the mid-level of average product-level innovation tend to *detract* from the design of a component according to our estimate rather than help it.

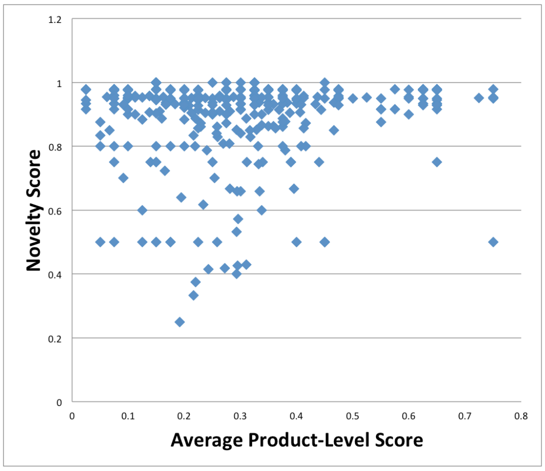

Figure 4: Novelty scores for the function-component pairs found within the survey dataset.

4.3 Latent Variable Dataset

The latent variable dataset provides us a unique chance to test the multi-agent system's response to negative product-level values. Scores in both our expert evaluation dataset as well as our survey dataset provide a training set which is bounded on the interval [0,1].

As shown in Figure 7 the novelty within this dataset has the most pronounced trend in having higher novelties at the extrema and lower novelty scores toward the center. This data was almost triangular in shape, and further supports the trend across all datasets that novelty correlates to extreme high or low average product-level score. This is particularly pronounced because all negative average product-level scores have a relatively high novelty. Figure 7 also shows a certain amount of discretization which also exists in the expert data. The data are not striated so much as simply sparse, an therefore these discretizations are likely less due to the scoring mechanism and more due to the fact that only 8 products are included in the analysis.

Figure 8 shows that as the average product-level score increases, the innovation score assigned to the components tends to increase as well. There are two outliers in this data located at (5.9, 0.209) which represent the function-component pairs *channel energy, em sensor* and *channel energy, hydraulic pump*. These data are somewhat different from the data found in Figure 5 in that the innovation score peaks prematurely in relation to the product-level score. The tendency for the data to spread out as product-level score increases, but this trend is not pronounced. The results shown in Figure 9 are trends toward a zero average. This data also shows a trend which has been observed in the other results, which is that it tends to spread out as the average product-level innovation in-

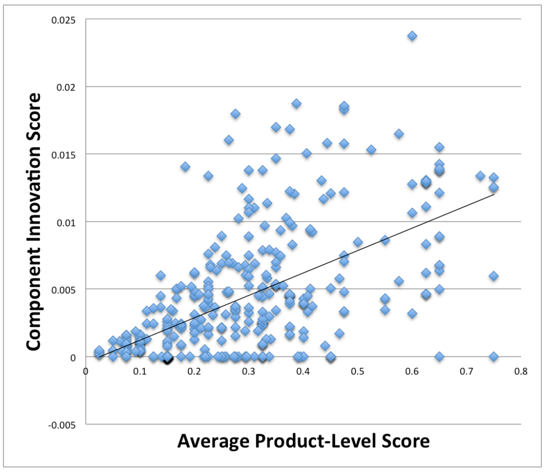

Figure 5: Agent-estimated values using the D^{least} reward with the survey dataset.

creases. The data in Figure 9 also shows several outliers which are worth mentioning.

The point with the highest estimated innovation score is located on Figure 9 at (9.86, 0.073) and corresponds to *signal signal, circuit board*. What this is saying is not that products within this dataset tend to be more innovative if they *have* a circuit board, but that they tend to be more innovative if they *transmit a signal* using a circuit board.

So what do negative outliers mean? There are a group of three negative outliers obvious on Figure 9. Interestingly, two of them also involve a circuit board, but it is used in a different manner. The points at (5.94, -0.145) and (5.91, -0.093) correspond to *channel energy, circuit board* and *channel signal, circuit board* respectively. The fact that these have a different effect not just based on the component involved but also how it is used suggests that innovation may come from a creative application of a component rather than necessarily identifying a component as creative. Channeling energy and signals are an integral function of a circuit board, and therefore will appear wherever there is a circuit board in the design set. They perform a functional role but do not contribute to the innovation of a device. A circuit board that transmits a signal indicates that some kind of display was found on the product analyzed. A designer attempting to leverage this suggestion might attempt to incorporate a graphical display on a new product design.

5 Analysis of the Results

Though there are differences in the source of the innovation scores as well as the method by which they are gathered, there are trends seen across all

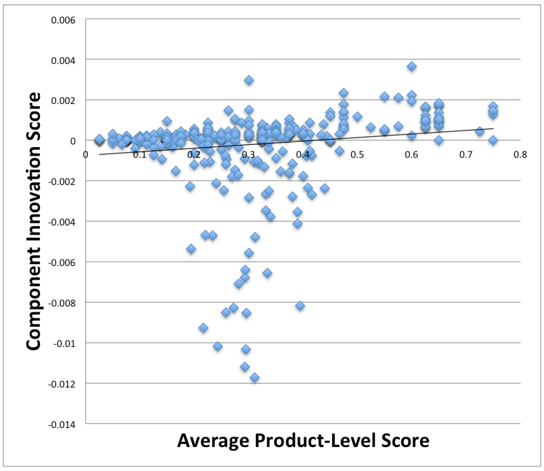

Figure 6: Agent-estimated values using the $D^{average}$ reward with the survey dataset.

datasets which reflect some of our key findings in this work. In Section 5.1 we identify these trends and offer some observations about the data. However the main focus of this work is to develop a tool for application to design. Thus in Section 5.2 we demonstrate how to use this information to perform a redesign on a well-known product to promote higher innovation.

5.1 Trends in Innovation Scores for Components

Results across all datasets show a dip in novelty scores as well as a spreading out of the innovation scores as the product-level score increases. Novelty is too dispersed throughout the dataset to draw definite conclusions, but it appears that higher novelties in all cases tend to appear at the extreme values of average product-level score.

The innovation scores for function-component pairs obtained using D^{least} across all datasets showed an upward trend, generally learning that components in scores with a higher average product-level score had more innovation value than the those found in products with lower scores. Conversely, the scores found by $D^{average}$ do not necessarily show an upward trend with the increase in average innovation score, but instead show the same dispersal of the data. This indicates that the difference reward is able to identify what, in products with high scores, is both contributing and not contributing to those high scores.

Outliers in this data can give us some insight into components that are particularly influential in promoting innovation. Through an analysis of the outliers in the Latent Variable dataset using the $D^{average}$ scoring, we were able to identify that the presence of a component does not necessarily correlate with a better novelty score--how it is *used* contributes

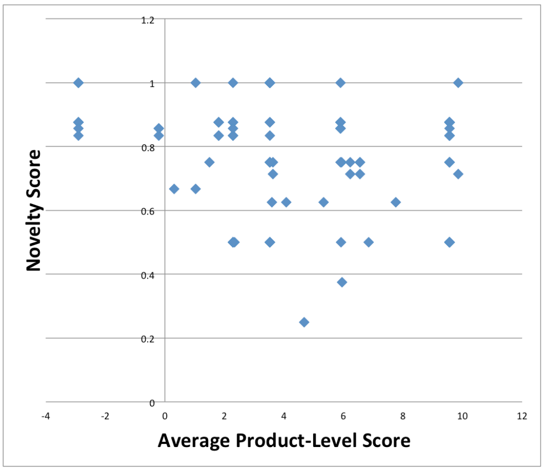

Figure 7: Novelty scores for the function-component pairs found within the latent variable dataset.

in large part to this score as well. Looking at the outliers, the results may have been influenced in part by the audience analyzing the dataset. We found that in the expert evaluation dataset we had two outliers which were the pairs *channel material,reservoir* and *control magnitude material,screw*. These do not seem like particularly innovative components, but the innovation metrics in this case were derived from consumer magazines, which likely had a level of functionality and market influence. The function-component pairs found to be particularly innovative also are highly *useful* in a final design, and this usefulness may have played a part in their identification as 'innovative'.

The outlier identified in the survey data using $D^{average}$, *convert energy, handle* was also underwhelming. It was identified as having the highest novelty score, but this may say more about the population surveyed than its true innovation contribution. Again usefulness comes into play here; 'innovation' and 'usefulness' are hard to distinguish from one another in many cases, and the survey-takers were not design engineering students. When they were shown the designs, it was as a picture with a brief explanation, and not a full breakdown of how the product works. A handle is an easily-seen component which would show up in a picture, and would add to the usefulness of a design.

The latent variable data outliers, which *were* gathered from design engineering students, show more interesting function-component pairs identified as innovative. These were the pairs *channel energy, em sensor* and *channel energy, hydraulic pump*. These identify components that have a higher technological complexity than the outliers of the other surveys, and therefore are potentially more interesting to design engineering students. It is likely that the design engineering students were able to better understand the functionality of a product, and therefore find it more

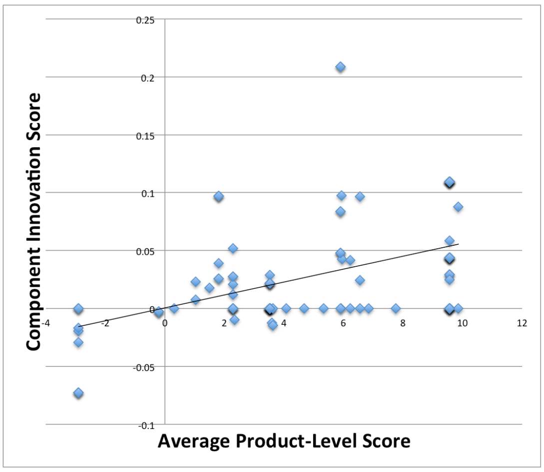

Figure 8: Agent-estimated values using the D^{least} reward on the latent variable dataset.

innovative than someone who had only a functional or aesthetic knowledge of a product.

The influence of demographics on training data may also be a parameter that can be leveraged. Innovation is a subjective measure, and companies cannot design a product with all demographics in mind. Products need a market. If the demographics of this market can be used to bias the training data for our multiagent system, we may obtain better evaluations of component innovation may be obtained for the target market.

5.2 Implication for Generating New Designs

Though we can demonstrate patterns in the data using our techniques, an example in an actual design application provides a more intuitive look at what the different techniques actually offer. For this reason, we present a redesign of the product which had the lowest innovation ranking according to our survey: the Dustbuster. We selected five functions in the Dustbuster with the lowest product-level average score ratings and calculated suggested replacements for the components according to the highest-ranked component solutions as discovered by our different techniques on the survey dataset. The results are shown in Table 1, which can be compared with the original design given in the 'Repository' column.

The assessment of the innovation of the different replacements for the design is difficult to perform objectively. Additionally, as practicality is separated from the innovation, not all suggestions are necessarily optimal or even possible to implement. Nonetheless, all suggestions offer a designer a different perspective on how to modify a vacuum. The novelty evaluation suggests using a hydraulic pump, which is rarely if ever present in vacuums and may offer an interesting perspective on how to

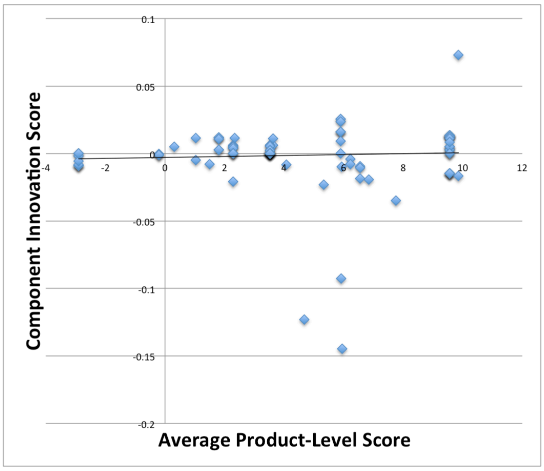

Figure 9: Agent-estimated values using the $D^{Average}$ reward on the latent variable dataset.

creatively channel material sucked into the vacuum. The D^{least} evaluation suggests that a speaker replace the electric switch to turn on the vacuum, which indicates a redesign featuring a voice-activated vacuum cleaner. The $D^{average}$ evaluation suggests that guiders may be used to replace the electric switch, suggesting that the vacuum might be designed to mechanically show when it is full. The material coming into the vacuum might brush past the guiders and put pressure on them which would activate another part of the design to detect fullness of the vacuum cleaner.

Though the two approaches have somewhat different results in their evaluation of innovativeness, they are equally valid approaches. The counterfactual term used in the difference reward for D^{least} will provide consistently positive evaluations for the innovativeness of devices, whereas $D^{average}$ provides approximately half negative evaluations. This may represent two possible ways of looking at the data; either the presence of a standard component does not help the innovation score, or it may be perceived to *bring down* the innovation score because most of the other options might contribute more meaningfully to the product's innovativeness.

Ultimately the usefulness of the design suggestions is still in the hands of the engineer, but the suggestions based on D^{least} and $D^{average}$ appear to offer more interesting solutions overall to the given functions. They do not appear to be *practical*, necessarily, as many of the suggestions do not present feasible design decisions. This draws attention to the fact that 1) we are using the most abstract language in the design repository, which may mean that components designed to handle certain specific tasks may not be applicable for all functionalities and 2) we need some sort of method of identifying which components are *possible* to perform these functions. This makes MEMIC [2] the ideal complement to this assessment technique, as its target is to identify component suggestions from designs which have had similar functional flows, and therefore this increases the chance that components may have similar functionality.

Table 1: Dustbuster original design components and redesign suggestions.

Function	**Repository**	**Novelty**	D^{least}	$D^{average}$
convert energy to signal	electric switch	light source	speaker	guiders
control magnitude energy	electric cord	washer	abrasive	coupler
channel energy	electric cord	abrasive	pulley	coupler
channel material	guiders	hydraulic pump	friction enhancer	shaft
convert energy	electric motor	cover	magnitude controller	handle

6 Conclusions

We have developed a method for using difference rewards under a multiagent framework to propagate product-level innovation scores down to the component level. By framing the design process as a multiagent component-selection process with functional requirements modeled as agents, we are able to have our agents learn scores for component solutions after seeing several example designs. From the data gathered in this work, we can draw three conclusions; i) we identify trends in the novelty relating to our average product-level innovation score, validating our approach; ii) we can use this method to identify components that both add to and detract from the innovation score of the device; iii) we can use our evaluations to perform design alterations and give an indication of which components have historically increased innovation for a particular function.

Our first conclusion is validated by the tendency for the novelty scores to polarize, and typically decrease for mid-level average product-level scores. Novelty *does* relate to the innovation score of the device, but it is not directly proportional to innovation score. The trend toward a dip in novelty scores also uncovers a major shortcoming in our attempt to identify innovative components; the more common components may still receive a mid-level innovation score, but this depends on the *configuration* of components rather than the components themselves. This accounts for solutions in which common components are assembled in a creative configuration. We also demonstrate that in the high levels of average product-level innovation scores, more novel components are more frequently seen. We also see novel components in the lower levels of innovation--this is most likely due to devices which perform only one task uniquely. For example a microwave heats objects in a way not found in any other product, but it only performs one task and therefore is not particularly innovative.

Our second conclusion best highlights the intent of developing our difference methods for innovation identification: in innovative products, there tend to be components that add more to the innovation of the device than other components. This comes with the parallel observation that some components actually *detract* from the innovation of a product at higher product-level innovation. This invites the use of this method for design improvement toward higher levels of innovation, as we can identify which components help and hinder the innovation of a product and modify it accordingly. The fact that we can identify components within innovative products that have a large positive or negative effect on the innovation of a design suggests two interesting findings: one, that the innovation of a product may be carried primarily by only a couple of components within the product, and two, that highly innovative products must rely on tried-and-true methods of performing functions in order to have a fully operational product. Because there are so many components that have negative scores under our D^{least} evaluation in the higher levels of average product-level score, this indicates that perhaps the most innovative products do *one* thing innovatively, and they perform other functions in a traditional and therefore reliable manner. This is consistent with the previous research done using functional subtraction [8], which identified functions existing in innovative components which did not exist in the common components.

Our third conclusion was validated by our redesign of the Dustbuster as given in Section 5.2. By using the data output by the multiagent system, we were able to demonstrate the fact that the learned data provides an accessible way of obtaining design suggestions and their associated innovation scores. We also identified the fact that bias in our datasets plays a large role in what components will be considered the most innovative; both the products and the people evaluating them were different across the datasets, which introduced bias not only from the backgrounds of people, but by the different novelty scores for components in a different dataset. There may be a way to leverage this and target a more specific demographic and design subset, but this is not explored in this work.

There is currently no standard for how close our innovation calculations are to the true innovativeness of the components. The *usefulness* of our measurements may be measured in the same way that the usefulness of concept generation techniques have been measured and validated. We have developed a tool which is intended to inspire creativity in engineers who use it. Though this technique is theoretically sound from a multiagent perspective, the actual usefulness in introducing innovation into the design process must be decided by the engineers who use it.

Acknowledgments

The authors would like to thank the National Science Foundation for their support under Grant Nos. CMMI-0928076 and 0927745 from the Engineering Design and Innovation Program and the NSF REU supplement grant CMMI-1033407. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators.

References

1. A.K. Agogino and K. Tumer. Analyzing and visualizing multiagent rewards in dynamic and stochastic domains. *Autonomous Agents and Multi-Agent Systems*, 17(2):320--338, 2008.
2. C. Bryant Arnold, R. Stone, and D. McAdams. MEMIC: An Interactive Morphological Matrix Tool for Automated Concept Generation. In *Proceedings of the 2008 Industrial Engineering Research Conference*, 2008.
3. Andrew G Barto and Thomas G Dietterich. Reinforcement learning and its relationship to supervised learning. 2004.
4. M. Bohm, J. Vucovich, and R. Stone. Using a design repository to drive concept innovation. *Journal of Computer and Information Science in Engineering*, 8(1), 2008.
5. David C Brown. Guiding computational design creativity research, 2008.
6. Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 746--752. AAAI Press, 1998.
7. J. Hirtz, R. Stone, D. McAdams, S. Szykman, and K. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13(2):65--82, 2002.
8. S. Oman, B. Gilchrist, C. Rebhuhn, I. Tumer, A. Nix, and R. Stone. Towards a Repository of Innovative Products to Enhance Engineering Creativity Education. In *Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information Engineering Conference IDETC/CIE 2012*, Chicago, Illinois, USA, August 2012.
9. Sarah Oman, Brady Gilchrist, Carrie Rebhuhn, Irem Y. Tumer, Anthony Nix, and Robert Stone. Towards a repository of innovative products to enhance engineering creativity education. August 2012.
10. J. Shah. Metrics for measuring ideation effectiveness. *Design Studies*, 24(2):111--134, March 2003.
11. K. Tumer and A. K. Agogino. A multiagent approach to managing air traffic flow. *Journal of Autonomous Agents and Multi-Agent Systems*, 2010.
12. D. Wolpert and K. Tumer. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research*, 16(2002):359--387, 2002.