Jacket's Garage: Software for a Physical Science Learning Environment

Christopher W. Kramer, Janet L. Kolodner
Email: cwkramer@gatech.edu, jlk@cc.gatech.edu

Abstract: The Learning by Design lab has been developing software and curriculum to support learning in the context of design challenges. We have used software to guide learners through design and investigation practices as well as help them describe their scientific observations. In this paper, we present easy-to-use software tools that support learners in their construction of robust science conceptions. We investigate how multiple representations such as diagrams, graphs, and animations can provide support for knowledge construction. In addition, we investigate how explanation templates scaffold learners' interpretations of science. We have combined these tools into one software environment, Jacket's Garage.

1. The Problem

Science education research tells us that understanding science and effectively communicating scientific knowledge is hard. During the 1970's researchers began to listen to what students had to say in science classrooms. Surprisingly, they found that students often had ideas about science that conflicted with what instructors were lecturing about (Smith, diSessa, Roschelle, 1993). Studies in physics classrooms led to research on how students form mental models and misconceptions of the physical world (e.g., diSessa, 1993; Clement 1983). Andrea diSessa, by asking students to discuss physics in everyday situations, wanted to discover how students formed "*senses of mechanism* – a sense of how things work, what is likely, possible, or impossible" (diSessa 1993). He discovered that even college physics students had elaborate misconceptions, and some students even provided justification that directly contradicted basic physics principles. Our goal is to help middle school students form mental models that represent how physics actually works; in other words, help them develop correct *senses of mechanism.*

The Learning by Design lab has been developing curriculum units and software to help middle school students understand physics concepts in the context of design challenges. Learning by Design (LBD) challenges learners to design artifacts to meet defined goals in the context of "engaging design-and-build activities that enliven students' interest in science" (Kolodner, Crismond, Gray, Holbrook, Puntambekar, 1998). The software we have designed adds to design experiences by allowing learners to run experiments and explore effects of variables, and it guides them through explaining the behaviors of their investigations. From pilot studies (e.g. SIMCARS, Vattam & Kolodner, 2006; SHADE, Vattam, Kramer, Kim, Kolodner, 2007) we found that when we provide explanation templates, learners effectively communicate their scientific knowledge using causal explanations. However, our software was limited. Our goal, therefore, has been to design and build more complete and robust software tools that would be aesthetically pleasing, easy to navigate, and more completely help learners build robust conceptions of the physical world.

In this project, we take an approach to software simulation design using multiple representations. We predict that using multiple representations within simulations can help learners approach a problem from many different directions. Research suggests that using multiple representations, such as diagrams, is a positive step towards getting learners to understand and solve physics problems (e.g., Ainsworth, 1999; Larkin, 1983; Rosengrant, D., Van Heuvelen, A., & Etkina, E., 2005). In addition to multiple representations, we have explanation construction tools in our software to provide support for learners to communicate scientific ideas. These features are part of Jacket's Garage, which uses hovercrafts to explore the science of forces and motion.

2. Background

Learners have difficulty understanding physics and instructors have a hard time teaching it for two reasons. First, learners are not *tabulae rasae* - clean slates - when they walk into a classroom (Smith et al., 1993). They already have ideas about how the physical world behaves before an instructor ever shows them an equation or theorem. Much of the time, students' ideas do not relate to the abstract equations they are seeing on chalkboards. In our case, the physical properties of hovercraft flight are extremely difficult to understand. A hovercraft lift fan crams molecules of air underneath a hovercraft hull. As a whole, these molecules act like an invisible fluid, and the hovercraft balances on top of this cushion of air. It is very difficult for learners to investigate something invisible. Making explanations about invisible phenomena is even more difficult.

Second, keeping learners' attention and keeping them motivated is difficult; "too often, science instruction fails to engage students' interests and is divorced from their everyday experiences" (Kolodner, Camp, Crismond, Fasse, Gray, Holbrook, et al., 2003). However, LBD "provide opportunities for engaging in and learning complex cognitive, social, practical, and communication skills" (Kolodner et al, 1998). These opportunities provide learners with the experience of using science to test their conceptions and debug their knowledge.

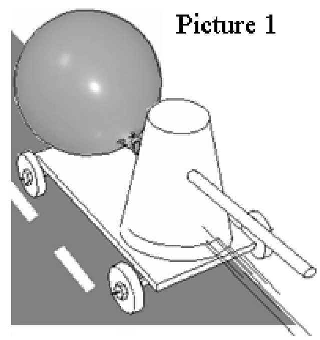
In this paper, we discuss the research that has led us to design new software, Jacket's Garage. First, we summarize LBD theories in practice and discuss how we have developed Jacket's Garage around LBD methodology. Second, we discuss what it means for learners to construct good senses of mechanism and we give examples of how multiple representations can provide students with the ability to address their conceptions and construct new knowledge (e.g., Ainsworth, 1999; Rosengrant, D., et al., 2005). We finally discuss what we learned from earlier software implementations about helping learners to construct correct senses of mechanism and scientific explanations.

2.1 Managing Engagement: Constructionism & Learning by Design

Constructionism can be simply defined as "the idea of learning-by-making" (Papert, 1991). However, Papert goes on to say that constructionism is "…much richer and more multifaceted, and very much deeper in its implications, than could be conveyed by any such formula." Constructivism, not to be confused with constructionism, is the theory that knowledge is constructed. Constructionism is an application of this theory, which requires that "you engage in experiences liable to encourage your own personal construction of something in some sense like it" (Papert, 1991). However, simply designing and constructing is not enough for all learners to learn from. Cognitive science literature (e.g. Collins, Brown, Newman, 1989) tells us how important referring back on learners' experiences is for promoting learning. LBD is problem-based learning in the context of design challenges that is carefully crafted to promote the kinds of reflection on design experiences needed for deep and lasting learning (Kolodner et al. 2003). Jacket's Garage centers around the LBD approach to understanding science and *Vehicles in Motion* is an example of a LBD unit.

2.1.1 Vehicles in Motion: An Example

Vehicles in Motion (VIM) asks students to design and build miniature vehicles to learn about forces and motion. One vehicle they design has: a chassis, four wheels, two axels, and a propulsion system consisting of a balloon and straw (*see picture to the right*). Learners first have a "messing about" session where the instructor provides them with different materials and the instructor encourages them to put the pieces together. Next, they begin investigating to answer questions raised while messing about. During this time, students discover the factors that affect a car's motion, for example, using three straws in their propulsion system instead of one. They share these ideas with the class and attempt to explain their findings. For instance, a learner may explain that three straws lets more air out of the balloon at once and therefore the car will travel faster down the hallway. They then begin an iterative design and building process. They improve upon their designs and rebuild and retest their vehicles until they have achieved the design challenge (e.g. a certain distance down the hallway), all the while explaining, using evidence from experiments and science knowledge from readings, why the car behaves as it does.

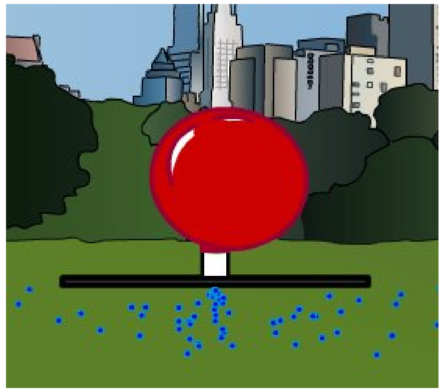


2.2 Promoting physics understanding: Multiple External Representations

Andrea diSessa (1993) has developed theories of learners' internal representations of the physical world; he names these representations *senses of mechanism*. A *sense of mechanism* is how someone thinks the physical world works. For example, someone may have an intuitive *sense of mechanism* of the motion of objects - kinesthetic sense of agency. He defines these *senses of mechanism* as complex knowledge structures, and at the heart of these structures are *phenomenological primitives, p-prims* for short. A *p-prim* is an atomic unit for diSessa's knowledge structures; in other words, a *p-prim* "…may be self-explanatory – something happens 'because that's the way things are'" (1993).

To provide an example, *force as a mover* is a *p-prim* that arises from the observation of pushing an object and watching it slide across a table in the direction of the push (Clement, 1982). The object eventually stops and a naïve observer may claim that the object has stopped because in order for the object to stay in motion, the force on the object must persist. Conversely, the expert observer sees friction between the table and object as a force, and the object slows down because this force of friction is acting in the opposite direction of motion. We suggest that using multiple representations during simulations can help learners debug their *p-prims* and construct correct *senses of mechanism.*

In Multiple External Representation theory (MER; Ainsworth, 1999), representations such as graphs, equations, and animations model the same problem and the affordances for solving and understanding the problem come from the ability to make connections between representations. If learners are able to make connections, it may provide a path by which they can conduct better investigations and develop stronger explanations.



For example, a learner wants to investigate pressure. A multiple representation tool would provide an animation of air interacting with the underside of a hovercraft (*see picture to the right),* a graph of the pressure over time, and a free-body diagram (*see* Section 2.2.3). A learner could relate the pressure graph with the animation of the air and hovercraft. They would see that when more air molecules were underneath the craft, the pressure was greater. Upon further investigation, by changing the size of the hull, they could see that the same number of molecules over a greater hull area decreased the pressure recorded in the graph. In addition, if the learners make connections between the free-body diagram and the pressure, they would see that an increase in the lift force led to an increase in pressure. Ideally from these observations, a learner could say that pressure and lift force are directly related and pressure and hull area are indirectly related. Through concrete representations, we may aid learners in understanding the equation for pressure: Pressure = Force / Area. Our research suggests that we should use three kinds of representations: animations, graphs, and free-body diagrams in Jacket's Garage to support learners' construction of good *senses of mechanism*.

2.2.1 Animations

An animation is the most concrete representation of the three types in Jacket's Garage; and is the most familiar form of representation for the learners. The primary function of the animation is to make all components of hovercraft flight visible, including air. A secondary function of the animation is "to constrain interpretation of other more abstract and unfamiliar representations" (Ainsworth, 1999). For instance, a free-body diagram (*see* Section 2.2.3) will be present next to the animation. The free-body diagram will have arrows that represent the forces acting on the hovercraft; and we do not want the learners to think the arrows represent velocity or displacement. Therefore, we are using the animation to show the direction and velocity of a hovercraft; and the interpretation of the free-body diagram is constrained, because the learner can see that the arrows and the hovercraft velocity or direction in the animation are not directly related. However, we want to provide them with the support to see how force relates to motion (*see* Section 2.2.3).

2.2.2 Graphs

Graphs provide learners with the support to see connections between the flight of the hovercraft in the animation and the behavior of a particular variable. For example, the pressure graph described earlier in *Section 2.2* relates the lift of the hovercraft with the pressure in the air cushion. Ainsworth (1999) describes a piece of software called *SkaterWorld* where learners use the domain of skating to learn about Newtonian mechanics. She says that they spend a significant amount of time relating a graph with other representations, supporting this ability to see relationships with graphs is important. The hope is “teaching how representations are related may encourage abstraction” (Ainsworth, 1999). Our primary goal for the graph is to provide learners with the ability to identify trends; and identifying these trends is a level of abstraction that can lead to robust knowledge construction.

2.2.3 Free-body Diagrams

Many physics instructors teach free-body diagrams (FBDs) as a means to solve mechanics problems. The intent of these diagrams is to transform an invisible concept, force, into visible vectors. These diagrams show all forces acting on an object at a given time. For example, if a ball is in free fall and there is no air resistance, the only force acting on that ball is gravity, represented as a downward arrow originating from the ball in the FBD (see Figure 1).

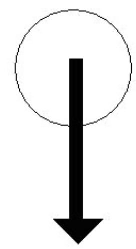


Figure 1

In our case, a FBD could model all the forces acting on the hovercraft and learners could see the FBD of the hovercraft during an investigation. The most important concept in hovercraft engineering is the balance between weight and power. Hovercraft designers want to build hovercraft that are as light as possible but create the largest amount of lift. When a hovercraft is hovering, there is a balance between the force pushing up on the hovercraft and the weight of the hovercraft pulling it back to the ground. Rockets have a much greater lift force than their weight, which accelerates them to space. Hovercraft designers, on the other hand, want a balance between these forces so that hovercrafts are able to hover close to the ground. We want learners to be able to understand this balance. We feel the best why to do this is to have FBDs integrated with our simulations because this will help illustrate the relationships between hovercraft weight and power.

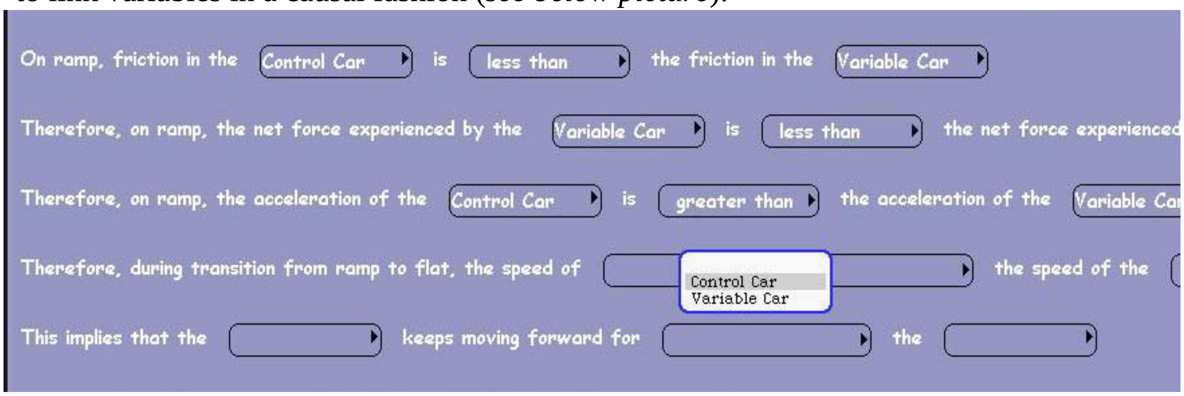
3. Previous Work

3.1 First implementation: SIMCARS

Swaroop Vattam developed our group’s first software tool in a physics environment. Vattam integrated SIMCARS with the Vehicles in Motion unit. Results from a study of the software in an after school program showed promise of bridging the design-science gap among learners and also helping expand students’ content understanding, important when teachers do not know the science they are teaching well (common in middle schools) (Vattam & Kolodner, 2006).

SIMCARS sped up the process of design and experimentation of vehicles and provided a tool for explanation construction. Learners used the software to change wheel, axel, chassis, and propulsion system properties and run experiments (*see below picture)*.

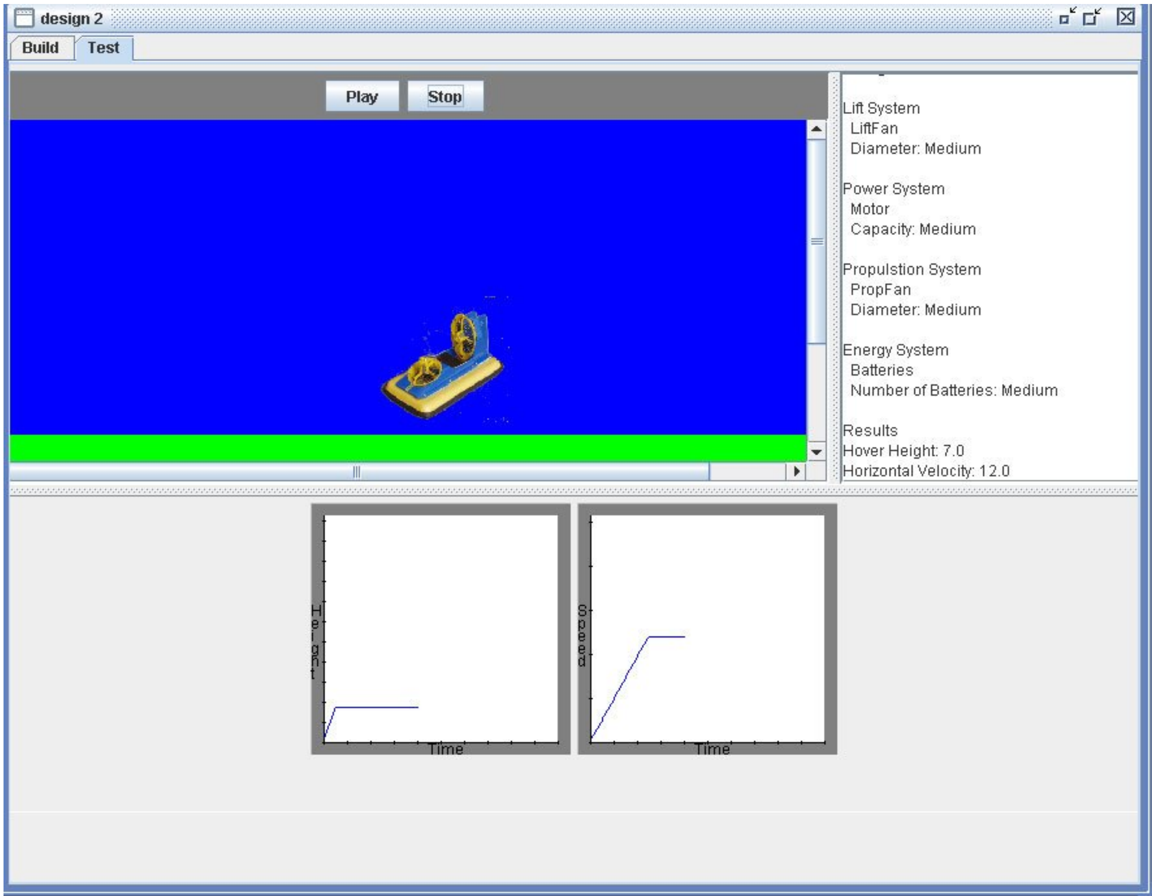
The explanation tool provided a template that allowed learners to express relationships between variables. This template provided a consistent structure for learners to link variables in a causal fashion (*see below picture*).



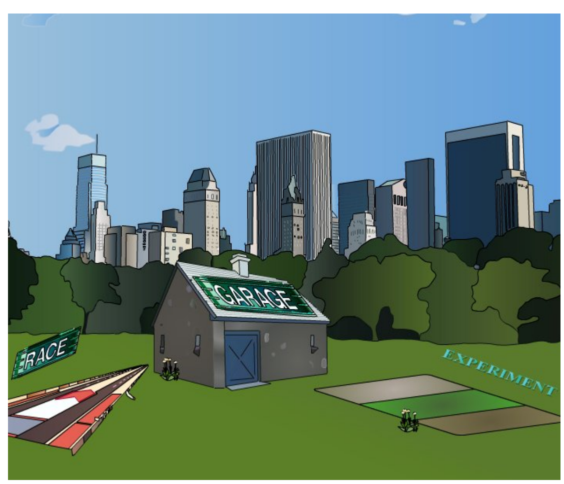
Research from SIMCARS showed that learners achieved a higher level of complexity by repeatedly forming these explanations after experiments (Vattam & Kolodner, 2006). Specifically, learners explored more variables and attempted to explain relationships between them. In addition, learners used more science vocabulary to make causal explanations of their science investigations. However, the software was difficult to navigate and we needed access to classrooms to test our ideas. Our next approach was to implement a LBD unit in a summer camp at Georgia Tech; developing our own curriculum around software gave us more control of the learning environment.

3.2 Second Implementation: SHADE

The second implementation of software was Science of Hovercraft Aided by Design and Explanation (SHADE; Vattam, Kramer, Kim, and Kolodner, 2007). SHADE was broken into two distinct parts. First, learners used an investigation tool that allowed them to rapidly construct hovercraft models and explore a model's behavior in a simulation. This tool includes two visualization components: (1) A two-dimensional animation of a flying hovercraft and (2) graphs of hover-height and speed (*see below picture*). A simple text area displayed a qualitative value for each of three dependent variables: hover time, hover height, and velocity.



Second, learners use an explanation template, adapted from SIMCARS, to construct explanations about what they discovered in their simulations. SHADE provides a means for supporting explanation construction (Vattam, Kramer, Kim, Kolodner, 2007); however, the software could not provide abundant support for learners to investigate physics using the simulation. In addition, SHADE was unattractive, unappealing to learners, and difficult to navigate. We only had a few weeks to debug the interfaces and make them more attractive. We spent the majority of our time implementing all of the backend of SHADE, and we built the interfaces after we had completed all of the functionality. We could have used Squeak to develop SHADE, which was the language we used for SIMCARS, but Squeak's API is not as robust as Java's and we needed a functional database, networking, and XML support.



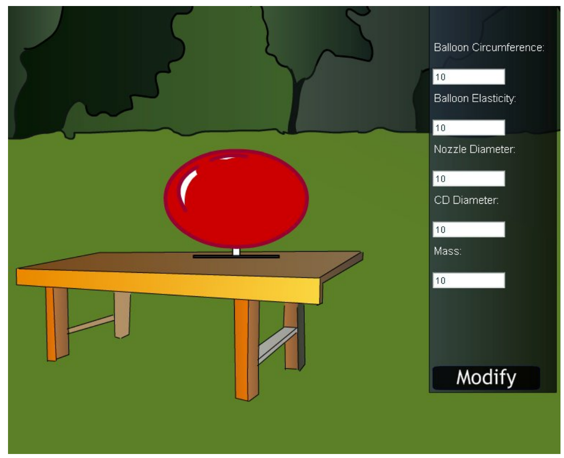
4. New Software: Jacket's Garage

To make our software more engaging and easy to navigate, we developed Jacket's Garage in Flash, which allowed us to develop aesthetically pleasing interfaces and allowed us, as developers, to debug and revise interfaces easily. Flash also provides robust support for networking, databases, and XML, which were functionalities that we wanted to carryover from SHADE.

When a user logs into Jacket's Garage they see a workshop, a racetrack, and an experiment track. We implemented navigation through pointing and clicking on object representations; for instance, if a learner wants to experiment with a hovercraft, they click on the experiment track (*see above picture*). By providing this upgrade in the interface, we hope to appeal to learners so that they are encouraged to interact with the environment, build their science knowledge, and construct robust explanations.

4.1 Garage and Body Shop

The garage is where all of the crafts are stored. There are shelves that display crafts, and scrolling over each craft displays a menu of craft properties. Attached to the garage is the body shop and it is where learners can design and build hovercraft models. They can choose to modify an existing craft from the garage or build a new hovercraft from scratch. A hovercraft sits on a workbench and learners can change properties of a hovercraft with sliders and see the changes affect the hovercraft (*see picture to the right)*. In addition, learners have the ability to test out their hovercraft. We integrated our physics engine (*see* Section 4.5) in every section of the software that requires simulation; this keeps the characteristics of hovercraft flight persistent. Once learners have tested their hovercraft, if they are satisfied with it, they can save the craft to the garage.



From our studies of LBD classrooms, we have discovered that learners spend a significant portion of their time building their hovercraft. The garage and body shop environment reduces this build time and learners can test crafts much faster than what they can do in the real world.

4.2 Experiment Track

We want experimentation to be a formal experience for the learners. Experiments need to be in very controlled environments and they require proper planning. Therefore, we want the software to portray this importance. For example, a learner wants to investigate how pressure affects hovering height. The first aspect of the experiment is for the learner to identify the question they are trying to answer, because we want to make sure that the learners have a goal in mind when they setup an experiment. For instance, "how does cushion pressure effect how high the hovercraft hovers?" Once the learner has defined their question, they can choose the variable they want to manipulate. For this example, they choose to change the hovercraft hull area. They can choose this from a drop down menu and we only provide them with the variables that actually change the pressure of the hovercraft air cushion. A learner must now choose three values for the hull area that they want to test.

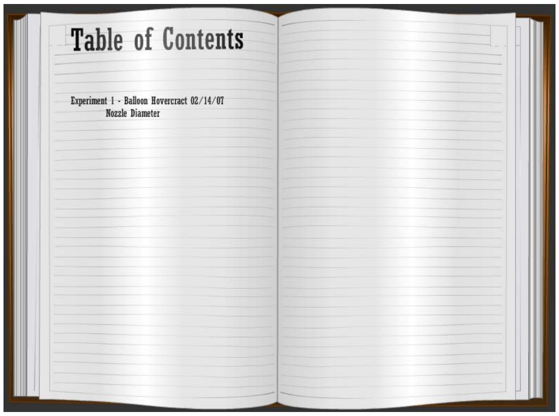
After selecting values, they make a prediction about how the variable they chose will affect the flight of the hovercraft. For instance, a learner could predict that the hull area will affect the hover height of the hovercraft. Specifically when the hull diameter increases, the hover height decreases.

Next, a learner selects the tools for the experiment. If the learner is interested in the cushion pressure during an experiment, they could choose to place a pressure gauge under the hovercraft. The addition of this tool in the experiment would provide numerical, graphical, and visual data about pressure during the simulation. This is consistent with what MER literature because the graph provides the numerical and graphical data of the specific pressure at each instance in time and the animations would show molecules in the air cushion of the hovercraft moving out from underneath the craft during flight.

The learner can run as many trials as they feel necessary to obtain data for analysis. Since the physics engine (*see* Section 4.5) provides randomness, no trial will ever be the same, even with the same experiment setup. Because of this, learners become familiar with actual scientific experimenting, where it is impossible to account for all of the possible variations in hovercraft flight. After running trials, learners identify trends in the data and create explanations using a software tool, a science journal, which is adapted from SHADE.

4.3 Science Journal

The journal interface looks like a bound journal (*see below picture*). The picture to the right shows the interface of the table of contents in the journal. All of the links are hyperlinks so learners can navigate through the journal easily.



The purpose of the journal is for the learner to record all types of data they come across in the software. We want them to feel like real scientists, who carry their laboratory journals everywhere they go. Learners are able to create entries about different hovercrafts, different experiments, and different races. Inside of the journal, not only are the learners encouraged to write freely about their data but we also provide hints within different sections of the journal to help the learner begin to ask scientific questions and think about the importance of their investigations.

4.4 Racetrack

The racetrack is an important aspect of Jacket’s Garage. In the classroom, learners design hovercraft to compete with peers. We provide this competition so that learners can set their own goals as design challenges. For instance, a learner may want to have the longest hover time for the day. A learner can race their crafts and use a global directory to find other learners’ hovercrafts on the server and race against them. The racetrack provides a means for the learners to rank their crafts with those of others.

4.5 Physics Engine

We implemented the SHADE physics engine with heuristics. We took hovercraft variables and multiplied them by a constant to reflect the positive or negative effects of the variables on hover height, hover time, or speed. For instance, in a hovercraft that had hull diameter and motor power as independent variables, the equation for determining the hover height would be as follows:

$$HH = - A * HD + B * MC$$

Where HH is Hover Height; HD is Hull Diameter; MC is Motor Capacity; and A, B are constants. The negative constant represents an indirect relationship and the positive constant represents a direct relationship between variables. Jacket's Garage uses real physics equations to calculate values of the dependent variables. There is also randomness incorporated within the equations. Therefore, each trial of an experiment will be different even if the values of the independent variables do not change. We looked at how a real hovercraft works to develop the physics engine.

A hovercraft hovers because of a cushion of air underneath the hull of the hovercraft. The force from the air molecules interacting with the hull overcomes the force of gravity on the craft and accelerates the craft upward. At a certain point, more air is escaping around the air gap of the air cushion than is interacting with the hull. The unbalanced gravitational force begins to accelerate the craft towards the ground. At a certain point, less air is escaping from the air gap and more air molecules are interacting with the hull. The force of the molecules begins to accelerate the craft upward again. In conclusion, the hovercraft is constantly accelerating up and down around an equilibrium point.

The lift system of the hovercraft provides the air for the air cushion. In full-scale hovercrafts, this is typically a fan pushing air underneath the hovercraft. The fan speed, motor power, and fan diameter determine the flow air of the air underneath the hovercraft. In the case of the software, the balloon and nozzle or the motor and fan for the model hovercraft determine the flow rate. At each instance of the simulation, the lift system determines the number of molecules and the acceleration of those molecules into the air cushion.

To implement this physics mechanism in the software it is impossible to account for an accurate number of air molecules in the air cushion. As an estimate, a kilogram of air underneath the hovercraft at a given time may accounts for many times Avogadro's number of air molecules, $6.022 * 10^{23}$. This would be impossible to keep track of in a simulation. The solution is to reduce the number of molecules to a reasonable number; in our simulation, we are using approximately one hundred at any given instance. To compensate for the much smaller number of molecules in the simulation than in the real world, we assign a mass to the molecules. Since force is acceleration times mass, the molecules are acting more like rubber balls underneath the hull.

Once a molecule enters into the air cushion it can bounce off any surface: hull, ground, or skirt. If the molecule interacts with the hull, the acceleration of that molecule in the vertical direction determines the force exerted by that molecule. By summing all these force interactions with the hull and then subtracting the weight of the hovercraft, we can calculate a vertical net force.

With a net force, the hovercraft accelerates in the direction of that force based on Newton's second law of motion. Once acceleration is determined, then a velocity and displacement can be determined. The addition of displacement to the current height of the hovercraft results in an animation of the hovercraft rising or falling.

5. Coming Up

We will test the effectiveness of our design during a science summer camp in 2007, sponsored by the Center for Education Integrating Science, Mathematics and Computing (CEISMC). The camp will be video recorded by two cameras and audio recorded by ambient microphones in the classroom. The instructor will take daily notes and at least one researcher will be observing the classroom at all times. We will be analyzing how learners use the software for science investigations and explanations. We hope that our analysis will confirm our design decisions. Specifically, we want to see what the experiment capabilities are for learners to understand physics and create good *senses of mechanism*. We also want to see if the changes in our explanation tool are justified and we hope to confirm this by analyzing learner discourse among peers, researchers, and instructors.

5. Discussion

We hope Jacket's Garage can embed meaning into representations, thus supporting learners as they construct good *senses of mechanism*. Furthermore, we want learners to use their understanding to construct meaningful explanations. We have developed our software on a backbone of LBD theories, physics misconceptions analyses, and multiple representation research. We hope that learners will construct good *senses of mechanism* with Jacket's Garage because it is engaging and provides MERs to help them with their science investigations. In addition, we hope that learners will use the software's explanation tool to construct significant explanations. We hope that our analysis of the summer camp in 2007 will shed light on how learners can use a combination of simulation and explanation tools for productive knowledge and explanation construction.

6. References

Ainsworth, S. (1999). The Functions of Multiple Representations. *Computers & Education, 33*(2), 131-152.

Clement, J. (1982) Students' preconceptions in introductory mechanics. *American Journal of Physics, 50, 66-70.*

Clement, J. (1983) A conceptual model discussed by Galileo and used intuitively by physics students. In D. Gentner & A. L. Stevens (Eds.), *Mental Models* (pp. 325-340). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453–494). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

diSessa, A. A. (1993). Toward an epistemology of physics. *Cognition and Instruction, 10*(2), 105-225.

Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., Puntambekar, S. (1998). Learning by Design from theory to practice. In A. Bruckman, M. Guzdial, J. L. Kolodner, & A. Ram (Eds.), Proceedings of ICLS 98 (pp. 16–22). Charlottesville, VA: AACE.

Kolodner, J. L., Camp, P. J., Crismond, D., Fasse, B., Gray, J., Holbrook, J., et al. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: putting Learning by Design ™ into practice. *Journal of the Learning Sciences*, *12*(4), 495-547.

Larkin, J. (1983). Problem Representation in physics. In D. Gentner & A. L. Stevens (Eds.), *Mental Models* (pp. 75-98). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1-11). Norwood, NJ: Ablex Publishing Company.

Rosengrant, D., Van Heuvelen, A., & Etkina, E. (2005). *Free-body diagrams: necessary or sufficient?*, Sacramento, CA, USA.

Smith, J. P., diSessa, A. A., & Roschelle, J. (1993). Misconceptions reconceived: A constructivist analysis of knowledge in transition. *Journal of the Learning Sciences, 3*(2), 115-163.

Vattam, S., & Kolodner, J. L. (2006). *Design-based science learning: important challenges and how technology can make a difference.* Paper presented at the International Conference of the Learning Sciences, Bloomington, Indiana.

Vattam, S., Kramer, C., Kim, H., & Kolodner, J. L. (2007). Effects of technology-based support for explanation construction on learners' discourse during design-based learning in science. *Accepted for* Conference of Computer Supported Collaborative Learning 2007. Rutgers, NJ.