Molecular Dynamics for Everyone: A Technical Introduction to the Molecular Workbench Software

Charles Xie
The Advanced Educational Modeling Laboratory
The Concord Consortium
qxie@concord.org

The world moves because molecules move. Studying the motion of molecules is important to the understanding of many critically important concepts in physics, chemistry and biology. A fundamental goal of scientific research is to learn how things work, which at the microscopic level primarily means how atoms and molecules move to perform certain functions such as chemical reactions, molecular recognition and protein synthesis.

The ability to reason about complex phenomena in terms of fundamental facts and theories describing the structures, interactions and dynamics of the atoms and molecules of which all things are made is called the *Molecular Literacy* [1]. This bottom-up perspective, in addition to its philosophical significance, is increasingly important, as nanotechnology and biotechnology are emerging as the twin engines of the next industrial revolution. Molecular Literacy empowers people to better understand known phenomena and to anticipate the properties of novel arrangements of matter under new conditions. Achieving Molecular Literacy at the grassroot level is therefore crucial for future scientists and engineers as it promotes the imaginative thinking needed to create and develop new nano-structured and bio-mimetic materials and devices for diverse applications to meet critical healthcare, energy, and environmental challenges.

Since the invention of molecular graphics, a subject that focuses on visualizing molecules using 3D computer graphics, chemists have embraced tools capable of rapidly displaying molecular structures and viewing them from different perspectives. Because of the appealing effect of 3D graphics, commercial computer-aided design packages often feature molecular visualization tools to promote the products. Since the advent of the Internet Era, several free tools have been developed for showing molecules on the Web. These tools are widely used by educators to teach molecules [2].

Most molecular viewers, however, are mainly designed to show static structures. The user can rotate and translate the entire rigid structure or change the view angle dynamically to create a motion effect, but the atoms do not move relatively to each other. Viewing static structures helps students learn the structures, but it is often far more important to learn the functions—after all, we study molecules because we hope to make use of what they can do. Although one can argue that in many cases there exists a strong structure-function relationship that can help people derive functions from structures, it is impractical to expect inexperienced students to be able to reason using the relationship that may be evident only to experts. Too often have we seen an excited chemist trying in vain to explain to non-experts what he or she sees in a 3D model of a molecule that is beyond a cool picture of some 3D structure. It would be most helpful if a molecular mechanism can "speak for itself" in a dynamic visualization to fill the gap. Some molecular viewers can sequentially display a series of static frames, which can be different states of the same molecule observed experimentally or computed theoretically, to create an animation of a conformational change. This is a step forward to help students learn molecular mechanisms, but it only allows them to passively watch what was set to happen. For a tool to do a better educational service, students should be permitted to "mess up" with the models, try many what-if's, and see what happens.

Funded by the National Science Foundation, a team of the Concord Consortium has been developing a free, open-source program called the *Molecular Workbench* (MW) (http://mw.concord.org), which brings a salient, dynamic molecular world to the computer screen and allows students to interact and experiment with it[1] based on *real-time* molecular dynamics (MD) calculations and visualizations [3].

MD modeling provides a powerful means to foster the Molecular Literacy because it complements and enhances traditional instructional approaches, including formal mathematics. Cognition can be viewed as a process of making and manipulating mental models of imaginary objects and events [4]. A scientific model that comprises basic units of coherently structured knowledge in a tested and integrated framework, if presented appropriately with effective pedagogy, can be enormously useful in helping students develop correct mental models from which they can make logical inferences such as explanations, predictions and designs. Molecular modeling, which is an important part of contemporary scientific research [5, 6], constitutes the theoretical foundation for creating "objective conceptual models" [7] that can be used by students to explain and investigate many natural phenomena at the molecular scale and thereby develop concrete mental models about them.

It is important to emphasize the educational significance of computational models based on science as opposed to movies composed of frames of images and animations based on simple timeline rules. In a computational model, critical behaviors emerge from algorithms derived from first principles, which give it the explanatory power that can be used to manifest existing knowledge and the predictive power that can be used to explore unknown domains. A computational model can accurately simulate a large number of different phenomena by varying parameters, configurations, initial conditions and boundary conditions. A movie or an animation, in contrast, can only illustrate a handful of situations that are recorded or preprogrammed. *If a picture is worth 1,000 words and a movie is worth 1,000 pictures, a model is worth 1,000 movies, or 1000,000 pictures, or 1000,000,000 words*. Because a model provides a much larger intellectual space and more freedom for learners to explore, create and invent, effective, profound learning is more likely to occur.

This article presents the scientific methods, the educational background and the technical ideas behind the software. It is assumed that you have some preliminary knowledge about computational science, educational technology and software engineering. In the first section, we review the basic procedures of classical MD simulations. In the second section, we discuss the requirements of educational simulations and the technical work needed to be done to meet these requirements. In the third section, we introduce our ideas to evaluate student learning based on MW materials. The results of our educational research are not covered in this technical article. Please see a paper by Pallant and Tinker [8] if you are interested in them.

How to simulate the motion of atoms and molecules?

Everything moves because of forces, which result from the interactions among atoms and molecules. We will begin with how theoretical chemists model interatomic and intermolecular interactions.

[1] You can watch a movie about MW at: http://mw.concord.org/modeler/small/small.html.

There are two levels of modeling for molecular interactions. One is based on quantum mechanics calculations [9], which is beyond the scope of this article. The other uses empirical forms, which will be introduced in the following.

Molecular mechanics force fields

Atoms basically interact with each other through van der Waals forces and electrostatic forces. When they are covalently bonded to others, strong forces hold them together as stable chemical groups. A widely used mathematical model for the potential energy of a molecular system consists of six types of interactions:

$$U = V_{\mathrm{LJ}} + V_{\mathrm{EL}} + V_{\mathrm{BS}} + V_{\mathrm{AB}} + V_{\mathrm{PT}} + V_{\mathrm{IT}}$$

The first type, V_{LJ}, is the Lennard-Jones potential that has an attractive part representing the van der Waals energy and a repulsive part representing the Pauli repulsion:

$$V_{\mathrm{LJ}} = \frac{1}{2} \sum_{i,j,i\neq j} 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{R_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{R_{ij}} \right)^{6} \right]$$

where R_{ij} is the distance between the *i*-th and *j*-th atom, $\varepsilon_{\mathrm{ij}}$ is called the van der Waals dissociation energy, and σ_{ij} is called the collision diameter. The dissociation energy is equal to the amount of energy needed to pull a pair of atoms in the strongest van der Waals binding state apart. The collision diameter is approximately the distance at which a pair of atoms bounces off from each other in a normal, non-reacting condensed state.[2] The power of the negative term, which is sometimes also called the London dispersion force, has a root in the quantum mechanical calculation of the binding energy of the hydrogen molecule, but the power of the positive term has no apparent theoretical basis (sometimes, it is set to be 9 to soften the repulsion core for dense systems).

V_{EL} is the electrostatic potential energy according to Coulomb's Law:

$$V_{\mathrm{EL}} = \frac{1}{2} \sum_{i,j,i\neq j} \frac{q_i q_j}{R_{ij}}$$

where q_i is the charge of the *i*-th atom. Compared with the van der Waals potential, the electrostatic potential is a stronger, more long-range interaction. A pair of charged atoms in vacuum will be able to "feel" each other from quite a distance away, whereas a pair of neutral atoms will "feel" each other's existence only when they are close. For crystals or solutions, the Ewald Sum is often needed to compute the summation of the weak contributions from numerous remote charges. But if only qualitative results are needed, this expensive procedure may be skipped to speed up the simulation.

V_{BS} is the bond-stretching energy standing for the elastic interaction between a pair of atoms connected by a covalent bond, V_{AB} the angle-bending energy standing for the interaction among three

[2] If a pair of atoms can react to form a covalent bond, the length of the bond between them can be smaller than the collision diameter.

covalently-bonded atoms that form a stable angle, and V_{PT} and V_{IT} the proper and improper torsional energies standing for the interactions among four covalently-bonded atoms that form a stable proper and improper dihedral angle (see Fig. 1):

$$V_{\mathrm{BS}} = \frac{1}{2} \sum_{m \in \mathrm{bonds}} k^{l}_{m} (l_m - l^{0}_{m})^2$$

$$V_{\mathrm{AB}} = \frac{1}{2} \sum_{m \in \mathrm{angles}} k^{\theta}_{m} (\theta_m - \theta^{0}_{m})^2$$

$$V_{\mathrm{PT}} = \frac{1}{2} \sum_{m \in \mathrm{torsions}} V_{m} [1 + \cos(n_m \omega_m - \gamma_m)]$$

$$V_{\mathrm{IT}} = \frac{1}{2} \sum_{m \in \mathrm{torsions}} k^{\xi}_{m} (\xi_m - \xi^{0}_{m})^2$$

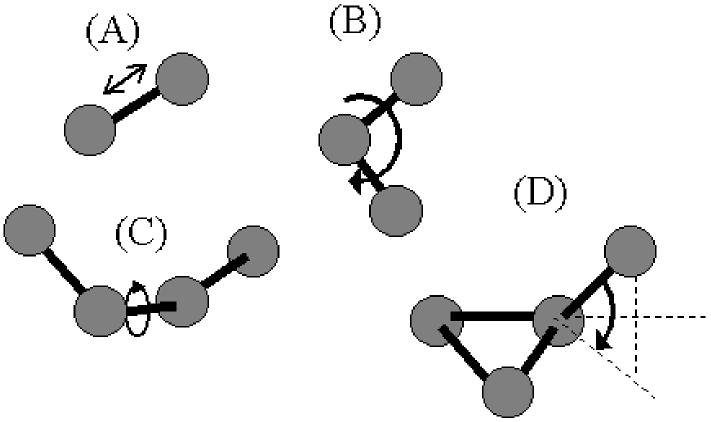


Figure 1: A schematic illustration of the interactions that model covalent bonding: (A) Bond-stretching force; (B) Angle-bending force; (C) Proper torsional force; (D) Improper torsional force.

where l_m is the distance between the two atoms of the m-th bond, l_m^0 is the equilibrium bond length, k_m^l is the bond strength, θ_m is the m-th angle between the two adjacent bonds that share a common atom, θ_m^0 is the equilibrium bond angle, k_m^θ is the strength, ω_m is the m-th dihedral angle between the two adjacent angles that share a common bond, n_m is the periodicity factor which determines the number of equilibrium dihedral angles in a 360° rotation, γ_m is the phase shift, V_m is the amplitude, ξ_m is the m-th improper dihedral angle among four atoms that are not bonded successively to one another, ξ_m^0 is the equilibrium improper dihedral angle, and k_m^ξ is the strength.[3]

The last four items are called the *bonded interactions*, which maintain the bond lengths, the bond angles and the dihedral angles so that chemical groups will remain sterically stable in an MD simulation[4] (in MW we call these constructs radial bonds, angular bonds and torsional bonds). The first two terms, the Lennard-Jones potential and the electrostatic potential, are called the *non-bonded interactions*. In MD simulations, they are more important than the bonded interactions. It is the non-bonded interactions among the atoms of a macromolecule that affect its secondary structure. It is the non-bonded interactions among the atoms of different molecules that organize them into crystals, complexes and other assemblies.

It should be pointed out that the decomposition of the potential energy of a molecular system into the above force field terms is purely empirical. In other words, they are the mathematical models for describing the chemical forces that stabilize the structures derived from diffraction patterns obtained

[3] The bond-stretching potential given by Hooke's Law does not permit a bond to break. The more a bond is stretched, the greater is the force to pull the atoms back. As a result, the above force fields cannot be used to simulate chemical reactions, which involve making and breaking bonds. We have proposed a method that allows bonds to make and break, and thus makes it possible to simulate some simple reactions [10].

[4] Some molecules, such as benzene, have delocalized bonds that involve more than four atoms. However, no higher terms of energy decomposition seem to be necessary in this treatment.

using crystallography. The force field approach *per se* has no power in explaining how the structures are initially assembled from discrete atoms, which is essentially a question of how to simulate the origin of life.

Most MD packages for biomolecular simulations are based on the molecular mechanics as described above, though they may differ in the parameterization (the determination of the bond strengths, the van der Waals parameters and the partial charges). In MW, to ensure broad applicability, an atom can be considered as a generic particle with parameters that can be changed freely.

Molecular dynamics simulations

Having defined the interactions among atoms, the position, velocity and acceleration of each atom are calculated using a numerical method (e.g. the Verlet method or the Runge-Kutta method) to solve Newton's equations of motion according to the forces derived from the gradients of the interaction potentials involving the atom:

$$m_i \ddot{\mathbf{R}}_i = -\nabla_i U(\mathbf{R}_1, \mathbf{R}_2, \ldots\ldots, \mathbf{R}_n)$$

where $\mathbf{R}_i$ is the position vector of the *i*-th atom and m_i is its mass. The numeric integration is carried out stepwisely. The process is repeated at each discrete time step. The trajectory of each individual object can be tracked by connecting its states to form a time series. The time evolution of the entire system can be viewed as a fiber bundle of time series in the phase space.

That is all you need to do to get an MD simulation up and running. For advanced topics, such as boundary conditions, thermostats, pistons, statistical analysis and so on, interested readers can consult with Ref. 6, or read the online User's Manual within MW.

The MD method is very useful in scientific research, because it satisfies the following fundamental physical laws:

- **The First Law of Thermodynamics**. The Law of Conservation of Energy automatically emerges in an MD simulation. If there is no energy input/output through external forces or dissipation through friction, the total energy, which is the summation of the potential energy and the kinetic energy for all the atoms in the system, remains constant within the tolerance of numerical errors. This can be used as a criterion to check if a simulation runs properly.
- **The Second Law of Thermodynamics**. Although the Reversibility Paradox suggests that classical dynamics might be at odds with the Second Law of Thermodynamics, MD simulations of basic processes such as diffusion, heat transfer and phase transition clearly show that the entropy of an isolated molecular system always tends to maximize. Despite of the fact that it is possible to create special initial conditions that lead to a process of entropy reduction in an isolated system,[5] in practice we have never found that such special conditions can spontaneously arise during an MD simulation for a many-body system.

[5] Consider an impact process in an isolated system: a high-speed atom bombards and breaks a microcrystal. If we stop the simulation after the crystal has been broken, reverse the velocities of every single atom in the system, and then continue to run it, we can reverse the process—the atoms re-assemble into the original crystal. The entropy decreases in this spontaneous process in

- **The Law of Momentum Conservation**. As the Law of Conservation of Energy, this Law also automatically emerges in an MD simulation. This Law dictates each collision among atoms. The overall result is that the total linear and angular momentum of the system conserves.
- **Other statistical laws**. Important laws in statistical mechanics, such as the Theorem of Energy Equipartition, Maxwell's Theorem of Speed Distribution, and the Boltzmann Distribution are all guaranteed in MD simulations. We can even simulate the Galton Board that demonstrates the normal distribution.[6]

What needs to be done to make molecular dynamics modeling accessible to students?

Most MD programs involve using a pre-processor to prepare simulations and a post-processor to analyze results. When a calculation is actually being done after it is submitted to a computing service, the user is rarely given a chance to intervene. Moreover, many programs require the user be able to work with command lines and scripts, and feel comfortable dealing with raw data. These requirements are prerequisites for scientists. But they become disadvantages when novice users in schools try to use them without the aid of an expert.

The overarching goal of a learner-friendly MD program is that average students can use it to learn science. The intermingled complexity of learning and science requires a highly integrated system that is capable of supporting both. Instead of having students work with a set of distinct tools and switch back and forth, the ideal technological solution encapsulates of the entire process of building models, setting up conditions, running and controlling simulations, visualizing results, recording observations, testing, monitoring learning progress, and feedback into a single program with a unified graphical user interface. In the following subsections, we will discuss the important facets of such a system in details.

Simulations must be interactive

When a scientist performs an MD simulation, the goal is not to watch how it unfolds on a computer screen. There is seldom a need to spend precious computing resources on visualizing the intermediate results on the fly. Neither is there a need for the user to alter the parameters and conditions arbitrarily during a simulation. Often, a simulation starts with a fixed set of inputs, and records the intermediate results while it is running. When it completes, the stored results can be analyzed to retrieve the needed information corresponding to the given set of inputs.

For a program to be more educationally useful, however, opportunities must be provided to students to *interact with simulations themselves*. To support inquiry, students must be allowed to adjust parameters and add inputs at any time, and see the emergent behaviors of the simulated systems

the simulation without work and/or cooling from the outside world. This paradoxical result seems to be a violation of the Second Law of Thermodynamics at first glance. But in practice, the probability of encountering the state of a many-body system in which all velocities take the values and directions as if they were manually reversed as described above is so low that it can be neglected. See: http://mw.concord.org/modeler1.3/mirror/thermodynamics/loschmidt.html

[6] http://mw.concord.org/modeler1.3/mirror/mechanics/galton.html

instantaneously. Only through interacting with simulations freely in many different ways and watching the results can students discover the cause-and-effect relationships revealed by the simulations and therefore construct their own mental pictures about the important physical and chemical concepts embodied in the simulations.

Technically, an educational simulation is required to do both the calculations and the visualizations at the same time in order for students to see the entire process in all possible levels of details, and in *real time* in order for them to manipulate the system and observe its responses right away. Translated into programming terms, the MD code, the visualization tool and the graphical user interface must be integrated seamlessly in the run time.

The requirement of interactivity, however, limits the sizes of simulatable systems on personal computers. There are two solutions to this issue. The first one counts on the continuous improvements on computer power. As the Era of Multicore Computing is upon us, parallelizing the software system to maximally use the power of multicore processors will allow larger systems to be simulated. The second one uses a coarse-grained approach to reduce large systems into models with a tractable number of particles, each of which represents a large number of grouped atoms that form a stable structure with insignificant internal vibrations. These effective particles interact with each other to move, join and break. For example, it is common to use a model in which an amino acid is represented by a single particle to study the mechanisms of protein folding [11]. Implicit solvation that employs effective fields to simulate solvent-solute interactions can be used to save the computational cost needed for the vast number of water molecules that have to be otherwise present in the simulations of molecules in aqueous solutions. The blue window in Fig. 2 shows a coarse-grained model for micelle with implicit solvation.

Simulations should be easy to create

Models are abstractions of data that are composed of different objects at different locations in the phase space under different conditions and with different initial settings. A simulation engine, if generic and powerful enough, allows us to build as many models as permitted by its capacity to match the diversity of reality. It becomes apparent that, to harness the power of this theoretical capability, users need to be able to turn their modeling ideas into computational models. A user-friendly system for constructing simulations has twofold importance. First, it allows educators to create simulations for teaching, as alternatives to traditional drawings

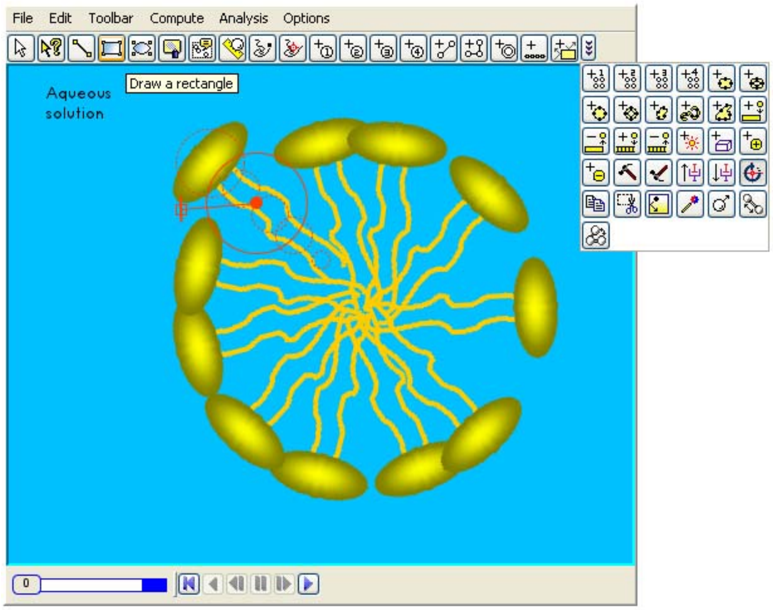


Figure 2: A screenshot of the 2D Model Builder in action. The menu bar and the tool bar above the view window provide many tools needed to construct models and set up simulations. Each type of object also provides a pop-up menu and a property editor from which the user can edit and modify the properties of an object of its type.

and illustrations. Second, it allows advanced students to design their own models, a process during which their modeling skills and creative thinking can be trained. Engaging students to construct models and simulations may also serve as an introduction to molecular modeling, which will benefit them if they end up choosing science and engineering as their future careers.

MW has a What-You-See-Is-What-You-Get (WYSIWYG) 2D model builder that makes models in a way that is as easy as making shapes with a drawing program (see Fig. 2). With this model builder, many types of objects can be added, and every object in a model can be cut, copied, pasted, dragged-and-dropped, and edited through the supporting pop-up menus and property editors. User actions are undoable and redoable. Annotations can be added to make a model easier to understand. For a coarse-grained particle model, custom images can be attached to "decorate" the particles so that the appearance of a model will bear a resemblance to illustrations commonly seen in textbooks, particularly for molecular biology (see the simplified graphical representation of lipid molecules that form the micelle in Fig. 2).

Thanks to the integration of simulation and construction, there is no border between construction and run in MW. The user can, at any time, run the model while constructing it. This characteristic feature, stemming from the dynamic nature of MD models, is a major difference between building a static model and building a dynamic one. In fact, test-running a model under construction is an important part of the constructing process as it allows the user to build through trail-and-error cycles. Unstable constructs can be automatically removed or spotted when a model runs (or a procedure of energy minimization is called).

For users who are not satisfied with the abstraction of 2D models, a rudimentary WYSIWYG 3D model builder is available for creating 3D models (Fig. 3). It allows the user to build molecules from scratch by laying down atoms in the 3D space with the assistance of movable helper planes and joining them by radial, angular and torsional bonds. More complex chemical systems can be built using a set of building blocks that includes all the amino acids and nucleotides and many small organic molecules. A crystal builder is also provided to build a limited number of crystal lattices. Atoms can be selected, translated, rotated, duplicated, and deleted as blocks. Three different views are available for the author to set the perspective to observe a simulation. These views include a regular view in which the model is viewed as a whole at different zooming distance, a navigation view in which the user

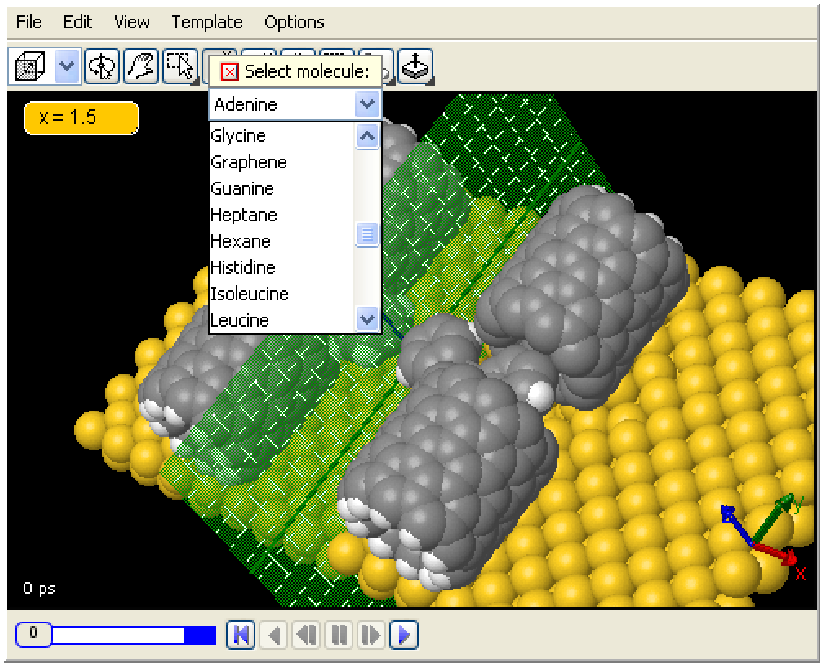


Figure 3: A screenshot of the 3D Model Builder in action. It can be used to build models as complex as this nano car, which has four short carbon nano tubes as the tires and wheels.

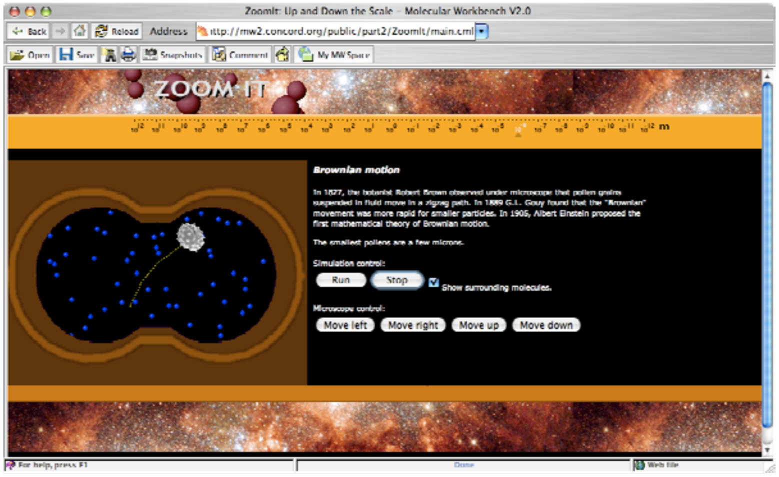
can move the "camera" around to experience an immersive effect of "flying into a molecule", and a rover view in which the "camera" is attached to an atom to mimic the effect of "riding on an atom" when a simulation runs.

There are many pre-made models that cover quite a breadth of science and are freely available in MW. The collection constitutes a solid scientific foundation of MW. As the software development continues, more examples will be added to consolidate and expand this foundation.

Simulations should be embedded in a learning environment

Simulations are not broadly useful in classroom without accompanying instructional materials. Many educational applications provide lesson plans or worksheets for students to use, separated from the software tools. But the optimal way of using simulations is to embed them in a learning environment that provides all the essential elements needed for a learning process. We call such a complete package a *learning activity*. A good learning activity motivates, scaffolds, and supports student exploration of models and simulations. It also provides background information, opportunities for reflection, and methods of monitoring student progress in a designed context and evaluating learning. MW is a versatile learning environment that offers this kind of classroom-ready learning activities. Moreover, it provides an *authoring system* for creating them.

A learning activity in MW usually consists of multiple pages. A page is a screen space in which text can be typed and styled like in a word processor and many kinds of components can be inserted and customized. The fact that these elements can be placed anywhere on a page and mixed with characters, images and links allows the author to design high quality, visually appealing and self-explanatory pages. Authors who have experience in creating simple HTML web pages should be able to author similar pages in MW without a problem.


ZoomIt: Up and Down the Scale – Molecular Workbench V2.0
Back
Reload
Address
http://mw2.concord.org/public/part2/ZoomIt/main.cml
Open
Save
Snapshots
Comment
My MW Space
ZOOM IT
m
For help, press F1
Done
Web file

Brownian motion

In 1827, the botanist Robert Brown observed under microscope that pollen grains suspended in fluid move in a zigzag path. In 1889 G.L. Gouy found that the "Brownian" movement was more rapid for smaller particles. In 1905, Albert Einstein proposed the first mathematical theory of Brownian motion.

The smallest pollens are a few microns.

Simulation control:

Run Stop ☑ Show surrounding molecules.

Microscope control:

Move left Move right Move up Move down

Figure 4: This screenshot shows that an MD model for the Brownian motion is customized and contextualized in an activity that teaches the concept of scale.

A model container, within which a simulation runs, is a core component that can be inserted into a page. With a rich set of pluggable components that can communicate with a model container through command and data channels, a custom user interface can be built for each simulation. The user interface can comprise controls of the simulation, buttons and sliders for changing the parameters, and graphs for displaying the outputs. Customized user interfaces are important because they establish a learning space that is constrained only within the topics covered in the activity. For example, Fig. 4 shows a customized user interface for showing the Brownian motion. Although standing behind the scene is the entire engine that is capable of doing numerous other kinds of simulations, the end user of the activity needs not be concerned with anything beyond what is presented on this succinct page in order to learn the intended subject.

Another kind of components that can be inserted into a page is questions. There are three types of questions that the author can set up: multiple choice question, free response question and image question. An image question is a type of question invented in MW, which requires the student to take a snapshot image of a simulation as an answer to the question. These questions can be used to test the student's prior knowledge (pre-test) and measure the gains after learning through an activity (post-test), for example.

The authoring system, along with the model builders, has allowed us to create a myriad of learning activities that can be found in the material repository within MW (the internal MW page when you press the Home Button on the tool bar of MW). Some of these activities have become reasonably sophisticated and self-contained enough in content to be qualified as *interactive textbooks*. For those who are interested in learning how to create models and activities, a comprehensive online *User's Manual*, which is written using the same authoring system, provides numerous working examples.

Simulations should be easy to share

A pedagogy made possible by the easy-to-use model builders is to involve students in creating simulations, a process that can be devised to embed instructional steps that lead to progressive conceptual understanding. This learning path, different from inquiry using interactive media based on existing simulations created by experts, seems to be practical at college level where students have obtained adequate prior knowledge needed to understand the basics about simulations to get started.

From the point of view of social constructivism, the creation process and the end product must be shared with others in order for the full effect of learning to take root. It is through the creation of a molecular model that is shared and becomes what Papert calls a "public entity" that learning is strongly reinforced [12]. In the process of building, sharing or collaborating, students learn their subjects well because they have to think hard about them and figure out how to present their modeling ideas to others.

There is no hurdle for students to share a simulation in MW. The MW system allows users to submit their modeling work through the MW Space, a web application that facilitates social interactions in virtual classrooms in a culture where models are the central elements. Students can easily upload their models to their MW accounts, and decide with whom they will be shared. They can choose to share with their classmates, their teachers, or the world users of MW. When the user opens an MW page that contains a simulation, it is ready to run. No extra step is needed. All saved parameters, configurations and initial conditions are restored in a snap when it is loaded. With the authoring system, students can make procedures that should be followed to produce the desired results, and the controls to achieve them, as they would with real experiments. They can also articulate their motivations, explain what their simulations will show, and narrate how they were constructed. This makes their work read more like presentations instead of just plain models that may be less comprehensible.

How do we know if students learn

To some extent, going through an MW activity for a student is similar to going through an experimental procedure in a wet lab. The common things are that a student needs to read instructions, follow certain procedures, operate some instruments, observe what happens, record some data, and write a report at the end. The Report System in MW monitors what students do during this process, automatically generates reports, and allows teachers to track down students' progresses.

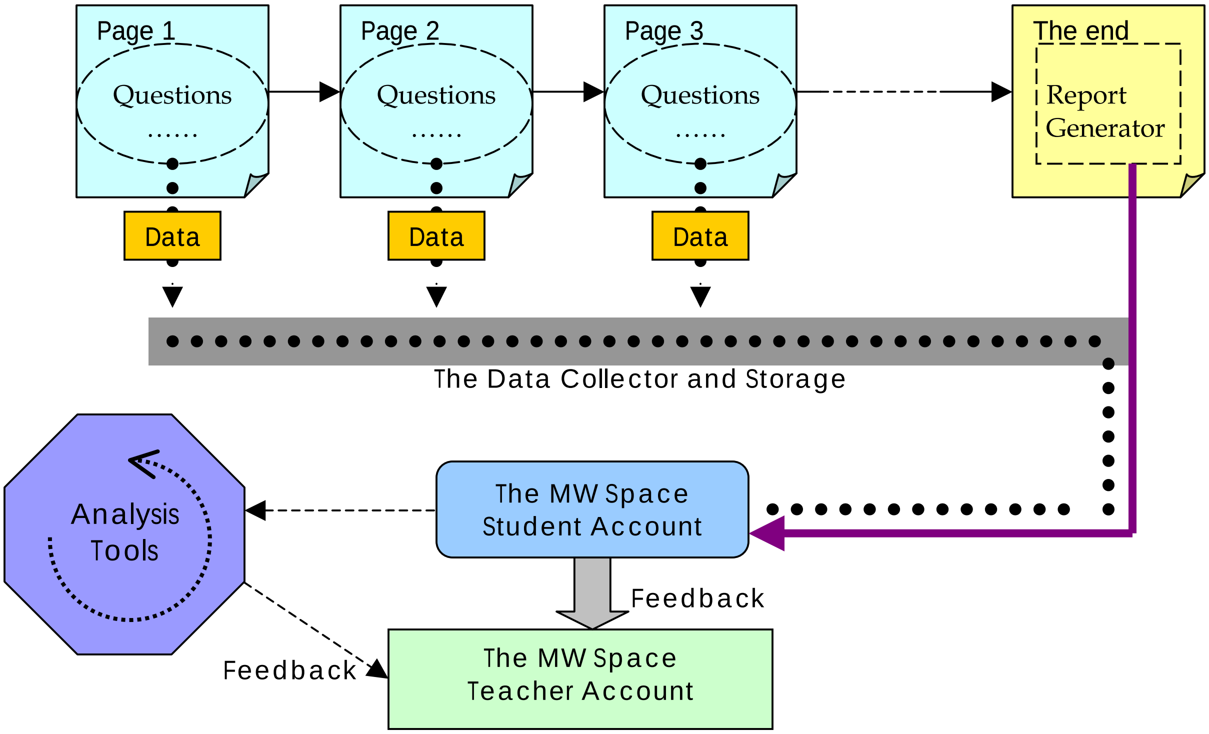


Figure 5: The flow of collecting student work during a learning activity, supported by the Report System.

The Report System consists of five parts. The first one is questions, which can be embedded into a learning activity as described in the previous section. The second is a data collector that gathers and stores a student's inputs from questions. The third is a snapshot facility that allows students to take a snapshot image of a simulation or a graph. A snapshot image captures what a student sees happening on the screen, which are sometimes difficult to describe merely with words or numbers. There is also a set of tools for annotating a snapshot image, which students can use to highlight and explain certain parts of the image. The fourth is a report generator that automatically converts student data into a readable page that can be printed, saved or submitted. The fifth is the MW Space that registers students and teachers, receives and stores reports in a database, and provides feedback to teachers. Fig. 5 shows a schematic illustration of the workflow of the Report System.

The importance of the feedback loop among material developers, students and teachers must be stressed. Producing learning materials, designing effective pedagogies, and implementing them in classroom are essentially a dynamic, time-consuming trial-and-error process that deeply rests on constructive interactions among the users, the implementers and the developers. It is important that the teachers be able to track the learning processes of the students, and report problems and inefficacies to the developers. The materials can therefore be revised and the quality be improved.

With the Report System and its future improvements on critical issues such as data mining and data analysis, MW can be a very useful tool for conducting educational research, particularly for studying the effectiveness of using computational models in education. While few disagree upon the extensive use of interactive media in teaching, many (including us) are still searching for the best design strategy of these media and the best pedagogy of using them. The research on this avenue may yield insights that would develop the next generation of educational media.

Acknowledgements:

The author thanks B. Berenfeld, D. Damelin, D. Markman, A. Pallant, E. Rosenbaum, B. Tinker and R. Tinker for their encouragement and numerous suggestions for developing and improving the Molecular Workbench System. This article is based upon work supported by funding from the National Science Foundation.

References:

1. Molecular Literacy, http://molit.concord.org; M. Patrick, unpublished manuscript.
2. World Index of Molecular Visualization Resources, http://www.molvisindex.org
3. R. Tinker and Q. Xie, *Applying Computational Science to Education: The Molecular Workbench Paradigm,* Computing in Science and Engineering, in press, 2008.
4. D. Hestenes, *Notes for a Modeling Theory of Science, Cognition and Instruction*, paper presented at The 2006 GIREP conference: Modeling in Physics and Physics Education, Amsterdam, 2006.
5. T. Schlick, *Molecular Modeling and Simulation*, 1st Edition, Springer, 2002.
6. A. R. Leach, *Molecular Modeling: Principles and Applications*, 2nd Edition, Pearson Education, 2001.
7. D. Hestenes, *Modeling Theory for Math and Science Education*, paper presented at the Mathematical Modeling ICTMA-13: Education and Design Sciences, 2007.
8. A. Pallant and R. Tinker, *Reasoning with Atomic-Scale Molecular Dynamic Models*, Journal of Science Education and Technology, **13**, 51-66 (2004).
9. D. Marx and J. Hutter, *Ab initio Molecular Dynamics: Theory and Implementation*, in Modern Methods and Algorithms of Quantum Chemistry, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 1, 2000.
10. Q. Xie and R. Tinker, *Molecular Dynamics Simulations of Chemical Reactions for Use in Education*, Journal of Chemical Education, **83**, 77-83 (2006).
11. H. Jang, C. K. Hall, and Y. Zhou, *Assembly and kinetic folding pathways of a tetrameric beta-sheet complex: Molecular dynamics simulations of simplified off-lattice protein models*, Biophysical Journal, **86**, 31-49 (2004).
12. S. Papert, *Situating Constructionism*, in I. Harel and S. Papert (Eds.), *Constructionism*, Ablex Publishing Corporation, Norwood, NJ, 1991.