

USING FLEXIBLE POINTS IN A DEVELOPING SIMULATION OF SELECTIVE DISSOLUTION IN ALLOYS

Joseph C. Carnahan

Department of Computer Science
University of Virginia
Charlottesville, VA 22904, U.S.A.

Erin C. Carson
Paul F. Reynolds, Jr.

Department of Computer Science
University of Virginia
Charlottesville, VA 22904, U.S.A.

Steven A. Policastro

Department of Materials Science and Engineering
University of Virginia
Charlottesville, VA 22904, U.S.A.

Robert G. Kelly

Department of Materials Science and Engineering
University of Virginia
Charlottesville, VA 22904, U.S.A.

ABSTRACT

Coercion is a semi-automated simulation adaptation technology that uses subject-matter expert insight about model abstraction alternatives, called flexible points, to change the behavior of a simulation. Coercion has been successfully applied to legacy simulations, but never before to a simulation under development. In this paper, we describe coercion of a developing simulation and compare it with our experience coercing legacy simulations. Using a simulation of selective dissolution in alloys as a case study, we observe that applying coercion early in the development process can be very beneficial, aiding subject matter experts in formalizing assumptions and discovering unexpected interactions. We also discuss the development of new coercion tools and a new language (Flex ML) for working with flexible points.

1 INTRODUCTION

Developing any good simulation model is a matter of selecting the right abstractions. By definition, models are abstractions of other systems. Therefore, decisions must be made about which phenomena to omit, which to include, and how to represent the included phenomena. The right abstraction decisions lead to a simulation that runs efficiently and provides valid insight into questions of interest.

Often there exists uncertainty about which abstractions are best for particular questions of interest, and finding the right abstractions for a model can be a significant research problem in itself. This uncertainty about abstractions makes modeling challenging, but it also provides flexibility that can be exploited during model development. COERCE is a simulation adaptation technology that exploits this flexibility

to automate parts of the adaptation process ([Waziruddin, Brogan, and Reynolds 2003](#)). In this paper, we show how COERCE can be applied early in the simulation life cycle to build better simulations and gain additional insight during the development process.

This paper describes the development of a simulation of selective dissolution in alloys. When an alloy of two or more metals of different activities (such as gold and silver) is submerged in an electrolyte and a voltage is applied, the more active metal may dissolve into the electrolyte while some of the less active metal may diffuse into spaces vacated by the more active metal. The result of this dealloying process is a nanoporous structure with extremely high surface area and other useful properties ([Erlebacher 2004](#)).

To simulate the dealloying process, abstraction decisions must be made about the bonding energies between different atoms in the system and about how to model concurrent dissolution events. By identifying and formally describing these decisions as flexible points, we were able to use coercion to explore possible abstractions and identify previously unknown relationships within the system. This approach succeeded in spite of (and in some cases, because of) the challenges of coercing a simulation early in its development.

2 SIMULATION OF SELECTIVE DISSOLUTION

Selective dissolution (also known as *dealloying*) refers to the preferential removal of one component of a metal alloy over another via corrosive processes ([Korb 1987](#)). For selective dissolution to occur, one or more of the metals in an alloy must be more reactive than the other(s). When the alloy is exposed to an electrolytic solution, the more reactive metal

or metals may dissolve into the solution, leaving the other metal or metals behind.

There are several common environmental processes in which dealloying occurs, such as the “dezincification” of brasses. In this instance, the zinc is preferentially removed, leaving behind a weakened copper skeleton. Dealloying processes have become a matter of increased interest recently, especially in alloys such as $\text{Au}_x\text{Ag}_{1-x}$, because the resulting porous gold structure could be used for several engineering applications including drug delivery and sensor foundations.

The underlying assumption of the dealloying model proposed in several recent papers is that both the dissolution rate of the more reactive component of the alloy and the surface diffusion rates of both components are dependent on the coordination number of the bonds of the atoms at the surface (Erlebacher et al. 2001; Legrand et al. 2002). Initially, the rates of diffusion and dissolution were derived empirically as best fit parameters in comparison with experimental data. The model given in Legrand et al. (2002) was later extended with first principle calculations using quantum chemistry to obtain more accurate estimates of the energies of dissolution (Diawara et al. 2004).

In this work, we have modeled the dealloying process by considering the role of the electrolyte as the medium by which both the high observed surface diffusivities of the more noble (less active) atoms can be explained and through which the cations of the less noble (more active) component are transported away from the dealloying front. We adopted a kinetic Monte-Carlo algorithm because it allows us to model fairly large systems for longer periods of time than a standard Monte-Carlo algorithm, and the natural processes that we are modeling lend themselves easily to this approach (Novotny 2001).

3 COERCE: SEMI-AUTOMATED ADAPTATION

To assist in the development of the dealloying simulation, we employed a technology called COERCE. COERCE consists of two parts, coercion and coercibility. *Coercion* is a semi-automated adaptation process that leverages subject matter expert (SME) insight about elements of the simulation code called flexible points to direct a simulation’s behavior. A *flexible point* is the reflection of a model abstraction in the simulation code, and a *flexible point binding* is the implementation of a possible abstraction that could be chosen at a flexible point. *Coercibility* requires the description of flexible points, their properties, their possible bindings, and their interactions.

3.1 The Coercion Process

The coercion process is sketched in Figure 1. The adaptation begins with the introduction of a new requirement

on the simulation’s behavior. Using insight about which abstractions affect each aspect of the simulation’s behavior, the SME selects flexible points that relate to the new requirement. From here, the simulation analyst must choose either automatic search or manual modification for changing the selected flexible points. If automatic search is selected, then the SME specifies constraints and an objective function to describe the required behavior. Then, an optimization technique is applied to search for the set of flexible point bindings that minimize the objective function. After the modification is performed or the optimization is run, the simulation’s behavior is analyzed and the SME determines if the result is acceptable. If not, then insight is gained from the new behavior and the process is repeated. This insight can include identification of new flexible points, observation of the effect of selected flexible points on simulation behavior, or discovery of interaction between flexible points and their effects. This continual accumulation of insight is critical to the success of the coercion process (Waziruddin, Brogan, and Reynolds 2004). Because coercion employs automatic search instead of manual modification, coercion yields a simulation that performs the required behavior with considerably less programming effort than conventional simulation development methods.

Coercion can be distinguished from other optimization-based simulation methodologies in several ways. First, flexible points include not only numeric parameters but also design decisions, such as selecting an equation for representing a phenomenon or omitting the phenomenon from the simulation altogether. Second, coercion does not require identifying all parameters in advance, which would often make the simulation too complex and interfere with the simulation’s performance (Parnas 1979). Instead, new flexible points can and should be identified at any point in a simulation’s life cycle. Most importantly, coercion takes advantage of SME insight about the flexible points and their interactions to guide the search process, which can significantly reduce the costs of search and improve search results relative to naive optimization methods (Tang and Reynolds 2007).

3.2 Describing Flexible Points

Flexible points are the key to COERCE. Because flexible points reflect model abstraction decisions, they have a degree of flexibility derived from uncertainty about what the right abstractions should be. However, SME knowledge about the phenomenon that is being modeled can also constrain the possible abstractions that could be selected. This combination of flexibility and constraints makes it possible to frame simulation adaptation as an optimization problem, which in turn allows automatic search tools to replace certain manual code modifications.

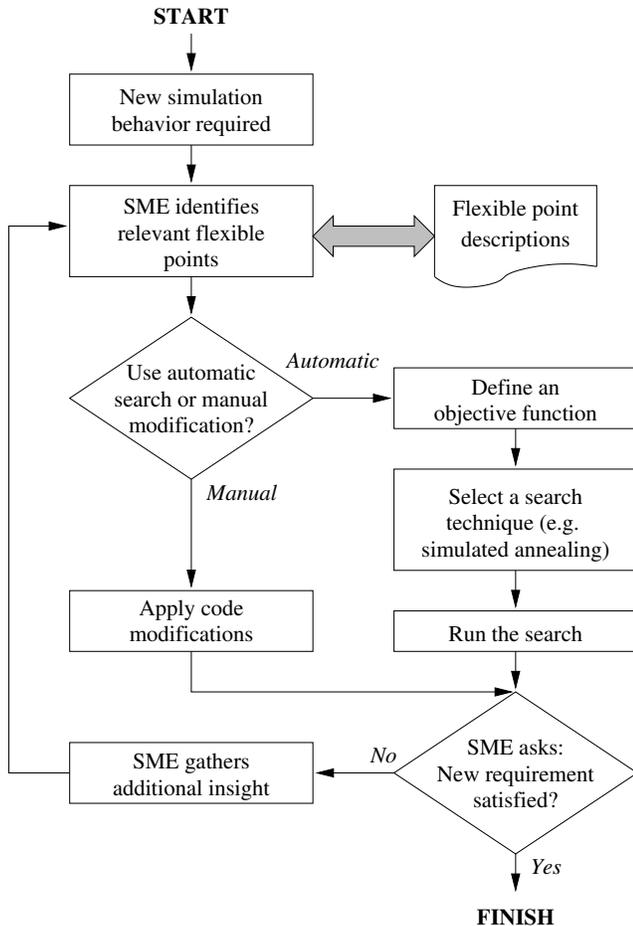


Figure 1: The coercion process.

Flexible points commonly include (but are not limited to) parameters, input distributions, and choices of equations. They can be identified at any point in the simulation's life cycle, including conception, design, implementation, validation, and adaptation (Carnahan, Reynolds, and Brogan 2004). The following information about flexible points is useful for coercion:

Possibly valid bindings. Before exploring the possible bindings for a flexible point, coercion requires information about what other bindings should be considered.

How each binding could be implemented. For flexible point changes to be applied automatically, coercion requires knowledge about how changing a flexible point binding changes the code.

Effects of each binding on simulation behavior. In order to know which flexible points should be changed to meet a specific requirement, coercion requires knowledge about the effects of each possible change.

Interactions among different flexible points. As flexible points can overlap with each other in the

code and change each other's effects, coercion requires information about how different flexible points affect one another.

Information about flexible point interactions is particularly important. When one part of the simulation depends on assumptions made in another part, changing flexible points without considering interactions can have unexpected effects or lead to invalid results. Examples of flexible point interactions include

- restrictions on which bindings can be selected simultaneously,
- requirements on which bindings must be selected if a given binding is selected,
- complementary effects, where changing two flexible points leads to a much greater effect on simulation behavior than selecting either in isolation,
- canceling effects, where changing two flexible points leads to a smaller effect than selecting either in isolation, and
- implementation conflicts, where selecting one binding changes how another binding must be implemented.

To capture all of this information and to process it with automatic coercion tools, a formal language for describing flexible points is needed. A language called Flex ML is currently being developed to fulfill this need. Flex ML (short for "Flexible Point Markup Language") is an XML-based language for describing flexible points in simulations. Its first use in COERCE is described in Gore et al. (2007). Note that even with a formal language, flexible point descriptions will always be incomplete, as it is impossible to anticipate all of the future requirements that might be placed on any given simulation. However, numerous examples have shown that coercion can succeed even with incomplete information about flexible points.

3.3 COERCE Examples

COERCE has been applied successfully to a variety of simulation adaptation problems. Drewry, Reynolds, and Emanuel (2002) describe the coercion of a simulation of carbon dioxide uptake in forests, which explored flexible points in a low-resolution model to yield behavior that more closely reflected the behavior of a trusted higher-resolution model. Carnahan, Reynolds, and Brogan (2003) describe the coercion of a simplified model of a bicyclist to mimic the behavior of a detailed full-physics simulation. Carnahan and Reynolds (2006) discuss the coercion of a logistical supply-chain simulation in response to new marketing data and inventory reports.

In each of these cases, flexible points were identified and automatic search was employed to find flexible point bindings that yielded the desired simulation behavior. Notably, each of these cases involved an existing (“legacy”) simulation, where the simulation code was already written and at least partially validated before coercion. This paper reports on the first effort to apply COERCE to a developing simulation.

4 CASE STUDY METHODOLOGY

To evaluate the benefits of applying COERCE to the early development of a simulation, we identified flexible points and performed coercion on the selective dissolution simulation introduced in Section 2. This required identifying the critical assumptions underlying the new simulation, gathering SME insight about flexible points and what reasonable bindings might be, and continually integrating changing baseline simulation code with the adaptations achieved through coercion. The results included demonstrably useful estimates for a number of uncertain simulation parameters, some surprising insights about interactions within the simulation, and several promising directions for expanding the simulation in the future.

4.1 Flexible Point Identification and Analysis

As with previous coercion experiments (e.g., [Carnahan, Reynolds, and Brogan 2003](#)), identification of flexible points in the dealloying simulation required cooperation between the simulation analysts and the SME. The analysts, once familiar with the simulation code, compiled an initial list of potential flexible points. This list included any parameters or code that seemed important to the function of the model. While not a complete or necessarily correct list of the flexible points in the simulation, it served as a starting point for discussions with the SME. As the SME was previously unfamiliar with COERCE, the tentative flexible point list helped to convey the idea of flexible points, which made it easier for the SME to identify additional (and often more useful) flexible points to use in coercion.

To acquire more information about flexible points, the analysts asked the SME the following questions:

1. What numerical value, code, or distribution is currently being used as the binding for each flexible point?
2. What is the explanation for each flexible point’s current binding? Would different SMEs use different parameter values or modeling approaches based on incomplete experimental data, or is the binding based on data that is widely accepted in the field?
3. How certain is the SME about the current binding for each flexible point? (For this study, we used a subjective scale of “certain”, “uncertain”, and “unknown”. Future work will address how quantitative uncertainty analysis techniques could be used instead.)
4. If the SME is uncertain about the current binding of a flexible point, what range of values or alternative approaches might be considered acceptable?
5. Which other bindings for this flexible point should be considered first?
6. What variables in the code should be directly affected by changing each flexible point?
7. What other flexible points are affected by changing each flexible point, and how?

A representative subset of the information gathered in this analysis is presented in Table 1.

4.2 Formalized Descriptions with Flex ML

Once a good working set of flexible points had been identified, the analysts translated the information into a formal representation using Flex ML. As mentioned in Section 3.2, Flex ML is a developing language for describing flexible points and all of their properties that are relevant to COERCE. We have developed a number of tools for Flex ML, including a graphical editor, an XML Schema for detecting errors in Flex ML documents, and a growing library of optimization routines that can automatically exploit information about flexible points and potential flexible point interactions.

The formal nature of Flex ML enables automated analysis and exploration of simulation implementation alternatives. When coercing a simulation to exhibit certain behaviors, the COERCE optimization process can limit the search to exploration of those flexible points that potentially affect the desired behavior. Likewise, the process could explore those flexible points with bindings marked as less certain before exploring those with bindings considered more certain. Whenever two flexible points are marked as potentially interacting in a contradictory manner, the optimization process could avoid changing those flexible points or flag the result of any coercion involving those flexible points as possibly suspect.

Several optimization algorithms could take advantage of the flexible point descriptions in algorithm-specific ways ([Tang and Reynolds 2007](#)). For example, simulated annealing takes steps in random directions through the space of possible bindings, but flexible point information could be used to bias that random choice in favor of directions that the SME wishes to explore first. Also, both simulated annealing and steepest descent have an initial step size pa-

Table 1: Sample of questions and answers in the flexible point analysis.

<i>Flexible Point</i>	<i>Question</i>	<i>SME Answers</i>
Au-Au bond energy	1. What is the default binding? 2. How was this binding chosen? 3. How certain is this binding? 4. How could this binding change? 5. How should this binding be changed first? 6. What variables does this effect? 7. What flexible points does this effect?	0.64 eV Based on others' experimental work Uncertain due to extrapolation from original experiment Must be greater than the Au-Ag bond energy More likely that decreasing this value will be beneficial than increasing Diffusivity and # of atoms dealloyed for Au Could counteract increases in Au-H ₂ O or Au-Anion bond energies
Factor for voltage-affected diffusion	1. What is the default binding? 2. How was this binding chosen? 3. How certain is this binding? 4. How could this binding change? 5. How should this binding be changed first? 6. What variables does this affect? 7. What flexible points does this affect?	0.0 The SME wished to explore other parameters' effects on diffusion first Completely unknown (no certainty) Must be in the range $0.0 \leq x \leq 1.0$ Must increase Should affect the diffusivity of Au atoms Changes bond energies' effect on diffusivity
Time advance mechanism	1. What is the default binding? 2. How was this binding chosen? 3. How certain is this binding? 4. How could this binding change? 5. How should this binding be changed first? 6. What variables does this effect? 7. What flexible points does this affect?	As each event occurs, advance the clock by a fixed amount based on the average of all rates of all possible events Comparison w/ Erlebacher (2004) Uncertain due to added consideration for electrolyte Time could advance as inverse of the rate of the selected event, or some compromise could be chosen Not sure – Avoid changing this Locations and #s of cations and anions Not sure

parameter, which can be set based on the units associated with each flexible point.

4.3 Flexible Points and the Dealloying Simulation

We used coercion to improve the dealloying simulation in several ways. For example, the SME observed early in the simulation development process that some gold atoms were dissolving along with the silver atoms. When dealloying an Au_xAg_{1-x} alloy, it is possible that a few gold atoms could be dissolved from the crystal, but the applied voltages are low enough that gold should not be ionized and should only diffuse across the surface of the crystal. At the same time, the silver atoms are expected to dissolve into the electrolyte. To capture these requirements, we stated the following objective function:

$$\begin{aligned}
 u &= \# \text{ of Au atoms dissolved} \\
 g &= \# \text{ of Ag atoms dissolved} \\
 D &= \text{Computed diffusivity of Au atoms in the simulation} \\
 D_{target} &= \text{Measured diffusivity of Au atoms in experiments} \\
 s &= \text{Scaling factor} \\
 \text{Objective Function} &= \begin{cases} |D_{target} - D|s + u & \text{if } u > 0 \\ |D_{target} - D|s - g & \text{if } u = 0 \end{cases}
 \end{aligned}$$

In other words, if any gold atoms dissolved, then the objective function was the number of dissolved gold atoms plus a weighted measurement of the accuracy of the gold's diffusivity. If none of the gold atoms dissolved, then the objective function became the accuracy of the gold's diffusivity minus the number of silver atoms that dealloyed. Using COERCE to minimize this function, the SME and analysts derived a simulation that dissolved silver without dissolving gold, while realizing a diffusivity acceptably close to the experimentally-predicted value.

4.3.1 Diffusivity Calculations

The value used for D_{target} was $100 \text{ \AA}^2/\text{s}$ ($10^{-14} \text{ cm}^2/\text{s}$). Within the simulation, the diffusivity of gold was calculated as follows:

$$\begin{aligned} d &= \text{Avg. distance moved by} \\ &\quad \text{Au atoms (\AA)} \\ t &= \text{Elapsed simulation time (s)} \\ n &= \# \text{ of elapsed simulation} \\ &\quad \text{steps} \\ \lambda &= \frac{d}{n} \\ D &= \frac{n\lambda^2}{2t} \end{aligned}$$

The scaling factor s was calculated from the size of the simulated sample that was being dealloyed:

$$\begin{aligned} x &= \text{Width of the sample} \\ y &= \text{Depth of the sample} \\ n &= \# \text{ of elapsed simulation} \\ &\quad \text{steps} \\ s &= \frac{n}{5000} \left(\frac{x}{10}\right) \left(\frac{y}{10}\right) \end{aligned}$$

The purpose of this scaling factor was to keep the difference in diffusivities comparable with the number of atoms dissolved. As the number of atoms dissolved increased with the area of the interface between the sample and the electrolyte (roughly $x \times y$) and the duration of the simulation (n), the weight of the diffusivity component in the objective function increased with each of these three factors. The numbers 5000, 10, and 10 were derived using trial and error on the dealloying simulation with its default (pre-coercion) flexible point bindings.

4.3.2 Coercion Results

The progress of the coercion itself is depicted in Figure 2. The flexible points that the SME selected for this coercion were the factor for voltage-affected diffusion, Au-Au bond energy, Au-Ag bond energy, Au-H₂O bond energy, Au-Cation bond energy, Au-Anion bond energy, Ag-Ag bond energy, Ag-H₂O bond energy, Ag-Cation bond energy, and Ag-Anion bond energy. After exploring the space of possible bindings, the optimizer converged on the changes given in Table 2.

Of these changes, the Au-Au, the Au-H₂O, the Ag-Ag, and the Ag-Cation bond energy changes were the most interesting, because they suggest possible areas for continued exploration of the physics of the interactions between the electrolyte and the crystal. For instance, the SME previously assumed that cations should diffuse away from the crystal and not have a significant effect on the dissolution and diffusion of the atoms in the crystal. However, because the coercion results indicated that changing the Ag-Cation bond energy was beneficial, cations must be affecting the dissolution of

Table 2: Flexible point binding changes.

<i>Flexible Point</i>	<i>Previous Binding</i>	<i>New Binding</i>
Au-Au bond energy	0.64 eV	0.74 eV
Au-Ag bond energy	0.56 eV	0.535 eV
Au-Anion bond energy	0.25 eV	0.15 eV
Au-H ₂ O bond energy	0.18 eV	0.074 eV
Ag-Ag bond energy	0.48 eV	0.255 eV
Ag-Anion bond energy	0.25 eV	0.275 eV
Ag-Cation bond energy	0.0 eV	0.025 eV

silver atoms in this simulation. This could mean that cations have more effect than previously thought, or it could indicate an error in another part of the simulation. To investigate this, the SME suggested investigating the concentration of cations and how cation diffusion is modeled in order to find an explanation for this surprising behavior.

4.3.3 Additional Flexible Points and Future Coercion

The results of coercing the dealloying simulation indicated other parts of the simulation that should be explored. This exploration can also be accomplished through coercion using several other flexible points.

Time advancement mechanism – As mentioned in Table 1, deciding how to advance the simulation clock is a flexible point in this simulation. Different Monte Carlo algorithms handle simulation time in different ways. One approach is to advance the clock with each event by the inverse of the rate associated with that event. This approach works well when all events operate on the same time scale. However, in the dealloying example, dissolution and diffusion events for the different atoms occur at rates several orders of magnitude apart. The dealloying simulation currently advances the clock by a fixed amount that is a weighted average of the inverses of the rates of the different events. As certain time-dependent processes in the simulation are behaving strangely (such as cation diffusion), other bindings for the time advancement flexible point should be considered.

Ion diffusion – Cation and anion diffusion in the dealloying example are currently modeled differently from gold and silver diffusion and dissolution. At each time step, each ion has a certain probability of swapping with one of its neighboring ions or water molecules. The probabilities depend on the charge of the ion, the direction of the neighbor, the direction of the applied voltage, and the current concentration of cations and anions in the electrolyte. One important question for this simulation is how the concentration of anions and cations should be maintained, as it is not known how much concentrations near the surface might differ from concentrations in the bulk electrolyte. Exploring this flexible point and considering different algorithms for

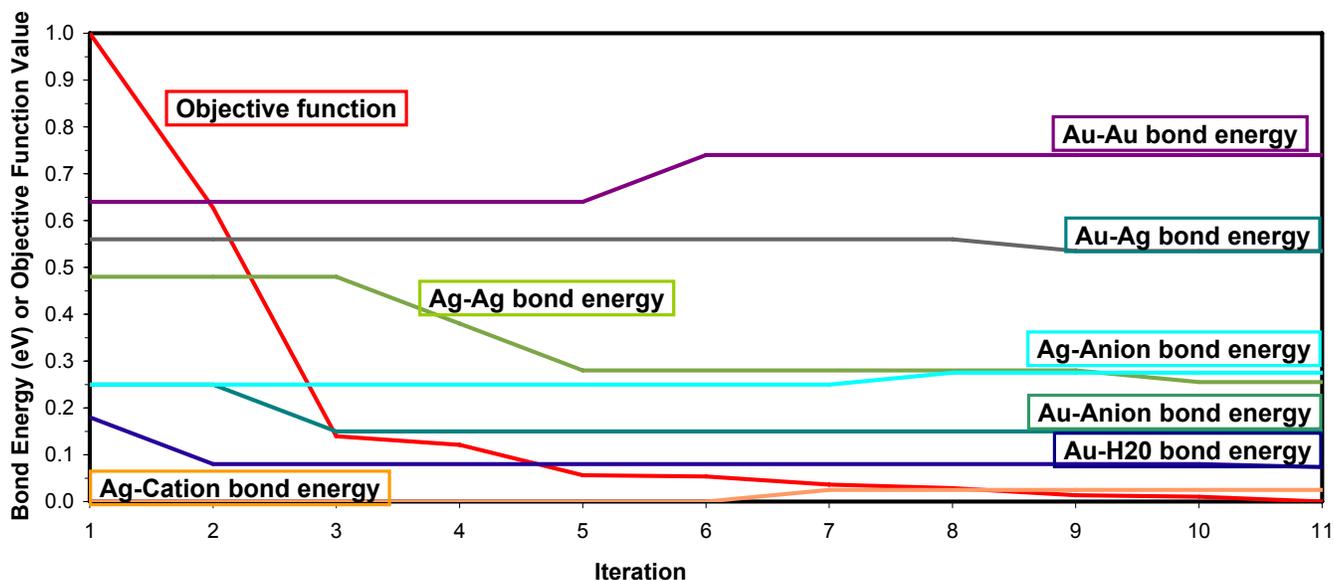


Figure 2: Coercing the dealloying simulation. Simulation was run with a $15 \times 15 \times 15$ crystal with 70% Ag and 30% Au for 10000 time steps per run. Objective function has been normalized to a 0–1 scale.

ion diffusion will clarify whether the observed effect of cations on silver dissolution is valid.

Bond energies in ternary alloys – Another direction for expanding this simulation is to consider ternary alloys, such as mixtures of gold, silver, and copper. Introducing another metal into the alloy means estimating bond energies between atoms of the new metal and the other atoms, molecules, and ions in the simulation. The simulation code has already been extended to support representation of ternary alloys. However, additional experimental data about ternary alloy dealloying are needed to provide constraints and an objective function for coercion of a ternary alloy simulation.

5 DISCUSSION

The selective dissolution simulation is still under development. However, several important lessons and benefits have become evident from applying COERCE to this developing simulation.

5.1 Improvements to Flex ML

As discussed in Section 4.2, the flexible points in this simulation were described using our Flex ML language. We discovered several ways to improve Flex ML. First, the dealloying simulation included several cases where the bound on one flexible point's bindings was actually the binding for another flexible point. For example, because gold is known to be less active than silver, the Au-Ag bond energy was constrained to be less than the Au-Au bond energy and greater than the Ag-Ag bond energy. To reflect this, we added the ability to specify bounds in Flex ML in terms of the bindings and bounds of other flexible points.

The selective dissolution simulation also highlighted the need for supporting *compound flexible points*, which are flexible points that consist of other flexible points. Flex ML can describe several classes of flexible points, such as algorithm selections, integer parameters, and real-valued parameters. However, certain flexible points involved both the selection of an algorithm and the setting of certain parameters. For example, picking a binding for the time advancement flexible point involves selecting an equation for calculating time step sizes and setting parameters that govern how different events' rates factor into the time step size. To describe flexible points such as the time advancement mechanism, Flex ML must include support for compound flexible points.

5.2 Integration with the Evolving Simulation

In our study, a few challenges arose from the fact that the dealloying simulation was still under development. In particular, it was occasionally difficult to determine if unacceptable simulation behaviors were the result of incorrect modeling choices or programming errors in how the modeling choices were implemented. COERCE assumes that the simulation actually implements the model abstractions that have been selected. With a previously validated legacy simulation, that assumption is reasonable, but with an untested new simulation, that assumption does not generally hold.

Fortunately, several developing COERCE-related technologies address the issue of validating evolving simulations. *Lightweight validation* applies static analysis tools and automatic search to efficiently determine if changing the simulation has violated any correctness properties that the SME identified. This is less expensive than a full validation,

making it economical to apply earlier in the the development of a simulation (Liu, Reynolds, and Brogan 2006). *Emergent behavior exploration* uses coercion to actively explore unexpected simulation behaviors, finding the limitations on when unexpected behaviors occur and allowing the SME to check if the simulation behaves as expected under better understood conditions (Gore et al. 2007).

5.3 Benefits of the Early Use of COERCE

Applying COERCE early in the life of the dealloying simulation provided a number of benefits relative to coercing a legacy simulation. One issue in previous coercion experiments has been a reluctance of SMEs or simulation users to reconsider certain abstraction decisions in their models. In the carbon dioxide uptake example, SMEs disagreed about whether a parameter to Beer's Law was a valid flexible point, even though changing the parameter slightly led to dramatic improvements in the simulation's behavior (Drewry, Reynolds, and Emanuel 2002). Likewise, an attempt to coerce a model of thermoacoustic coupling in combustion reactions failed when only parameters were allowed to change and the combustion equations themselves were ruled out as possible flexible points (Zambon 2005). This is primarily a social problem, but it affects the technical success of coercion.

Because it is still under development, every part of the dealloying simulation was considered eligible for identification as a flexible point. The materials science SME permitted the analysts to treat almost any parameter as a flexible point, even those with values derived from experiments. For example, the Au-Au bond energy inside a bulk crystal was reported by Howe (1997) to be 0.64 eV. However, the SME observed that at the interface, when atoms have some vacancies among their neighbors, the energies of the remaining bonds may be stronger or weaker than the bond energies inside the bulk crystal. Through coercion, we discovered that raising the Au-Au bond energy was an effective way to prevent Au atoms from dissolving under circumstances when they were not expected to dissolve. This suggests that increased bond energies at the surface may play a significant role in determining the critical voltage for gold. Without a willingness to question modeling assumptions, this discovery might not have occurred.

6 CONCLUSION

We have successfully applied COERCE to a developing simulation of selective dissolution of gold-silver alloys. In the process, we identified a variety of important flexible points in the simulation and made a number of improvements for representing flexible points in Flex ML. Using these flexible points, we coerced the dealloying simulation to eliminate undesired behaviors and generate more real-

istic results. This coercion was successful in spite of the instabilities associated with a newly-developed simulation. In fact, the coercion process benefited from the added flexibility that comes from working with a newly-developed simulation.

ACKNOWLEDGMENTS

We gratefully acknowledge support from the DDDAS program at the National Science Foundation (ITR 0426971), the NSF-NIRT grant, award number 0507023, and from our colleagues in MaSTRI, the Modeling and Simulation Technology Research Initiative at the University of Virginia.

REFERENCES

- Carnahan, J. C., and P. F. Reynolds. 2006, December. Requirements for DDDAS flexible point support. In *Proceedings of the 2006 Winter Simulation Conference*, ed. L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, , and R. M. Fujimoto. Institute of Electrical and Electronics Engineers, Inc.
- Carnahan, J. C., P. F. Reynolds, and D. C. Brogan. 2003, December. An experiment in simulation coercion. In *Proceedings of the 2003 Interservice/Industry Training, Simulation, and Education Conference*. Arlington, Virginia: National Training Systems Association.
- Carnahan, J. C., P. F. Reynolds, and D. C. Brogan. 2004, September. Language constructs for identifying flexible points in coercible simulations. In *Proceedings of the 2004 Fall Simulation Interoperability Workshop*. Orlando, Florida: Simulation Interoperability Standards Organization.
- Diawara, B., M. Legrand, J.-J. Legendre, and P. Marcus. 2004. Use of quantum chemistry results in 3d modeling of corrosion of iron-chromium alloys. *Journal of the Electrochemical Society* 151 (3): B172–B178.
- Drewry, D. T., P. F. Reynolds, and W. R. Emanuel. 2002, December. An optimization-based multi-resolution simulation methodology. In *Proceedings of the 2002 Winter Simulation Conference*, ed. E. Yücesan and C.-H. Chen. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Erlebacher, J. 2004. An atomistic description of dealloying: Porosity evolution, the critical potential, and rate-limiting behavior. *Journal of the Electrochemical Society* 151 (10): C614–C626.
- Erlebacher, J., M. J. Aziz, A. Karma, N. Dimitrov, and K. Sieradzki. 2001. Evolution of nanoporosity in dealloying. *Nature* 410 (6827): 450–453.
- Gore, R. J., P. F. Reynolds, L. Tang, and D. C. Brogan. 2007, June. Explanation exploration: Exploring emergent behavior. In *Proceedings of the 21st Workshop*

on *Principles of Advanced and Distributed Simulation (PADS)*. Association for Computing Machinery.

- Howe, J. M. 1997. *Interfaces in materials*. New York, NY: John Wiley & Sons.
- Korb, L. J. 1987. *Corrosion*, Volume 13 of *ASM Handbook*, 131. ASM International.
- Legrand, M., B. Diawara, J.-J. Legendre, and P. Marcus. 2002, April. Three-dimensional modelling of selective dissolution and passivation of iron-chromium alloys. *Corrosion Science* 44 (18): 773–790.
- Liu, X., P. F. Reynolds, and D. C. Brogan. 2006, June. Using abstraction in the verification of simulation coercion. In *Proceedings of the 20th Workshop on Principles of Advanced and Distributed Simulation (PADS)*, 119–128. Association for Computing Machinery.
- Novotny, M. A. 2001, September. A tutorial on advanced dynamic monte carlo methods for systems with discrete state spaces. *Annual Reviews of Computational Physics IX*:153–210.
- Parnas, D. L. 1979, March. Designing software for ease of extension and contraction. *IEEE Transactions on Software Engineering* SE-5:128–138.
- Tang, L., and P. F. Reynolds. 2007, December. Agile optimization for coercion. In *Proceedings of the 2007 Winter Simulation Conference*, ed. S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Waziruddin, S., D. C. Brogan, and P. F. Reynolds. 2003, September. The process for coercing simulations. In *Proceedings of the 2003 Fall Simulation Interoperability Workshop*. Orlando, Florida: Simulation Interoperability Standards Organization.
- Waziruddin, S., D. C. Brogan, and P. F. Reynolds. 2004, September. Coercion through optimization: A classification of optimization techniques. In *Proceedings of the 2004 Fall Simulation Interoperability Workshop*. Orlando, Florida: Simulation Interoperability Standards Organization.
- Zambon, A. C. 2005. *Modeling of thermo-acoustic instabilities in the counterflow flames*. Ph. D. thesis, University of Virginia, Charlottesville, VA.

AUTHOR BIOGRAPHIES

JOSEPH C. CARNAHAN is a Ph.D. Candidate in Computer Science and a member of MaSTRI at the University of Virginia. His interests include software engineering and programming language design as they apply to the challenges of flexible and reusable simulation development. He earned his B.S. in Computer Science at the College of William and Mary and previously worked as a scientist at the Naval Surface Warfare Center Dahlgren Division. His email address is carnahan@virginia.edu.

STEVEN A. POLICASTRO is a Ph.D. Candidate in Materials Science and a graduate researcher in the Center for Electrochemical Science and Engineering at the University of Virginia. His research interests include computational materials science, specifically in the areas of electrochemistry and phase transformations. He received his M.S. in Materials Science from the University of Virginia. His email address is sab8p@virginia.edu.

ERIN C. CARSON is an undergraduate computer science major and materials science minor at the University of Virginia and a member of MaSTRI. Her email address is ecc2z@virginia.edu.

PAUL F. REYNOLDS, JR. is a Professor of Computer Science and a member of MaSTRI at the University of Virginia. He has conducted research in modeling and simulation for over 25 years, and has published on a variety of M&S topics including parallel and distributed simulation, multiresolution models and coercible simulations. He has advised numerous industrial and government agencies on matters relating to modeling and simulation. He is a plank holder in the DoD High Level Architecture. His email address is reynolds@virginia.edu.

ROBERT G. KELLY is a Professor of Materials Science and Engineering at the University of Virginia. He has been conducting research on the corrosion of metals for the past 20 years, and he is a Fellow of NACE International. He has rendered technical assistance to the NRC and DOE concerning the Yucca Mountain Project, the USAF Aging Aircraft Program, the NASA Safety and Engineering Center, and assisted with material selection for the 9/11 Pentagon Memorial design team. His email is rgkelly@virginia.edu.