SAN098-1006C
SAND--98-1006C

WinMod: An Expert Advisor for Investment Casting

CONF-980814--

Hugh P. Bivens, George A. Williamson Jr., and George F. Luger
hbivens@cs.unm.edu, gallen@cs.unm.edu, luger@cs.unm.edu
Computer Science Department, University of New Mexico, Albuquerque NM 87131

Rob G. Erdmann
erdmann@u.arizona.edu
Department of Materials Science and Engineering, University of Arizona, Tucson AZ

Michael C. Maguire, Michael D. Baldwin, and Dennis J. Anderson
mcmagui@sandia.gov, mdbaldw@sandia.gov, djander@sandia.gov
Sandia National Laboratories, Albuquerque NM 87185

Abstract

Investment casting is an important method for fabricating a variety of high quality components in mechanical systems. Cast components, unfortunately, have a large design and gate/runner build time associated with their fabrication. In addition, casting engineers often require many years of actual experience in order to consistently pour high-quality castings. Since 1989, Sandia National Laboratories has been investigating casting technology and software that will reduce the time overhead involved in producing quality casts. Several companies in the casting industry have teamed up with Sandia to form the FASTCAST Consortium. One result of this research and the formation of the FASTCAST consortium is the creation of the WinMod software, an expert casting advisor that supports the decision making process of the casting engineer through visualization and advice to help eliminate possible casting defects.

The Casting Process

To demonstrate WinMod's utility for a casting engineer, investment casting is briefly described. The first requirement of the process is the specification of the geometry of the part to be cast. A CAD file is most commonly used for this purpose. From the CAD file, a pattern of the part is generated and is used to create a ceramic shell into which the molten metal will be poured. There are two ways to create this pattern. The first uses the CAD file to produce an aluminum wax injection tool from which a wax pattern can be formed. The alternative is to generate a resin pattern directly from the CAD file using rapid prototyping (RP) technology.

After making the wax or RP pattern, a gating configuration is devised. Gates on a casting are the inlets through which molten metal enters the cast. Figure 1 shows a gated RP pattern. Next, the gated pattern is repeatedly dipped in a ceramic slurry, a sand stucco applied, and allowed to dry. Figure 2 shows a pattern being dipped in slurry. The slurry/stucco layers form a shell around the prototype. The wax or plastic is then removed from inside the shell in a process called burnout, as illustrated in Figure 3. In this process, the shell is placed in an oven and the prototype is burned out leaving a hollow shell. Molten metal is then poured into the shell and allowed to cool and solidify. The shell is then broken off leaving only the metal part. The last step is to machine off the metal extensions that were the gates of the mold.

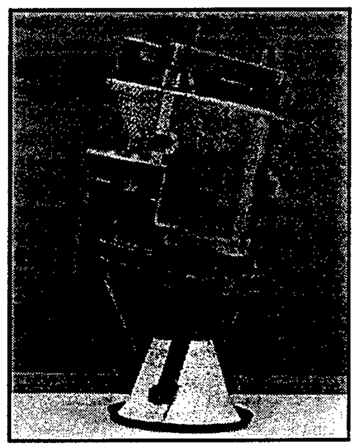

Figure 1. A wax gating of a rapid prototype (RP) pattern.

Complex part geometries and improper placement of gates for entrance of molten metal into the cast are the most common causes of casting defects. Some common defects are surface and interior shrinkage and cold shuts. Surface shrink causes irregularities to occur on the surface of the finished casting. Interior shrinkage is caused by an isolated hotspot within the casting volume. As metal surrounding a hotspot solidifies, the hotspot does not receive any more molten metal, and produces a void within the part as it cools and shrinks. Cold shuts are caused by thin sections in the part's geometry in which the metal prematurely solidifies preventing the cast from completely filling.

DISCLAIMER

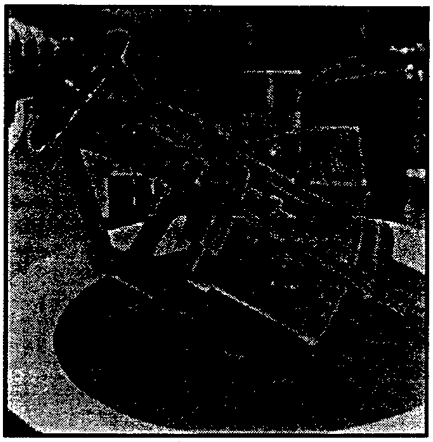

Figure 2. Dipping of RP pattern into investment slurry.

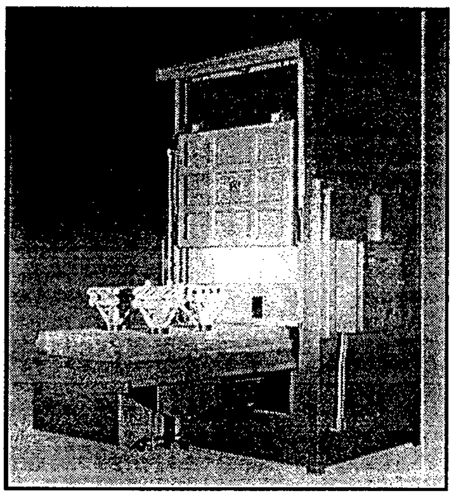

Figure 3. Flashfire burnout of RP pattern and wax gating.

What is WinMod?

The goal of creating the WinMod software is to aid the metallurgist or casting engineer with the ability to predict whether a casting will have defects. The analysis that WinMod performs can be split into two types: the geometric visualization of the part and cooling process and the rule-based decision support for optimal placement of chills, insulators, gates, and runners.

Geometric visualization, a reflection of a part's geometry, is based on the CAD representation of the part. The traditional modeling of a cooling process is represented by finite element codes reflecting fluid dynamic properties. Traditional finite-element codes, with sufficiently fine mesh design, can take hours to run. The approach used in WinMod is more heuristic, that is, an approximate, yet conservative, estimate of the transition of heat through the molten part is generated. As a result of this heuristic approach, WinMod is extremely fast, with geometric calculations generally taking only a few seconds on a Pentium 200 MHz PC.

The visualization process begins by importing the casting's geometric constraints into WinMod via a standard CAD data exchange format known as Stereo Lithography (STL) files. These files contain a list of triangles which define the surface of the part. Figure 4 shows an STL file of a manifold that has been imported into WinMod. After the STL file has been imported, a point modulus is calculated. Point modulus is the geometric algorithm used by WinMod to calculate relative cooling times throughout the part and is the basis for the diagnosis of interior shrinkage caused by hotspots.

WinMod operates by creating a three-dimensional mesh of the part. The point modulus is then used on each mesh element to calculate the average distance of 26 vectors that extend from the element's center point to the surface of the part. Each mesh element is then assigned a numeric value returned by the point modulus. This value reflects the mesh element's depth into the part and is an estimation of the element's cooling rate. Once the modulus has been calculated, WinMod represents the cooling times in the part by coloring the cooler mesh elements blue and the hotter ones red. The red areas will be the last to solidify while blue areas will solidify first. Thus, WinMod is a powerful visualization tool for detecting hotspots. The casting engineer can use the sectioning tool to "cut open" the mesh, as shown in Figure 5. The value-blanking tool strips away occluding layers of mesh elements as shown in Figure 6.

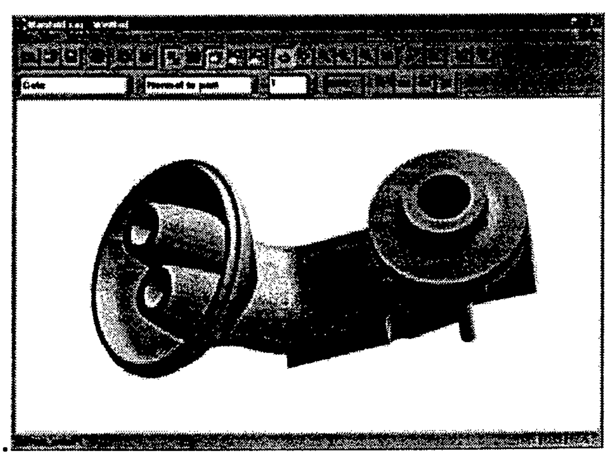

Figure 4. Shaded view of a manifold

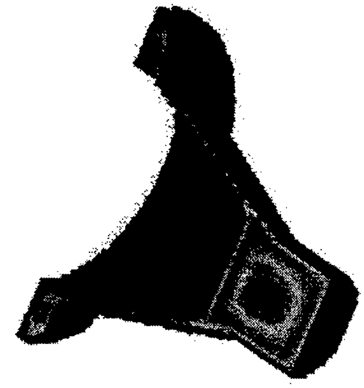

Figure 5. View of a meshed part with point modulus calculated, a sectioned view.

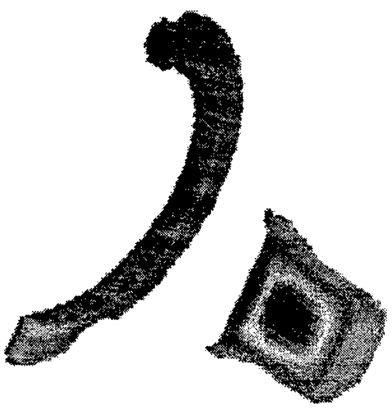

Figure 6. View of a meshed part with point modulus calculated, the value blanking view.

Finally, casting engineers can use WinMod to test different gating configurations. By placing gates on a particular area of the part, as shown in Figure 7, the point modulus algorithm includes it in its heat-cooling-time calculation. The engineer can then see how a particular gating configuration will affect the heat movement out of the casting. See the meshed gates of Figure 8. Figure 9 shows a value blanking view of the point modulus before gates have been applied. The value blanking tool has peeled away the cooler mesh elements revealing isolated hotspots within the part. The effect of the point modulus calculation with the two gates can be seen in Figure 10. The hotspots have been removed from the casting by adding the gates. WinMod's visualization supports the human expert's decision as to whether the gates need to be increased in size or even repositioned. The interface for adding gates is very similar to a two-dimensional drawing program: the user simply draws a projection of the gate onto the surface of the part and the system generates the appropriate gate.

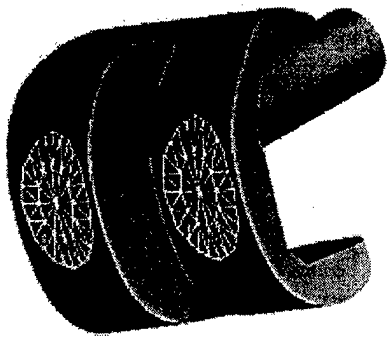

Figure 7. A view of two gates on a part. The gates here have a circular cross section and are rendered as 2-dimensional patches.

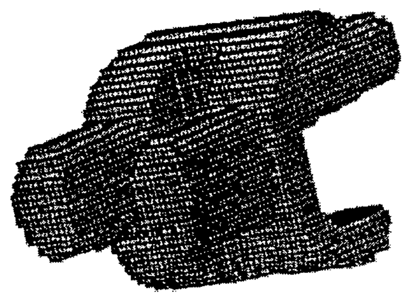

Figure 8. View of the meshed gates. The part's mesh has been extruded where the gates have been placed.

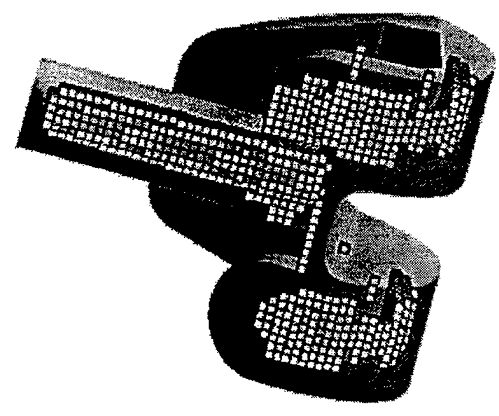

Figure 9. Point modulus visualization before adding gates.

A similar approach allows the engineer to place chills and thermal insulators on the casting's surface. These chills and insulators are placed on the outside of the shell to alter the cooling rates in the areas of the cast to which they are applied. Chills act as heat sinks, removing heat while the metal is poured, and insulators have the opposite effect, trapping heat. WinMod allows engineers to place

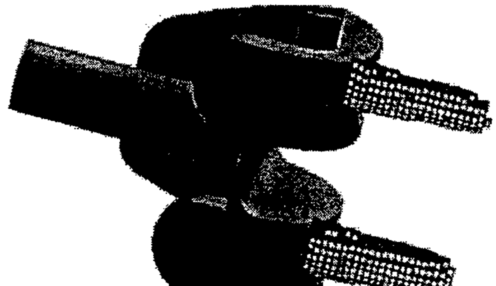

Figure 10. Point modulus visualization after adding gates. Hottest parts of the casting are in the gates.

chills and insulators on the casting and then to visualize how each of them will affect the point modulus calculation of heat transfer. The interface for placing chills and insulators is the same as for gates.

To this point we have described the visualization support that WinMod offers the casting engineer. We have also built expert recommendations into the software to help the less experienced engineer make decisions about the size and location of gates and runners, as well as the placement of chills and insulators, for the cast. Knowledge and advice from years of work with several casting experts have been implemented into WinMod. The remainder of the paper describes the system of casting properties, rule, and alloy editors that assist the expert in customizing WinMod.

The Use of Expert Knowledge in the Casting Process

The task of building knowledge and advice for the casting engineer into WinMod required the careful acquisition and analysis of the knowledge from skilled human casting experts as well as from the casting literature (Campbell 1991, Ravi 1995). Knowledge was gathered through many discussions with casting professionals from Sandia National Laboratories and other companies that make up the FASTCAST Consortium. Once this expertise was worked through and understood, it was encoded into a traditional data-driven rule-based expert system (Luger and Stubblefield 1998). WinMod uses the CLIPS expert system to deliver the rule-based analysis for each casting. CLIPS was designed at NASA Johnson Space Center to be a highly portable, low cost, easily integrated rule-based expert system (Giarratano and Riley 1989).

The CLIPS knowledge is in the form of *if... then...* statements. This modularity makes adding new knowledge relatively easy. For example, CLIPS may conclude that if two facts are true, then a third fact must also be true. The *if... then...* rules cause only changes to the known facts and do not change the meaning of any other rules or information in the system. Thus, new rules can be added without the need of rewriting any of the previously existing rules.

In addition to this easy method of adding new knowledge, CLIPS *if... then...* rules support data-driven analysis. When a casting is being analyzed, WinMod generates information based on part geometry and the alloy being used, including potential hotspots and insulator constraints. The engineer also enters specific casting properties, such as the casting temperature, sprue height and volume, and runner volume. All this information is then be used to drive the search for more information and constraints based on the *if... then...* rules of CLIPS. The data-driven aspect of the human expert's casting knowledge makes CLIPS an ideal knowledge representation and reasoning engine for WinMod.

To summarize, when analyzing a casting, the CLIPS rules are triggered by information acquired from three sources: the analysis of the part geometry, the casting engineer, and the properties relating to the alloy being used. The rules then infer new knowledge and recommendations about the casting which are then reported to the engineer. These conclusions support the decision making of the human casting engineer. Several sample CLIPS rules are presented in the next section.

A Sample of CLIPS Inference Rules

The following three CLIPS rules demonstrate how rule-based analysis works. These rules use information about the cast geometry and the desired alloy to determine if the combined area of the gates is large enough to prevent a defect in the final part. Note that each rule is made up of four components. First, the designater, `defrule`, indicates the structure is a rule; this is followed by the rule name, in the first example below, `volumetric-fill-rate`. The third and fourth components of the rule are the conditions and the actions, separated by the => symbol. The first rule contains two conditions that, when true, cause the fact describing the volume fill rate to be asserted as true.

```
(defrule volumetric-fill-rate
  (initial-fill-rate ?IR)
  (density ?Density)
  =>
  (assert (volume-fill-rate (/ ?IR
           ?Density))))
```

```
(defrule metal-velocity
  (volume-fill-rate ?VFR)
  (gate-area ?GA)
  =>
  (assert (metal-velocity (/ ?VFR ?GA))))
```

```
(defrule recommended-increase-in-gate-area
  (metal-velocity ?MV)
  (critical-velocity ?CV)
```

```
=>
(if (> (/ ?MV ?CV) 1)
  then (assert (increase-gate-area (- (*
                (/ ?MV ?CV) 100) 100)))
  else (assert (gate-area-ok))))
```

The first rule, relating to `volumetric-fill-rate`, calculates the rate at which the cast is filling with metal, based on the density of the alloy being used and the initial fill rate of the cast. The rule can be read as "If the initial fill rate is known to be equal to ?IR and the density is known to be equal to ?Density, then the volumetric fill rate is known to be equal to ?IR divided by ?Density." The ? before a symbol, such as in ?IR, indicates that IR is a variable that will take on some specific value passed into it by WinMod or by the casting engineer. The property editor, as seen in Figure 11 (top), is the interface that enables the engineer to input casting parameters (such as fill rate) into the CLIPS environment. The recommendations window (bottom, Figure 11), displays the recommendations returned from CLIPS.

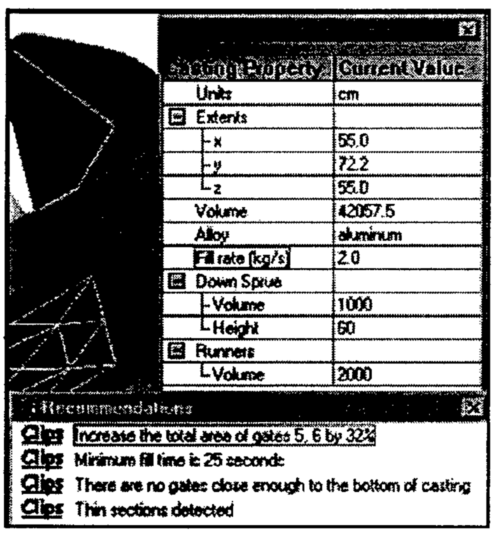

Figure 11. Casting property editor and recommendations.

Once the volumetric fill rate is calculated, it is asserted into the CLIPS environment so that it can be matched on other rules. For example, the rule `metal-velocity` matches on the volumetric fill rate and the (calculated) combined area of all the gates to determine the required velocity of the metal as it enters the cast through the gates. The last rule, `recommended-increase-in-gate-area`, uses the `critical-velocity` of the alloy and the `metal-velocity` calculated in the second rule to determine if the gate area is large enough to prevent damage to the cast. Alternatively, if the gate area needs to be increased by a certain percentage, this knowledge is conveyed to the casting engineer so that the cast can be redesigned.

WinMod is designed to be used by metallurgists and casting engineers who may not be skilled programmers. This being the case, a concerted effort was made to support the non-programmer in modifying the set of CLIPS rules. To achieve this, a rule editor was designed that allows the addition or changing of rules with a minimal number of syntax errors. In this way, the engineer needs to learn only a small subset of the CLIPS language. To support the creation and changing of rules, the editor divides a CLIPS rule into its parts, as seen in Figure 12, so the engineer can concentrate on what is to be achieved rather than on the syntax of the code. The rule's components are its name, the comment describing it, the antecedent or *if* condition, and the consequent or *then* action, as seen in Figure 12. By looking at rules already loaded in the editor, the engineer can get an understanding of how the rules work and, with only a little more effort in learning the code, can be able to edit or create new rules.

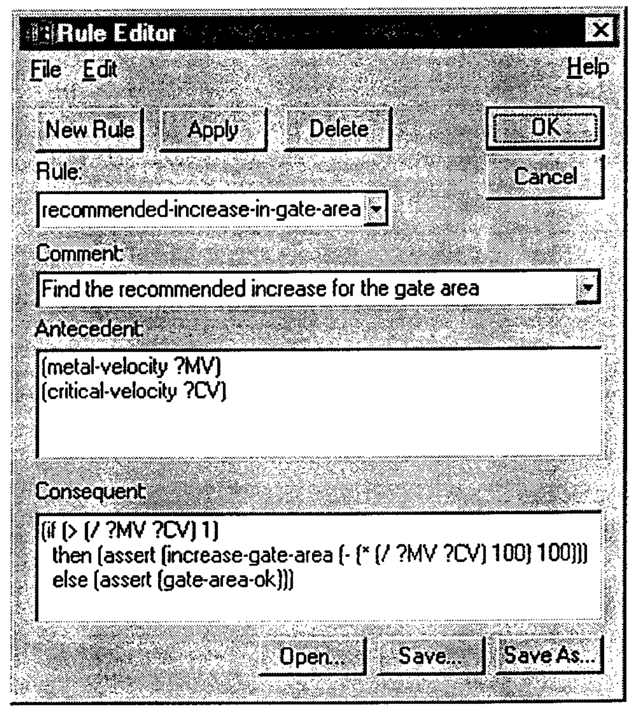

Figure 12. A view of the rule editor.

Because of the potentially large number of different alloys that can be used in casting, we built an alloy editor. This alloy editor, as seen in Figure 13, supports two functions. First, it allows the engineer in building and maintaining a large database of all potential alloys required for different castings, along with the each alloy's properties. Second, the alloy editor defines the particular CLIPS facts that are associated with an alloy's properties during each execution of WinMod. When CLIPS is run, facts describing the alloy's properties are matched with the appropriate rules to provide new alloy-specific recommendations about that cast.

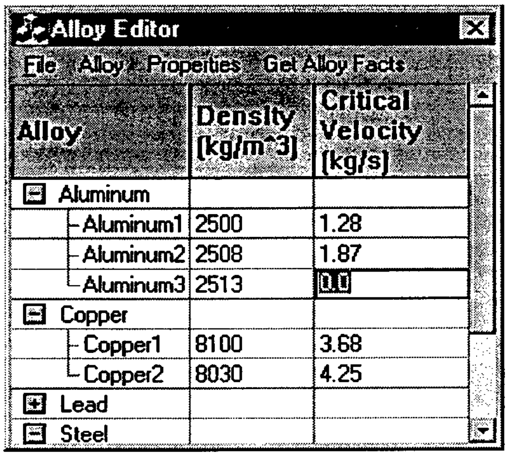

Figure 13. A View of the Alloy Editor.

Finally, after CLIPS is run, the newly acquired knowledge is in the form of a CLIPS fact set that reflects the result of the expert reasoning but may have little meaning to anyone who doesn't have extensive knowledge of the code that produced the facts. To resolve this problem, facts are associated with messages that describe what that particular fact means. The fact/message pair editor is represented in Figure 14. After CLIPS has run, WinMod examines the existing facts and tries to match them with their associated messages. A fact containing variables can be incorporated into the message to provide more information.

For example, the fact `increase-gate-area 5` means that the expert advisor recommends that the gate area of the cast be increased by 5 percent to prevent a defect from developing in the part. If the user wants a message relating to this fact be included in a final report, an entry in the fact/message pair editor must be added. The first step is to specify a pattern for the fact, such as `increase-gate-area ?percentage`. A message can then be associated with this fact that reads, "The gate area must be increased by ?percentage% to prevent damage to the cast." When the report is generated, the variable, ?percentage, will be replaced with "5" and the casting engineer will be informed of the defect.

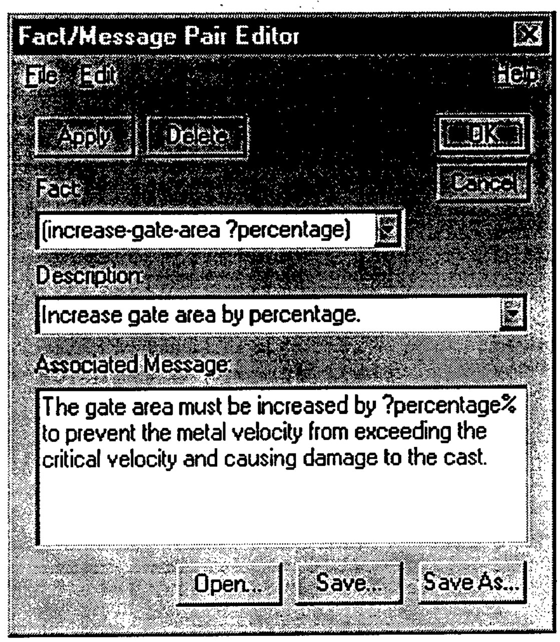

Figure 14. Picture of Fact/Message Pair Editor.

Conclusions

The goal in developing the WinMod software is to support the decision-making requirements of the casting engineer. To accomplish this goal, we developed a modeling code. CAD specifications of the part to be cast are imported into WinMod and rendered. The ability to test gate, chill, and insulator configurations is another powerful tool. A heuristic simulation of relative solidification through the part was created and the visualization of solidification enables the identification of hotspots as well as other potential problems. Because the casting of complex parts also requires determination of other critical measurements, such as alloy density and velocity along with optimal alloy casting temperature, sprue height and volume, and runner volume, a rule-based reasoning system was designed to support this decision making. Finally, since the casting engineer may not be an accomplished programmer, a system of rule and alloy editors was developed to assist in the adding and editing of rules to increase the functionality of the system.

The WinMod software system, although released to the FASTCAST consortium, is still being used as a prototype. Testing and revision of the alloy database and decision rules continues, utilizing the casting experience of experts at the Sandia National Laboratories' MeltLab, as well as the knowledge of several members of the consortium. WinMod has been used in several casting applications and its results were found to be slightly conservative, an ideal situation when the resources associated with each cast can be quite expensive.

Acknowledgments

Rob Erdmann at the University of Arizona, under Sandia support, developed the first versions of WinMod. These versions included the point modulus, meshing, and graphic visualization tools. The University of New Mexico group implemented gate, chill, and insulator placement, worked to enhance the WinMod user interface, and added extensive help features. The UNM group also added the CLIPS expert system reasoning component to WinMod, along with the casting properties, rule, and alloy editors. We are grateful for the support of the Department of Energy, Sandia National Laboratories, and especially for the support and encouragement of the member companies of the FASTCAST consortium. Sandia National Laboratories is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-ACO4-94AL85000.

References

Campbell, J. 1991. *Castings*. Jordan Hill, Oxford.: Butterworth-Heinemann Ltd.

Giarratano, J., and Riley, G. 1989. *Expert Systems: Principles and Programming*. Boston, Mass.: PWS-KENT Pub. Co.

Luger, G. F. and Stubblefield, W. A. 1998. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Reading, Mass.: Addison Wesley Longman, Inc.

Ravi, B. 1995. Intelligent Design of Gating Channels for Casting. *Materials Science and Technology* 13:785-790.

M98005724

Report Number (14) SAND--98-1006C

CONF-980814--

Publ. Date (11) 199804

Sponsor Code (18) DOE/DP, XF

UC Category (19) UC-700, DOE/ER