Software Consulting Services, LLC

Chess Computers and Classified Pagination: Smart Programming Yields Superior Results

By Richard J. Cichelli, SCS President, and William Bader, SCS Director of Research and Development

The week of May 3, 1997, "Deep Blue," a chess machine from IBM, faced off in a second match with world champion chess player Garry Kasparov. In the 1996 match, Deep Blue acquitted itself quite well, scoring two out of a possible six points. Worldwide attention was drawn to the IBM program when it unexpectedly defeated the world champion in the first round. It went on to win the match.

Deep Blue gets its chess smarts by brute force searching. Deep Blue considered millions of chess positions per second. Kasparov's searching powers are considerably less. He will examine perhaps three positions per second. Why doesn't Deep Blue's computational advantage assure its superiority? Well, the difference is quality. Deep Blue spends most of its time looking at all positions, including many poor ones, whereas Kasparov's training allows him to focus on the critical positions. Kasparov has superior pattern recognition.

Even though Deep Blue's forte is considering as many positions as possible, searching positions in best first order yields enormous efficiencies for the computer. Amazingly, the search ordering is so critical to Deep Blue's performance that it is programmed to start searching by looking at all possibilities just a few positions ahead, and then, after finding the best moves through shallow searching, it will start the search over again, each time going deeper and deeper in considering all possibilities until it is time to move. Deep Blue uses "progressive deepening" to improve its chances of looking at the most critical positions.

SCS/ClassPag™ uses "progressive repagination," which works much like "progressive deepening." Progressive repagination is one of many techniques SCS used to build a computerized classified pagination solution that provides efficient production of accurate, aesthetic output.

Efficient production is important for several reasons. Minimizing the labor involved in producing tight, accurate classified sections saves labor costs. With efficient classified pagination, the production costs involved in creating new publications are greatly reduced.

An obvious benefit of reduced classified pagination production time is the ability to extend deadlines so that ads can be sold closer to press time. An additional hour or two of selling time can yield significant revenue.

Efficient classified pagination can offer opportunities to create new products. The newspaper is likely to offer zoned classifieds as its first step into new product offerings. Zoning classifieds greatly complicates automated classified pagination programming issues. SCS/ClassPag makes doing zoned editions easy.

In an environment with efficient classified pagination, the classified pagination system becomes a data management tool for creating many new selling opportunities. This is sometimes referred to as "re-purposing" classified data. The right classified pagination solution allows newspapers to start with daily classifieds and then offer weekly consolidations that are career, real estate or auto sales guides. Efficient production allows newspapers to compete effectively with other publications which would challenge them in these markets.

Accurate output is the next critical attribute of an effective classified pagination solution. To paginate, one needs exceptional accuracy.

The first challenge to accuracy is making sure that the PostScript pages generated by the classified pagination system will image properly on a PostScript RIP. There are a surprising number of potential problems with producing accurate pages. Metro newspaper-sized classified sections ordinarily include many ads which are produced by agencies or by the advertisers themselves, each with their own methods and software.

Software Consulting Services, LLC
630 Selvaggio Drive, Suite 420
Nazareth, PA 18064
Sales: 1-800-568-8006
Fax: 610-746-7900
E-mail: sales@newspapersystems.com
www.newspapersystems.com

Fortunately, desktop publishing programs support a common imaging model. The universality of Encapsulated PostScript (EPS) facilitates the inclusion of classified display ads from multiple sources. However, accepting ads from outside sources is not without problems. Even for as rigorous a de facto standard as EPS, there are variants in its production from using Mac-, PC- and UNIX-based ad production tools, to say nothing of the incompatibilities between different versions of the same software on the same platform. PageMaker®, QuarkXPress®, Multi-Ad Creator®, etc., all have subtle differences in the EPS that they produce.

Two additional problem areas that can arise from almost any classified display ad source are getting ads the right size and getting fonts which are available on your imagesetters. There is only one production-oriented solution which adequately deals with these potential problem areas. Preflight software is necessary to validate all classified display ads. Some newspapers use a printing proofer to validate incoming ads. There are a number of problems with this option. If the image area or bounding box of a classified display ad is even a few pixels (hundredths of an inch) too big, then placing the component within the classified pagination system will create an invalid page geometry and is likely to cause the PostScript RIP to flush the job. Visual inspection is a poor substitute for automatic validation. Production PostScript RIPs are engineered for efficient throughput. Ordinarily they produce little diagnostic information when they encounter an invalid PostScript job. Thus, preflighting software needs to provide diagnostic information that leads directly to the source of the problem.

One rather subtle difference between proofing and preflighting is that the visible image area produced on a proof may not actually correspond to the image area that is generated by the EPS component. We have seen instances where an ad makeup person cropped a graphic that extended beyond the boundaries of the ad by simply pasting white space on top of the offending area. The EPS job was obviously larger than the allocated insertion order-based size. When proofed, the ad looked fine. Preflighting with a PostScript RIP showed the problem.

Fonts are another issue. For many fonts, the appearance is not protected by copyright. However, the actual name used for selecting the font is under copyright. Automated substitution of font names frequently solves missing font problems.

Of increasing concern is the arrival of ads in PDF format. PDF is an excellent technology for viewing documents or ads on screens. Even the rendering of an ad on a printer can be fairly faithful, although fonts may inappropriately be substituted and the lack of well-engineered color specifics can be a problem. Accurate translation of PDF into EPS usually requires access to insertion order data to compute the appropriate EPS bounding box. A further concern with PDF is how it's presented in conjunction with bundled PostScript RIPs. The danger of a non-PostScript image format (e.g., PDF) is that accurate rendering may be under the control of a single vendor. We think the newspaper business would welcome an open solution to the PDF issue, which can be provided by making a translator from PDF to EPS available that is both platform independent and available in source code. In our desire for efficient classified production, we found the best solution for preflighting was a high-performance diagnostic RIP that is fully scriptable. With it, nearly completely automated validation of classified display ads occurred at the rate of over 30 megabytes per minute. We use the diagnostic output to automatically adjust ad sizes, etc. At one metro newspaper, SCS/ClassPag tests over 2,500 classified display ads for each edition. SCS/ClassPag handles PDFs appropriately as well.

The other accuracy issue is satisfying advertisers' requests. Selling opportunities are increased when advertisers feel that they can make position request choices for their classified display ads. Some wish their ads to be near or touching certain classifications; others want theirs to be near or even touching certain other classified display ads. Being in a square-off or service directory is important to some. When advertisers are allowed to make requests for the placement of classified display ads, it is surprising how many will do so. The ability to make requests increases customer satisfaction and commitment, which leads to increased sales. Of course, satisfying the position requests of three or four hundred classified display ads is a task that no one wants to do manually.

Achieving excellent aesthetic results is the final important criterion of a classified pagination system. Of course, the agate ads should have flawless typography, clean layout of type, accurate

hyphenation, and precise and appropriate placement of internal graphics and logos. Since producing fully paginated pages requires the composition of what is sometimes called "page furniture" (e.g., banners, folios and indices) and these are, of course, context-sensitive, the classified pagination system must be able to handle composition. While we can accept agate ads in fully composed EPS form, we also happily translate text with markup commands into fully composed agate advertising. We do as well or better than typical front-end systems can do. What makes a page look right is the appropriate distribution of white space. If vertical justification is necessary, then it needs to be done in an aesthetically appropriate fashion. It is important that the inter-ad leading appears consistent from the first to the last page of a publication. Consistent visual presentation is the hallmark of an aesthetically produced classified section. Such consistency is almost impossible to produce through manual paste-up because each paste-up person has a different style for paginating the pages under his or her control.

When we first worked on classified pagination in the late eighties, we built a QuarkXTensions® solution to layout a classified section. Classified display ads were manually placed and the agate ads were flowed onto the page. The operator progressed through the editions page by page, fixing them one page at a time. It was agonizingly slow, so slow that you couldn't imagine taking a second pass through an edition after you found out how big it was really going to be. It would just take too much time. Fillers and house ads had to be used late in the process to make up for mismatches between expected size and actual size. Using this technique to paginate a classified section was like sitting in front of a chess computer and being told that the program knew everything about displaying positions and letting players move pieces, but nothing about playing as an opponent. The display might be pretty, but the results and the process are unacceptable. Manual electronic cut-and-paste pagination is no better than using the computer to display the chess board and the mouse to move the pieces by hand.

Is there a better model than going through an edition page by page until the section is complete? After all, this is the model that one uses when one is manually pasting up galleys of ads. This model fails if you want efficient, accurate, optimal, aesthetic production. It's too slow, it doesn't help resolve position request conflicts, it doesn't help with space-filling problems, and it certainly can't produce an aesthetic result without lots of manual intervention. The single pass model is inadequate.

SCS/ClassPag uses a new model that we call "progressive repagination." To make it work, two extremely difficult computational problems needed to be resolved. Both involve providing efficient solutions to the space-filling problem of placing ads on a page. One distinguishing factor of our solution is that we are able to quickly place 30 or more pages of classified display ads fully solid-set. SCS/ClassPag supports full or partial page square-offs for one or more pages at a time. The computational problem is like the chess problem. Just as a chess program searches many different positions in order to find the best move, SCS/ClassPag tries many different ad placements in order to pack pages solid. This "two-dimensional bin packing problem" has been a concern of computer and operations research scientists for over 30 years. The problem facing classified pagination for newspapers is nearly the same problem that many industries face in efficiently using sheet materials such as cardboard, fabric, sheet metal and even gold leaf. (At the price newsprint is going for these days, perhaps we should treat it as if it were gold. If you save a page, you save about $4.00 per thousand copies. The savings can be quite substantial when you have a metropolitan size circulation.)

Placing agate ads is similar to placing display ads. As the galley is positioned in the columns, it is important to be able to use the flexibility of agate ad reordering in order to get tight, consistent solutions for each column. SCS/ClassPag efficiently solves this "one-dimensional bin packing problem" by considering not just the first fit of the next few ads to be placed, but the best fit of the multiplicity of ads that could finish the current column and start the next. Few fillers are needed with SCS/ClassPag.

Clearly, if you have a formula for placing classified display ads and then flowing in the agate ads around them, you have the first step toward significant automation. Why should this process be elaborated into the progressive repagination model? Only after a section is paginated do you know how dense the entire publication will be. SCS/ClassPag provides a single-pass solution for an edition that gives a first good cut approximation at resolving all the issues. It does this at the rate of over 10,000 ads a second on a 200 MHz Pentium! With this

speed, it's possible to allow the entire page layout process to repeat even for the largest classified sections. Repagination offers opportunities for improvement. The trick is to use the information from each pass to guide the next one, as SCS/ClassPag does.

What kinds of problems will be solved in each successive pass? Well, the way the display ads are placed may create conflicts with the agate ads. For example, it may be better to place display ads over several pages even when they would actually fit on a single page, so that the classifications for both the display and classified ads are in sync. Doing classified headers and fillers well is challenging and seems best handled with multiple passes. Sometimes spreading out agate ads can reduce the number of headers needed. Placing fillers automatically requires keeping a library of fillers, while tracking their use, modifying their sizes, and leaving them in or taking them out as the progressive repagination process optimizes ad placement.

With SCS/ClassPag one can see pages on the screen as rendered by our PostScript RIP. Ads that are manually placed are treated as having fixed position requirements during the pagination and repagination processes.

Watching SCS/ClassPag in action is rewarding. It is astonishingly fast. Even its first pagination pass is likely to be publishable. Refined solutions are of excellent quality.

We like to take classified display ad sections produced manually or with other pagination systems and prove that we can do them in one or two pages less than the published editions. This is particularly gratifying when we're working with ad data from a large metro newspaper because we know we can save them a lot of money.

Finally, it's nice to see fully paginated pages coming out with colors automatically separated, perfectly straight columns of classified display ads precisely fitting in the column grids and all inter-ad agate spacing uniform and consistent. It's a tight, clean, neat solution produced without burdensome manual processes.

We like applying the best of computer science to the problems that newspapers have and providing solutions that make a real positive difference for them, their advertisers and their readers.

With appreciation:
Richard Koonce, a business programmer at the *Richmond (VA) Times*, a Media General company, developed a fast batch classified pagination program that was used by the paper in the mid-1990s. This program provided the inspiration for SCS to develop its own high-performance classified pagination technology. Many of Mr. Koonce's original concepts found their way into SCS/ClassPag.

The Times is a long-term Layout-8000 user. It provided Mr. Koonce's program in exchange for five years of support for the Media General Layout-8000 sites.

Richard J. Cichelli *is president of Software Consulting Services, LLC (SCS), a leading supplier of newspaper pre-press and post-press systems. SCS is best known for Layout-8000™, the industry standard advertising dummying system. Mr. Cichelli did early research into chess algorithms and was a founding member of the International Computer Chess Association and the Pascal Users Group. He was Delaware Chess Champion from 1965 to 1967. His background includes five years of graduate-level computer science teaching at Lehigh University in Bethlehem, PA and nearly ten years of newspaper computing research at the American Newspaper Publishers Association/Research Institute (ANPA/RI), now the Newspaper Association of America (NAA), where he was Research Manager for Computer Applications. Mr. Cichelli is also a former associate member of the American Institute of Architects.*

William Bader *is Director of Research and Development at SCS and principal author of SCS/ClassPag. Mr. Bader wrote a chess program while in high school that used progressive deepening and other advanced chess algorithms. At Lehigh University where he earned his Master's Degree in Electrical Engineering, Mr. Bader designed electronic circuits for chess algorithms. Mr. Bader is responsible for the early technical developments at SCS of Layout-8000, which began as Layout-80® under Mr. Cichelli's leadership at the ANPA/RI.*