Custom Legal Documents for the IBM AutoLoan Exchange

Douglas Lovell
IBM T. J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
dcl@watson.ibm.com

Introduction

Our group at the IBM T. J. Watson Research center develops transaction-oriented systems which reduce the processing barriers between businesses and their customers. We do this by developing advanced, computationally active user-interfaces, reengineering business processes, and building interfaces directly to existing data processing systems.

Early in 1996 we faced the task of providing document support for a pilot project we were undertaking with Chase Manhattan Automotive Finance Corporation (hereafter "Chase") a subsidiary of The Chase Manhattan Bank. The project would connect automobile dealers to the Chase automobile loan decision system via the internet. It would give dealers a loan decision in minutes for most credit applications. It would replace a system of data entry from faxes that produced loan decision times measured in hours to days.

Dealers know that customers generally do not return to the dealership a second time. If they say they'll be back, they won't. When a customer leaves without buying a car, odds are there will be no sale. Quick financing approval for customers with good credit would allow dealers to give approved financing and sell an automobile to a customer on the crucial first visit.

In addition to an electronic credit application, the system would allow dealers to submit an electronic loan contract. The electronic contract would enable the financial institution to almost immediately provide funds, thus reducing the "float" time between the dealer's payment for the automobile and the bank's payment to the dealer.

The first lenders on the system would anticipate growing market share due to increased service to the dealer. All lenders on the system would enjoy lower costs for approving loans.

In spite of the electronic nature of this reengineering effort, lenders, dealers, and purchasers would (for the forseeable future) need printed credit applications, contracts, and other documents related to the legal terms, conditions, disclosures, and obligations understood by all parties. In particular, purchasers must drive away from the dealership in their new automobiles with a copy of the agreements they have made to pay for them.

Our Decision to use TEX

We knew we needed to print, at a minimum, credit applications and financing contracts. Given this broad requirement we began exploring in greater detail what the document printing requirements would be, what the documents look like, and what our document objectives were. We generated and analyzed several technical approaches and selected one for further development.

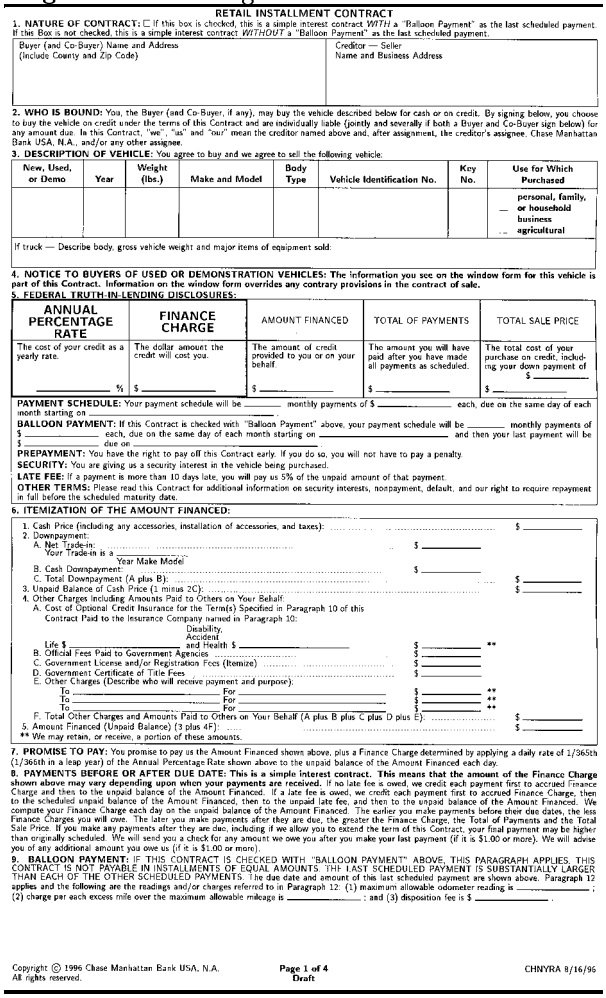
Characterization of the document problem Our first move was to gather as many of these documents as we could to learn what they were like. Chase was able to provide us with a sampling of their financing contracts from various states.

Figure 1 shows a representative example for New York.

We made the following observations:

1. The documents were printed front and back
2. They were standard width but lengths varied up to 28 inches, all longer than 14 inches
3. They were bound at the top with carbons (or carbonless), as white, yellow, and pink copies for the lender, purchaser, and dealer.
4. The copies were usually identical, however; in some cases the purchaser copy was missing the section used by the lender or dealer to assign the contract to another party
5. There was a large variety of type sizes and weights, but not a large variety of typefaces
6. There was a large variety of layout and graphics features– boxes, rules, brackets, columns, paragraph styles, bullets, enumerated lists at multiple levels, etc.
7. The documents were primarily printed with a single color. There was some use of shading using screens. There was rare use of a second color.

Figure 1: First Page of A Chase Retail Contract



RETAIL INSTALLMENT CONTRACT

1. NATURE OF CONTRACT: ☐ If this box is checked, this is a simple interest contract *WITH* a "Balloon Payment" as the last scheduled payment. If this Box is not checked, this is a simple interest contract *WITHOUT* a "Balloon Payment" as the last scheduled payment.

Buyer (and Co-Buyer) Name and Address (include County and Zip Code)	Creditor — Seller Name and Business Address

2. WHO IS BOUND: You, the Buyer (and Co-Buyer, if any), may buy the vehicle described below for cash or on credit. By signing below, you choose to buy the vehicle on credit under the terms of this Contract and are individually liable (jointly and severally if both a Buyer and Co-Buyer sign below) for any amount due. In this Contract, "we", "us" and "our" mean the creditor named above and, after assignment, the creditor's assignee, Chase Manhattan Bank USA, N.A., and/or any other assignee.

3. DESCRIPTION OF VEHICLE: You agree to buy and we agree to sell the following vehicle.

New, Used, or Demo	Year	Weight (lbs.)	Make and Model	Body Type	Vehicle Identification No.	Key No.	Use for Which Purchased
							_ personal, family, or household _ business _ agricultural

If truck — Describe body, gross vehicle weight and major items of equipment sold:

4. NOTICE TO BUYERS OF USED OR DEMONSTRATION VEHICLES: The information you see on the window form for this vehicle is part of this Contract. Information on the window form overrides any contrary provisions in the contract of sale.

5. FEDERAL TRUTH-IN-LENDING DISCLOSURES:

ANNUAL PERCENTAGE RATE	FINANCE CHARGE	AMOUNT FINANCED	TOTAL OF PAYMENTS	TOTAL SALE PRICE
The cost of your credit as a yearly rate.	The dollar amount the credit will cost you.	The amount of credit provided to you or on your behalf.	The amount you will have paid after you have made all payments as scheduled.	The total cost of your purchase on credit, including your down payment of $ _____
_____ %	$ _____	$ _____	$ _____	$ _____

PAYMENT SCHEDULE: Your payment schedule will be _____ monthly payments of $ _____ each, due on the same day of each month starting on _____

BALLOON PAYMENT: If this Contract is checked with "Balloon Payment" above, your payment schedule will be _____ monthly payments of $ _____ each, due on the same day of each month starting on _____ and then your last payment will be $ _____ due on _____

PREPAYMENT: You have the right to pay off this Contract early. If you do so, you will not have to pay a penalty.

SECURITY: You are giving us a security interest in the vehicle being purchased.

LATE FEE: If a payment is more than 10 days late, you will pay us 5% of the unpaid amount of that payment.

OTHER TERMS: Please read this Contract for additional information on security interests, nonpayment, default, and our right to require repayment in full before the scheduled maturity date.

6. ITEMIZATION OF THE AMOUNT FINANCED:

1. Cash Price (including any accessories, installation of accessories, and taxes)			$ _____
2. Downpayment:			
A. Net Trade-in:		$ _____	
Your Trade-in is a _____ Year Make Model			
B. Cash Downpayment		$ _____	
C. Total Downpayment (A plus B)			$ _____
3. Unpaid Balance of Cash Price (1 minus 2C)			$ _____
4. Other Charges Including Amounts Paid to Others on Your Behalf			
A. Cost of Optional Credit Insurance for the Term(s) Specified in Paragraph 10 of this Contract Paid to the Insurance Company named in Paragraph 10:			
Life $ _____ Disability, Accident and Health $ _____		$ _____ **	
B. Official Fees Paid to Government Agencies		$ _____	
C. Government License and/or Registration Fees (Itemize)		$ _____	
D. Government Certificate of Title Fees		$ _____	
E. Other Charges (Describe who will receive payment and purpose):			
To _____ For _____		$ _____ **	
To _____ For _____		$ _____ **	
To _____ For _____		$ _____ **	
F. Total Other Charges and Amounts Paid to Others on Your Behalf (A plus B plus C plus D plus E)			$ _____
5. Amount Financed (Unpaid Balance) (3 plus 4F)			$ _____

** We may retain, or receive, a portion of these amounts.

7. PROMISE TO PAY: You promise to pay us the Amount Financed shown above, plus a Finance Charge determined by applying a daily rate of 1/365th (1/366th in a leap year) of the Annual Percentage Rate shown above to the unpaid balance of the Amount Financed each day.

8. PAYMENTS BEFORE OR AFTER DUE DATE: This is a simple interest contract. This means that the amount of the Finance Charge shown above may vary depending upon when your payments are received. If no late fee is owed, we credit each payment first to accrued Finance Charge and then to the unpaid balance of the Amount Financed. If a late fee is owed, we credit each payment first to accrued Finance Charge, then to the scheduled unpaid balance of the Amount Financed, then to the unpaid late fee, and then to the unpaid balance of the Amount Financed. We compute your Finance Charge each day on the unpaid balance of the Amount Financed. The earlier you make payments before their due dates, the less Finance Charges you will owe. The later you make payments after they are due, the greater the Finance Charge, the Total of Payments and the Total Sale Price. If you make any payments after they are due, including if we allow you to extend the term of this Contract, your final payment may be higher than originally scheduled. We will send you a check for any amount we owe you after you make your last payment (if it is $1.00 or more). We will advise you of any additional amount you owe us (if it is $1.00 or more).

9. BALLOON PAYMENT: IF THIS CONTRACT IS CHECKED WITH "BALLOON PAYMENT" ABOVE, THIS PARAGRAPH APPLIES. THIS CONTRACT IS NOT PAYABLE IN INSTALLMENTS OF EQUAL AMOUNTS. THE LAST SCHEDULED PAYMENT IS SUBSTANTIALLY LARGER THAN EACH OF THE OTHER SCHEDULED PAYMENTS. The due date and amount of this last scheduled payment are shown above. Paragraph 12 applies and the following are the readings and/or charges referred to in Paragraph 12. (1) maximum allowable odometer reading is _____; (2) charge per each excess mile over the maximum allowable mileage is _____; and (3) disposition fee is $ _____.

Page 1 of 4
Draft

CHNYRA 8/16/96

8. There was some variation in in presentation from state to state.

9. There was some variation in content from state to state; but, much of the content was the same.

At the dealerships, the documents are stored in boxes as they came from the printing company. The financing agent at the dealer puts the preprinted document into an impact printer which blindly types data at given coordinates. A successful print results when the program doing the printing has coordinates which match the form, and when the form is aligned correctly in the printer.

We learned that the business of supplying the preprinted forms and the software coordinates to print the data on them is lucrative for the printing company and expensive for the banks. The business is somewhat painful for the dealer, who has to keep all of the forms stocked. It is also wasteful of resources. The dealers and suppliers discard all of the preprinted forms whenever a bank releases a revision, or when a state or federal law changes and all of the banks release revisions.

We thought we could improve on this by printing the documents from a laser printer. The dealer could stock blank legal paper and get document revisions from us at a fraction of what the printers charge to add a new preprinted form to their form printing program.

We had to begin by redefining features of the preprinted forms that we could not reproduce on a laser printer. One of these was the format. There is no ready supply of laser printer paper in 28 inch lengths, nor are there any acceptably priced printers which will handle this. One possibility was to use a laser printer that supported a ledger format. We hoped to avoid this due to the expense of installing such a specialized printer at each dealer site. We wanted to use a standard office printer.

Does a "single document law," stating that the agreement must be one document, mean that the document must be a single sheet of paper? It turns out that multiple page contracts are acceptable to the financial institutions, provided they are numbered in a "Page n of m" format. The format of the preprinted forms as long, single sheets was a convenience for shipping, handling, and feeding into an impact printer. It ensured that the entire document was present without introducing a need for stocking multiple sets of carbon-duplicated pages.

Another feature we could not readily produce on a laser printer was the second color. A second color was used on the Chase contracts for the states of California and Virginia, which have laws requiring that specific features of the contract be highlighted. Whether the highlighting must be achieved with a second color or whether it could be addressed by some other graphic mechanism– bold and big typeface, white space surrounding, bold frames surrounding –was an open question for some time. We investigated a color ink-jet printer in case the color features could not be redesigned; but, we made a decision not to let a few two-color contracts scuttle the whole printing project.

We learned that the multiple carbons do not have to be printed on differently colored sheets. It was sufficient to print multiple copies provided we identified each copy in the footer, e.g. "dealer copy," "customer copy."

Printing objectives Given the capability to reproduce the documents on a laser printer, we produced the following requirements:

1. Provide a means to model all of the layout and graphics features of the printed contracts.
2. Provide a means to print the documents with or without data in the spaces provided for data.
3. Provide a means to re-use portions of a document in other documents. We felt this would be important for efficiency and consistency given that much of the document content was repeated in multiple documents.
4. Provide a means to display portions of a document in the user interface.
5. The process for adding data to the document and printing must run without any special user actions, other than pressing a "print" button on the user interface.
6. Find a solution that can be realized quickly at reasonable cost.

Part of the variation in the content of the financing contracts was in the disclosures and notices which appear above the signatures on the documents. One of our goals was to transmit an electronic contract, including signature, from the dealer to the bank. This implied that we might need to display part or all of the contract in the user interface – at least the disclosures – before taking the purchaser's and dealer's signature.

We believed that we should consider extracting parts of the document content for display in the user interface.

As we would print each document specifically for each financing agreement. The following additional capabilities seemed reasonable:

1. Omit (or include) specific portions of the document as applicable to the agreement.
2. Reformat paragraphs and boxes to fit the data.

The first additional capability allows us to eliminate paragraphs pertaining to options, such as "optional credit life insurance" or "balloon payment options," which are not exercised for a particular agreement.

The second solves the classic problem we've all encountered when completing forms– blank spaces too small or grossly oversized for the data we need to write into them.

Technical alternatives Having worked with LaTeX in the past, I believed that TeX could carry the project where we needed to go. Before settling on TeX, we explored a set of other strategies. We separate these strategies into two classes– overlay and template.

Overlay strategies use a static image of the form with blank spaces for data. The static image is typically obtained by scanning a paper original, or by using a preprinted form. The overlay process prints the data over the static image using a map which matches data items with locations of the blank spaces.

Overlay strategies faithfully reproduce features of the printed document. When the image quality of a faxed document is acceptable, the image file can be acceptably small. The data maps may be produced fairly quickly without much skill.

Overlay strategies fail to meet the two additional capabilities we hoped to incorporate in our system. Given a static image of the form, there is no stretching of spaces too small for the data, no shrinking of spaces too large, and no capability to eliminate unneeded paragraphs and repaginate the document.

Overlay strategies do not readily lend themselves to extraction of content for the user interface, or to re-use of content in multiple documents. An overlay has no internal structure that may be taken apart and reassembled in new ways.

When an overlay document changes the entire document must be rescanned and the data map rebuilt with new data locations. There is also a trade-off with overlays between image quality and file size. High quality text requires very large image file sizes.

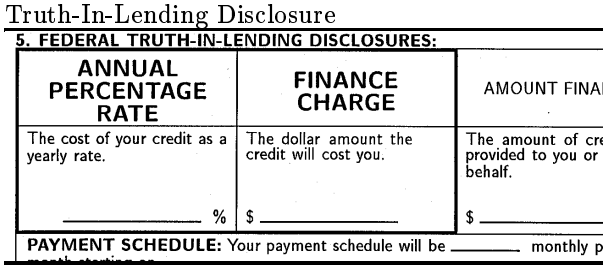
We found that the overlay strategy is acceptable for some documents which do not require high quality text, sharp lines, or reformatting. We use it to print credit applications, for example. We did not find the overlay strategy acceptable for our contract printing requirements. For those we turned to template strategies.

Template strategies represent the document as a series of layout and graphics controls with text. Place-holders in the document mark positions for the data values. A merge process that we call "data injection" replaces the data place-holders with actual data. The typesetting system reformats the document with the data in place.

A number of word processors and page composition systems have formatting and data injection capabilities; however, we eliminated these immediately because they were not designed as unattended background processes and use mostly proprietary data formats.

This left us with text markup languages which have typesetting engines on the PC platforms we run. We briefly considered SGML, but the high-level markup must still be translated into some lower-level typesetting commands, like TeX. The full generalization of SGML was more than our needs

Figure 2: Portion of the Federal Truth-In-Lending Disclosure



5. FEDERAL TRUTH-IN-LENDING DISCLOSURES:

ANNUAL PERCENTAGE RATE	FINANCE CHARGE	AMOUNT FINA
The cost of your credit as a yearly rate.	The dollar amount the credit will cost you.	The amount of cr provided to you or behalf.
%	$	$

PAYMENT SCHEDULE: Your payment schedule will be _____ monthly p

required. While we were concerned with high-level markup, we were more immediately concerned with using a typesetting engine that could reproduce all of the graphic features of the contracts. We also had LaTeX as an example of higher-level markup implemented using TeX.

After some exploration my initial inclination to try TeX seemed justified. Feeling reasonably assured, we set out to prove to ourselves that we could build our contract printing solution using TeX as the typesetting engine.

Proof of Concept

Given a decision to use TeX, we had to show that we could reproduce the contracts in TeX and that TeX could support our need to inject data into and extract content from them. In a project like this it is important to get something working quickly, then improve upon the working prototype. We decided on a two phase campaign to prove our template printing approach using TeX. First we would produce the contract document needed for the Chase pilot using TeX markup. Second, we would implement a small program to inject data into the document.

The first document The task of creating the first document was assigned to me. Prior to this project I had used LaTeX to typeset papers and letters. The contract documents we had to duplicate presented a new set of challenges. Knowing that the LaTeX standard formats were not appropriate for these documents, I began immediately using plain TeX.

I began with the basic page layout and a paragraph of text, then tackled the parts of the documents that had a large number of rules and alignments. We determined that these were the most difficult portions of the documents to reproduce and chose to work on these first to demonstrate that TeX could reproduce them.

The section which gave the most difficulty was the "Federal Truth-In-Lending Disclosures" (see fig. 2).

The challenging features were center-justified text, text centered vertically as well as horizontally, paragraphs within columns, and rules which changed weight going across the page. I had to fully understand the way TeX builds boxes in its various modes and how to control an `\halign`.

Learning TeX was a great deal of fun. The TeX layout model of boxes, glue, and springs was a delight to learn. Most confusion came from understanding the modes– the difference between vertical, horizontal, and restricted horizontal modes– and more, when the transitions occur between modes. Controlling vertical space on the transition from horizontal back to vertical mode was a puzzle.

It took two weeks to code the first page of the contract (fig. 1) given a study of the first few chapters of Knuth's TeXBook (Knuth, 1994), and using the task to direct further reading and experimentation. Most of the rest of the contract was text supplied by a fast typist. We were now well satisfied that we could faithfully reproduce the contract using TeX.

Data injection The second task was to develop a method to get our data values into the document. Our data comes to the document printing subsystem as a stream of characters which we parse into named "tables" and "lists."

A table contains a set of named "fields," each of which has a value. A list contains an array of such sets. We refer to a table field using the name of the table and the name of the field, e.g. "AutoContract.CashPrice." We refer to a list field by adding an index, e.g. "AutoOtherCharges[0].Amount"

We have a set of functions written in the C programming language which return values from the data stream given table, list, and field names.

We initially develped data macros to support table values only. We designed the first macro, "`\DataTarget`" with three arguments'

1. Table name – the name of the table,
2. Field name – the name of the field from which to take the value, and
3. Default width – the size of the blank to leave when there is no value.

This macro output an `\hrule` on the baseline using the given width. It completely ignored the first two arguments. This provided a reasonable default behavior which enabled us to process the documents without any data present and produce a usable blank document.

To get data into the document I wrote, in C, a simple text filter which accepted a document file and a data file. It output a copy of the document file

with data in place. The filter passed all characters until encountering the string, "`\DataTarget`." It then parsed the three arguments– table, field, and width. The filter used the first two arguments to find a data value. If there was a data value it output the value inside of a macro, "`\DataValue`." If there was no value, the filter output the width inside of a macro, "`\DataBlank`."

The `\DataValue` and `\DataBlank` macros could format their arguments in any way we chose. We chose to format the value of `\DataValue` in the `\tt` font. `\DataBlank` output an `\hrule` on the baseline with width given in the third parameter.

The preprocessing filter did its job very satisfactorily. We had proved TEX and our data injection process to be a viable, capable solution to our document printing needs. We had not yet tested reading a portion of a document or displaying a portion in the user interface; however, given that the TEX sources were all ASCII text files and given our experience with the preprocessor, we believed there would be no serious obstacles to implementing those functions.

Pilot

The user interface for gathering credit application data and the electronic submission and approval process was well under way before we began work on printing contract documents. We were able to integrate the Chase contract into the system before giving a preview of the system, with Chase, to the National Automobile Dealers' Association (NADA) convention in February 1996.

The then unnamed IBM AutoLoan Exchange system created a small sensation at NADA '96. Dealers and financial institutions were excited about what we had done and were anxious to take part. It was clear that we had made a significant advance with high impact and significant benefit in an area that had been neglected by technology providers. The industry response demonstrated considerable promise for our system.

The success was *not* primarily due to documents. Dealers were impressed that the system could give quick loan approvals and promised to close financing contracts and enable rapid delivery of payment for the cars they sold. It didn't hurt that we could produce a completely filled-in contract from a laser printer– one more thing they hadn't seen done before.

What the success meant for documents was that the document portion of the system had to grow fast. We had to scale it to multiple states and multiple products. Given a successful pilot project with Chase, we would need to produce documents for multiple financial institutions as well. We had to produce hundreds of documents. We needed to build a document creation team.

The first problem was finding people skilled with TEX. It appears that TEX is primarily used by professionals as a tool for doing their other work, or publishing their work. These are not people who are available to write automobile finance contracts.

Another niche for TEX is in the scientific and technical publishing business. We did identify a few people from this community who bid to do the work.

It was amusing to try to explain needed skills to representatives from personnel agencies. None had heard of "tech." Giving the spelling and saying "tex" as in "Texas" helped some; but, searches in their databases for the keyword "tex" yielded few potential candidates. Queries for markup skills– SGML or GML –met with little better success.

Fortunately, one vendor was willing to dig deeper to find the skills we needed. He began with the listings in the back of *TUGboat*, and with a reference supplied from an email inquiry to TEX Users Group. He did the phone calling and followed leads to identify people who could code the documents and experts who could write the macros.

The backgrounds of the people who worked on our documents is indicative of the problem of finding skilled document production help for TEX. They were:

1. a Physicist turned TEX expert,
2. a Crystolographer turned IS department manager,
3. a former secretary to a large university mathematics department, and
4. me, a computer programmer.

In addition, while we preferred to find people who could come to work with us on-site, it became clear that we would have to settle for a distributed, electronic, virtual workplace. We coordinated by telephone and shared files by ftp transfer. I had to act as administrator and librarian to keep synchronized the hundreds of files we generated.

Many of the TEX experts who bid for the macro work focused on the data injection portion of the problem, perhaps because we had explained this most clearly. In fact, this was the problem we felt most confidently we had solved.

The part of the problem less clearly expressed, but of greatest concern to us was the problem of efficiently producing the full, large variation in page

layouts and typographic features of automobile finance contracts. We felt that a flexible, extensible, high-level markup was needed to meet our goals of

1. having administrative level document production staff quickly produce fifty to seventy documents for a new financial institution,
2. faithfully reproducing the style and content of those documents, and
3. quickly tooling for a new style by reusing as much as possible the work done to produce previous styles for previous financial institutions.

The TEX expert we selected understood this problem and directly addressed it with a syntax for extensible markup and a framework for TEX macro development and reuse (Ogawa, 1994; Baxter, 1994).

Rollout

At this time we have complete contracts for most of the fifty states for Chase, and a few dozen states' documents for three other financial institutions. Chase has begun using the documents for actual, legally binding financing contracts at dealerships using the ALX system.

The help we got from an expert TEX "insider" was crucial to our success with these documents. He was able to code some of the features which were giving us a hard time, such as page numbering and footers, section cross-references, and two-column layout. More importantly, he was able to capture features as separate elements which could be applied repeatedly as needed in the many documents we produced. He was also able to develop content-level commands we used to code the structure of the document. He configured the appearance of these for each document style. Where the appearance varied within a document, he provided attributes we could code to guide the presentation.

Consider the case with sections. The presentation of the section number and title varies from document to document. Figure 3 shows examples.

Those of us coding the documents wrote a `\section` command with a `title` and `label` attribute. Where there was a box around the section we could specify `framing`. The document style took care of the font change of the title position of the title, and position of the text relative to the section number.

All of this meant that the document coders could focus on producing the content of the documents using fairly high-level markup. The details of presentation, the hard and messy typesetting considerations, were taken care of by lower-level

Figure 3: Section style variations

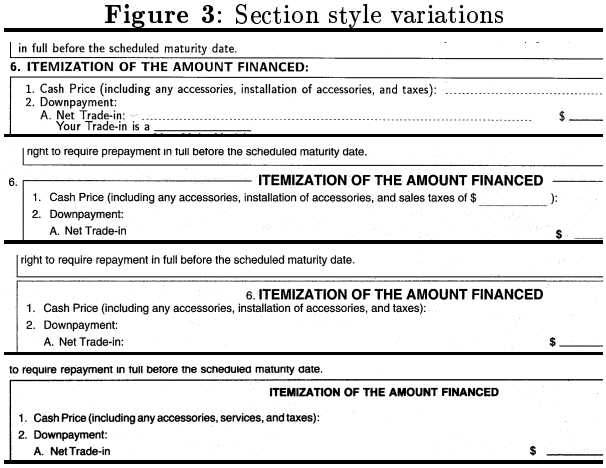
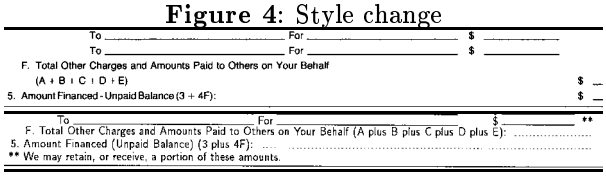


Figure 4: Style change



presentation code. This allowed us to specify a large lead time for first-of-a-kind documents and realize a short turn-around time for n'th-of-a-kind documents.

The only drawback was that I no longer well understood the formatting code. We became reliant on the consultant. When we wanted a change quickly we had to appeal to him to give our work priority, or resort to low-level commands to implement the change ourselves, locally in the document.

One example occurred when a client added a footnote within an itemization section (see fig. 4).

The consultant was not able to implement this immediately; so, we had to revert the `\item` markup to an `\halign`.

We believe that, as we do more and more documents, fewer and fewer of these new typesetting needs will arise. As we add each new feature to our toolkit we will be able to produce new document styles with less new coding effort. We have had some encouragement in this belief in our experience with adding new financial institutions. The second and third financial institution required considerably less work due to the large amount of code reused from the Chase pilot.

Conclusion

The IBM AutoLoan Exchange has delivered on much of its promise. We now deliver rapid loan decisions and contract funding for dealers subscribed

to the system through Chase. with many other finance institutions and dealers to follow. The application has defined a new direction for the business of automobile financing unforseen by existing vendors in the field.

The document capabilities of the system form a small, but essential part of the overall success of the IBM AutoLoan Exchange in defining a new paradigm for the business of making and closing automobile financing. Out go truck-loads of preprinted, carbonless forms in quadruplicate. In come laser printers and reams of fresh, blank, legal-sized laser printer paper.

Dealers print only the documents they need, when they need them, complete with the data values for each particular financing deal. When a form changes there are no bundles of preprinted, obsolete forms to discard.

Our world is a networked, distributed world of electronic commerce where complex legal documents are delivered from a server and customized by a client at the point of delivery. We have shown that TEX can provide the capability needed to typeset these documents transparently, without recourse to a page-layout or word-processing system.

The TEX typesetting system was vital to the success of the printing subsystem of the IBM AutoLoan Exchange. TEX can play a vital role wherever custom legal documents are needed in the world of the Net.

Acknowledgements

Thank you to Stephen Boies and Lauretta Jones, visionary leaders of the IBM AutoLoan Exchange project.

Thank you to Arthur Ogawa and his many helpers in the TEX community who provided the advanced TEXnology to make the IBM AutoLoan Exchange documents really work.

Thank you to Alan Bednowitz and Catherine McGinnis for their detailed and dedicated work to produce nearly one hundred contracts to date for the IBM AutoLoan Exchange.

References

William Erik Baxter An object-oriented programming system in TEX *TUGboat*, 15(3):331–338, September 1994

Knuth, Donald E. *The TEXbook*. Addison Wesley, Reading, Massachusetts, 1994.

Ogawa, Arthur. "Object-oriented programming, descriptive markup, and TEX". *TUGboat* **15**(3), 325–330, 1994.