リース業パッケージ開発における HYPERPRODUCE II の適用

Application of HYPERPRODUCE II in Development of Lease Business Package

北 川 克 也

要 約 近年の PC サーバの急速な発展に伴い，オープンな環境でのシステム構築が強く望まれている．

リース業パッケージにおいても同様であり，Windows 環境下で稼働する LeaseCreation を開発した．このような中で開発環境として何を使用していくかということは，生産性はもとより，今までオフィスサーバの開発に従事してきた SE（System Engineer）やプログラマをどのように方向づけるかという意味で大変大きな課題である．

当初はもっとも一般的で SE やプログラマの人口も多いと思われる Visual Basic を採用するつもりであったが，最終的には HYPERPRODUCE II を採用した結果，生産性の点で他のプロジェクトと比較して作成コストが約半分という効果が得られた．

今後は，客先展開におけるカスタマイズや保守の生産性を如何に効率よく行なうかが課題となる．

本稿では，このプロジェクトにおいて Visual Basic ではなく HYPERPRODUCE II を選択するに至った経緯と実際に使った結果について述べる．

Abstract According to the recent rapid growth of PC server, it is highly desired that many business solution systems be inplemented on a PC-based open platform.

The same needs have arisen in the lease business solution market, and we have developed the LeaseCreation package on Windows platform. In the circumstances, the devlopment environment is an important problem in the development on the PC-based platform for not only the productivity but also skill orientation both of system engineers and programmers attended in the devlopment of proprietary platform.

First, our project inteded to use Microsoft Visual Basic that is the most familiar programming language, but finally determined to employ HYPERPRODUCE II of Mitsubishi Electric co., which brought us the significant productivity twice as much as the other development project. Now, we concerned about the productivity of customizing and maintaining of developed software packages to match user requirements.

This paper describes the decision process of choosing HYPERPRODUCE II, not Visual Basic and the outcome of its employment.

1. はじめに

LeaseCreation は，リース業向けのパッケージであり，オフィスサーバで稼働しているリース業パッケージ LEASTATION を，Windows NT をプラットフォームとするクライアント・サーバ型で提供して欲しいという要望に応えるべく新たに開発した．

オフィス・サーバでの開発から，パッケージと開発者という二つの資産を継承することが，この開発プロジェクトの課題であった．パッケージは，多様な利用方法に対応してカスタマイズできなければならないため，通常の開発に比べ要件定義工程の比

率が高く，見積もりも難しい．このことを補うため，実装工程をいかに効率よく短期間で行なうかが課題となる．今回の開発でも，要件定義に手間取り，厳しい開発スケジュールとなった．そして，オフィスサーバの開発に従事してきた SE やプログラマをどのように方向づけるかがもう一つの課題である．

いずれの課題に対しても，開発環境の選定が鍵となる．Windows での開発環境は，さまざまな製品があり，それぞれに特徴がある．当初は最も一般的で，SE やプログラマーの人口も多いと思われる Visual Basic を採用するつもりであったが，検討を重ねた結果，三菱電機社製の HYPERPRODUCE II を採用した．

本稿では，このプロジェクトにおいて Visual　Basic ではなく，HYPERPRODUCE II 選択するに至った経緯と，実際に使った結果について述べる．まず，LeaseCreation 開発の概要を機能と開発プロセスの面から紹介する（2 章）．次に，開発環境に必要な要件と HYPERPRODUCE II の特徴（3 章）を踏まえて，選定の経緯（4 章）を述べる．最後に，開発の結果（5 章）と，そこから得た経験について考察する（6 章）．

2.　リース業パッケージの開発

2.1　LeaseCreation とは

LeaseCreation は，リース業トータルシステムであり，営業支援から業務管理，経営支援までの機能を幅広く提供する．LeaseCreation の主な機能を表 1 に示す．

リース業を営む会社は，LeaseCreation を使用することにより，顧客から要求されたリース契約の見積を行なったり，契約の締結や契約に関わる物件の管理，リース料の請求，固定資産税や各保険料の支払等の業務を効率よく行くことができる．

なお，LeaseCreation はユーザベースで 100 社ほどを見込んでいるが，リース業においては契約や資産管理方法など各社運用形態が異なる部分があり，そのような部分に関してはその会社独自の機能としてカスタマイズを適用することで対応する．

2.2　開発プロセス

LeaseCreation の開発は，以下のような手順で作業を進めていった．

1）　要件定義

新世代のリース業アプリケーションとして必要な機能を，実際に現場で作業を行なっている SE や営業の意見を取り込み定義した．なお，今回開発するのはパッケージなので，社内のリース業務担当者を中心に要件を聞き出した．

2）　論理設計

要件定義に従って，設計担当者は，開発メンバの中の業務担当者からインタビューを行い，業務要件の詳細を設計した．

次に設計担当者が作成した論理設計書を業務担当者がレビューし，不足部分や勘違いを補正し論理設計書として仕上げた．ただし，ここではあくまで業務要件を明確にすることを目的にしているため，実際の画面イメージやデータの入力方法等については明確にしてはいない．

また，必要な項目の洗い出しとグループ化を行ない，ER 図を作成しデータベースの関連を設計した．

3）　物理設計

表 1　LeaseCreation の主な機能

項番	機能	概要
A	見積・契約・検収	見積や契約の入力，契約の締結，検収等の処理
B	満了	満了する契約の通知書の発行や再リース等の処理
C	処分	満了した契約の物件の処分の処理
D	中途解約	契約の中途解約の処理
E	契約・資産変更	契約の分割や変更，資産の変更の処理
F	地位継承	契約の継承の処理
H	異動取消	契約の変更等の異動の取消等の処理
I	請求	請求書の作成や振替処理，回収変更等の処理
J	入金	入金の消し込みや取消の処理
K	支払	支払いの承認や変更，実行の処理
L	会計	月次計上や前渡金振り替え明細表の作成処理
M	保険	動産保険，賠責保険，信用保険の付保等の処理
N	固定資産税申告	固定資産税申告のデータ作成や明細書の作成処理
O	減価償却	帳簿価額増減表の作成処理
P	決算	償却実施額計算，期末繰越し指示等の処理
Q	経営企画	予定データ作成
R	統計	リース月次・年次統計調査表の作成等の処理
S	照会	契約等の照会処理
T	リース会計基準	リース取り引きのご案内データ作成等の処理
U	取り引き管理	検収管理表や満了管理表，処分管理表の作成処理
V	運用管理	各情報のメンテナンス処理
W	システム管理	コントロール情報のメンテナンス処理

論理設計書に基づき画面イメージを作成し，各項目の表示イメージや表示および入力データの型や桁について詳細を詰めた．また，同時に ER 図上に具体的な各項目の型や桁について定義を行なった．この時に各テーブルの項目の依存性やテーブル間の参照整合性などについても決定した．

画面及び帳票の設計は，Visual Basic や HYPERPRODUCE II の帳票設計機能を使って設計書を作成し，実装時にはこれを流用して工数を減らせるよう工夫した．

4）実　　装

物理設計に従って実装を行なった．この際，クライアントに関しては，物理設計時点で画面フォームはできていたので，後はイベント処理や画面の遷移等についてコードを記述していけば良かった．

サーバに関しても物理設計書に記述してある処理ロジックに沿って，ビジネスロジックを日本語スクリプトで記述していけば良い，また，ファイル設計と帳票設計についても物理設計の段階で終了しているので，それをそのまま使用することができた．

5）統合・システムテスト

統合・システムテストは，要件定義で定義したビジネス要件を満たしているか，また，見た目や操作性に違和感はないかなどを確認した．

3. HYPERPRODUCE II の特徴

ここでは，HYPERPRODUCE II の狙いと特徴・機能について述べる．

3.1 事務処理システムでの開発ツールの要件と HYPERPRODUCE II の狙い

従来，汎用機やオフィスサーバ市場でアプリケーションを開発するための言語として，技術計算や事務処理システム向けに FORTRAN や COBOL 等の第 3 世代言語が使われてきた．

事務処理システム向けには，さらに簡略化した記述でプログラムを開発でき，COBOL より生産性が高いと言われている RPG（Report Program Generator）系の第 4 世代簡易言語もオフィスサーバを中心に使われてきた．当社の多くのシステム構築で採用している三菱電機社製開発言語プログレス II（P II）という言語も RPG 系に属している．

一方オープン市場では，C 言語/Java 言語を中心に，Basic 系の言語も多く使われており，新規プログラマを中心に要員が急増してきている．

しかしオープン系の多くの言語は，オペレーティングシステムやデータベース，通信インタフェースなどのプラットフォームに依存した仕様を多分に含んでいるため，プラットフォームの機能を引き出すためには柔軟な記述ができるが，COBOL や RPG のようなあまりプラットフォームに依存しない，事務処理システムを記述する言語とは異なっている．したがって事務処理システム構築を得意とする SE がビジネスロジック設計をプログラム化する作業に，オープン系言語を採用した場合プラットフォームに依存したプログラミングテクニックを習得する必要があり，SE にとって大きなスキルギャップが生じてしまう．

このような状況で事務処理システムを構築する SE 向けに，Windows 環境で RPG 系の言語仕様を取り込んだ開発環境である HYPERPRODUCE II は，プラットフォームに依存する部分を極力隠蔽し，ビジネスロジックの設計・プログラミングに専念できる環境を用意している．HYPERPRODUCE II は，COBOL と同様に既存の事務処理システム構築ノウハウやビジネスロジックのプログラム部品を流用できる．また言語仕様も事務処理システム構築 SE にとって，理解しやすい処理サイクルなどの機能を内蔵しており，習得が容易と考えられる．ただし，残念なことに現バージョンの HYPERPRODUCE II は，クライアントプログラムを Visual Basic で作成する必要があるため，本来の目的を実現しきれてはいない．今後，クライアント開発についてもサーバ開発環境と同様のロジック設計を可能とする環境を望みたい．

3.2 特　　徴

HYPERPRODUCE II は，Windows 環境下で稼働する 3 階層のアプリケーションプログラムを開発するための環境である．位置づけは下流 CASE であり，システム設計から保守まで幅広くアプリケーション構築を支援する．ファイル設計や帳票設計，そしてクライアント（プレゼンテーション層）とサーバ（ビジネスロジック層）間のインタフェースを規定するトランザクション設計などの部品設計機能があり，さらに，クライアントとサーバ双方のプログラムロジックをウィザード形式で自動生成するプログラム構築支援機能，単体テストやシステムテストが容易に行なえるテスト支援機能がある．また，アプリケーションの開発や保守における作業の効率を上げるため各種のドキュメントを生成する機能等がある．

HYPERPRODUCE II でアプリケーションを開発する際には，まずトランザクショ

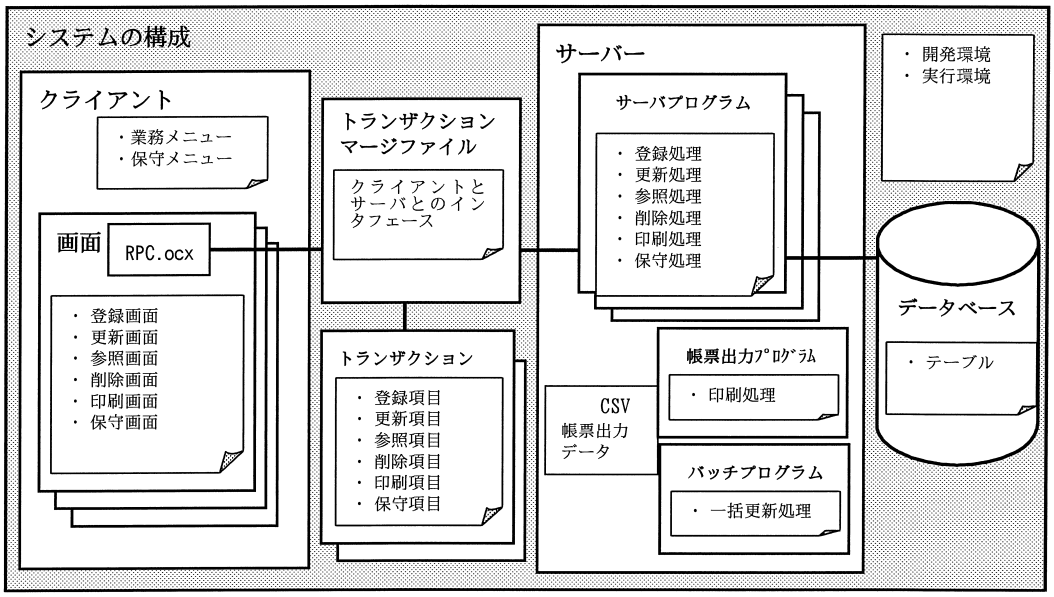

図 1　HYPERPRODUCE II が生成するシステム構成

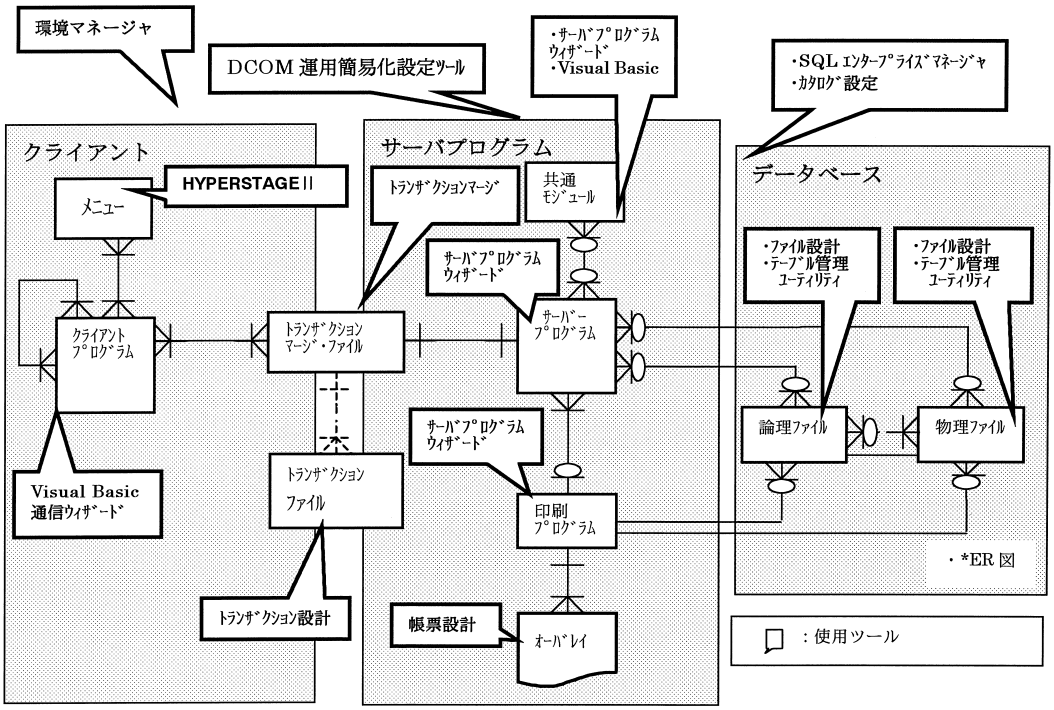

図 2　HYPERPRODUCE II の機能とモジュールの関係

ンやファイルなどの部品を設計し，次にそれらを利用するクライアントプログラムやサーバプログラムを作成していく．プログラム作成時にはデバッグ機能を用いてクライアントプログラムおよびサーバプログラムそれぞれを個々に単体テストすることが

できる．そして，それぞれの単体テストが終了した時点で，実際にクライアントプログラムとサーバプログラムを結合し，クライアントプログラムとサーバプログラムが正しく連携して動作するかをテストする．

このようにして個々の機能を持ったプログラムを作成し，でき上がったプログラムを集大成してアプリケーションプログラムが完成する．

3.3 機　　能

HYPERPRODUCE II が生成するシステムの構成を図 1 に，ツールとモジュールの構成を図 2 に示す．

また，HYPERPRODUCE II の各ツールの機能概要は表 2 の通りである．

表 2　HYPERPRODUCE II 機能一覧

	名称	機能
部品設計機能	環境マネージャ	システム環境の管理を行なう
	リポジトリマネージャ	システム環境内のプログラム，ファイル，帳票などの部品，部品内の定義項目の管理を行なう
	ファイル設計	プログラムで使用する入出力ファイルの設計を行なう
	トランザクション設計	データ・トランザクションの設計を行なう
	トランザクションマージ	トランザクション設計で設計した複数のトランザクション情報の結合を行なう
サーバ プログラム開発	サーバプログラム・ウィザード	サーバ上で動作する業務プログラムに関してソースプログラムの作成，コンパイル等を行なう
ユーティリティ	ドキュメント・ファイル出力	各部品設計を元に各種ドキュメントの出力を行なう
	カタログ設定ユーティリティ	DB 別名（論理的なデータベース名）と ODBC データソースの関連付けを行なう
	テーブル管理ユーティリティ	ファイル設計で生成されたテーブルをデータベースでの作成，削除を行なう
	帳票ジョブウィザード	印刷処理のバッチプログラムの生成を行なう
	ｻｰﾊﾞﾌﾟﾛｸﾞﾗﾑ支援ﾗｲﾌﾞﾗﾘ	サーバプログラムで使用する支援ライブラリ
	外部アサインツール	サーバプログラムのディフォルト属性の変更を行なう
クライアント プログラム開発	通信ウィザード	クライアントプログラムに通信処理の追加を行なう
コントロール	HPCRPC32.OCX	クライアントプログラムがサーバプログラムと通信を行なうためのコントロール
	HPCFLD32.OCX	表示編集機能を持った入出力領域用のコントロール
	HPCSHT32.OCX	表示編集機能を持った明細入出力用のコントロール
テスト・デバッグ	テスト項目作成	シミュレーションデータファイルの作成支援を行なう
	デバッガ	クライアントアプリケーション，または，サーバアプリケーション個々のデバッグを行なう
	トレーサ	クライアントアプリケーションとサーバアプリケーション間で送受信されるメッセージの採取を行なう
メニュー機能	HYPERSTAGE II	HYPERPRODUCE II とは別製品であるがメニュー機能やサーバのジョブ制御，ログ機能などを提供している

4. HYPERPRODUCE II の選定理由

開発環境の選定作業は，LeaseCreation の要件定義と並行して行なった．WindowsNT の環境で使用可能な多数ある開発環境の中から選定する際の条件として，開発メンバーの習熟度があげられる．少なくとも開発メンバーにとってまったく新しい環境を使用するということはありえないことであった．今回の開発メンバーは，業務に強く P II での開発に熟練しているオフィスサーバ系の SE やプログラマと，業務にはさほど強くないが Visual　Basic を使った WindowsNT での開発の経験があるオープン系の SE やプログラマというメンバーでチームを編成した．なお，この構成は，LeaseCreation のみならず今後の他のアプリケーション開発でも同様であると想定された．そのため，オフィスサーバ系の SE やプログラマに馴染みやすい HYPERPRODUCE II とオープン系の SE やプログラマが慣れている Visual Basic の二つの開発環境を候補とした．当初論理設計が始まる頃までは，細かく処理を記述でき Windows 環境での標準開発環境ということで Visual Basic の採用をほぼ決めていた．ところが，今回のプロジェクトを遂行する上で重要なポイントになったのが，開発期間の短縮や保守，カスタマイズの容易性であった．そのため，使用する開発環境を見直してみることになり，HYPERPRODUCE II が今回の開発で使用に耐えうるのかを検討した．

4.1 開発環境としての考察

HYPERPRODUC II を開発環境として検討するにあたって以下の点について考察を行なった．

1） 初期開発の生産性

他社の HYPERPRODUCE II での開発経験によると工数は P II の 1.2 倍程度かかるということであるが，クライアントを Visual Basic で作成する必要があるため，LeaseCreation の開発では 1.5 倍かかるという見積りを採用した[*1]．ちなみに，当社では COBOL の工数は P II の 1.4 倍程度かかるという実績がある．

また，設計情報は，オフィスサーバの表記法に準拠しているため，独自の環境になっているため，オープン系の SE やプログラマには馴染まない部分がある．しかし，実際にプログラムを作成する SE の半数近くは以前 LEASTATION を作成した経験を持っており，P II の発想でコーディングをすることができるということは，短期間で作成できるとことにつながる．

また，クライアントアプリケーションとサーバアプリケーションを独立して開発，デバッグが可能である．これにより，クライアントアプリケーションとサーバアプリケーションの開発を非同期に行なえるため，開発リソースの割り当ての自由度が高まる．

なお，日本語スクリプトで実現できない機能等は，Visual Basic で作成した ActiveX.DLL との連係機能により実現することが可能である．

2） カスタマイズの容易性

長年のオフィスサーバのアプリケーション開発に関与してきた P II の流れを汲むため現時点では，システム環境の管理単位がサブシステム単位になっているなど機能の不足はあるが，P II で培われたノウハウが活かされており，自由度の高い Visual Basic に比べるとプログラムを均質に作ることができる．これは，保

守やカスタマイズを行なう上で重要なことである．Visual Basic は自由度が高いため，開発者の好みによりいかようにでも作り込むことができる．パッケージを作成するには，成果物の均質性を保つために，ある程度制約がある方がよい．つまり，誰が作成しても同じような結果が得られる方が，保守や機能要求に対するカスタマイズ等をスムースに行なうためによいと判断した．

また，トランザクションごとにビジネスロジックを対応させるため，プログラムの局所性も保たれている．ただし，再利用性が弱いため，同じようなコードを何度も記述するケースもあるという欠点がある[*2]

3) メンテナンスの低負荷

他社の開発ツールと比較して，国産の開発ツールということもあり技術サポートについて実績がある．

また，HYPERPRODUCE II の独自環境でプラットフォーム等を隠蔽しているため，プラットフォーム等がバージョンアップした場合も，HYPERPRODUCE II が対応するだけであり，アプリケーションを変更する必要は無い．

4.2 工数についての考察

工数の比較をするために，開発環境の比較レポートを援用できる書籍販売の簡易版をプロトタイプとして，HYPERPRODUCE II と Visual Basic 6.0 で作成した．

開発作業は，それぞれ Visual Basic でのプログラム作成に慣れている SE と，P II でのプログラム作成に慣れている SE が行なったが，後者に関しては，HYPERPRODUCE II のスタディに掛けた時間は省いてあり実作業に掛かった時間のみ記録している．

実際にかかった工数は，表 3 の通りである．

表 3　書籍販売の簡易版の作成工数比較表

作業	工数			備考
	Visual Basic	HYPERPRODUCE II		
		1 次	2 次	
設計	5.0H	10.0H	6.0H	
部品作成	−			
ｸﾗｲｱﾝﾄ作成	10.0H	8.0H	2.0H	
サーバ作成	26.0H	4.0H	12.0H	1 次は断念
テスト	4.0H		4.0H	
合計	45.0H	22.0H	24.0H	

HYPERPRODUCE II で 1 次作成と 2 次作成があるのは，担当者が HYPERPRODUCE II に不慣れだったため，ファイルやトランザクション等の部品設計が十分ではなくプログラム作成を途中で断念したためである．そこで，三菱電機殿のサポートを受けながら再度作成したのが 2 次作成の部分である．

この結果から工数としては，開発初期段階では Visual Basic でも HYPERPRODUCE II でも大差が無いといえる[*3]．ただし，作成するモジュールが増えていくにしたがって，HYPERPRODUCE II での開発工数は Visual Basic に比べて半分程度までになる可能性がある．つまり 1 次作成での経験不足からくる余分な工数がなくなり，

2 次作成相当の工数のみになると思われる．それに加えて，今回は Visual Basic に対するリスクを見ていないが，実際に業務系の SE やプログラマの中には Visual Basic の未経験者も多い．

経験不足からくる余分な工数を除いたとき，HYPERPRODUCE II 対 Visual Basic の工数は，次のようになる．

HYPERPRODUCE II：Visual Basic＝24 ： 45
＝ 1 ： 1.875（少ないほうがよい）

5. LeaseCreation の開発

4 章で述べたように成果物の均質性や実際に作業する要員のスキルと教育期間の兼ね合いから，論理設計の中盤頃には HYPERPRODUCE II で開発を行なうことを選択した．そして，経験不足を補うために物理設計および実装の担当者に対して HYPERPRODUCE II の教育を行なった．また 4 月上旬は，各協力会社の担当者の HYPERPRODUCE II への不慣れのためのリスク期間と見ていた．

当初の予定では，3 月末までに物理設計を含めすべての設計作業を終了し，4 月から一斉に実装作業を開始する予定であったが，物理設計のレビューで不整合が見つかり，設計が完了するのは 4 月中旬以降にずれ込むことになった．この結果，不慣れを解消する期間がなくなってしまうことになった．そこで，各社のリーダと相談して，サンプルコードを増やし極力不慣れのための作業遅延をなくすよう努めた結果，予定通りに実装を終了することができた．ピーク時には，30 人以上のプログラマプログラム開発を行なった．

結局，LeaseCreation の実装成果物の総ステップ数は 308.2 KLOS であり，掛かった総工数は 343.1 人月であった．また，ファンクションポイント（FP）は 2457 ポイントである．

これらより下記の値を求めることができる．

1 FP 当りのステップ数 ：308.2 KLOS÷2457 ＝125.4 STEP/FP
1 人月当りのステップ数：308.2 KLOS÷343.1＝898.3 STEP/人月
1 人月当りの FP 値 ：2457÷343.1 人月 ＝7.16 FP/人月

ここで，比較のために他の言語を使った類似のプロジェクト X におけるこれらの値を引用すると以下の通りである．

1 FP 当りのステップ数 ： ＝128 STEP/FP
1 人月当りのステップ数： ＝483 STEP/人月
1 人月当りの FP 値 ： ＝3.77 FP/人月

これらを比較すると 1 FP 当りのステップ数は LeaseCreation の 125.4 に対してプロジェクト X は 128 であるので，HYPERPRODUCE II とプロジェクト X の開発環境の 1 FP 当りに必要なステップ数はほぼ同等である．

また，1 人月当りの FP は，LeaseCreation の 7.16 FP に対してプロジェクト X は 3.77 FP であり，1 人月当りのステップ数は LeaseCreation の 898.3 ステップに対してプロジェクト X は 239 ステップである．つまり，一つの機能を実現するために必要なステップ数は同等だが，作成に掛かる工数はほぼ半分であることがわかる．

このことより，HYPERPRODUCE II の生産性の高さを認識することができる．

6. 課題と今後の展開

厳しいスケジュールにも関わらず，何とか納期を守ることができた．今後は，実際に客先に展開していくことになるが，カスタマイズや保守の生産性を評価し，効率よくしていくことが課題である．そのためには，現時点で HYPERPRODUCE II に対して下記のような機能強化を望みたい．

1） ツール等をまとめて，より統合された開発環境にする．
2） ジェネレーションするソースコードをより効率化させ性能を向上させる．
3） 部品化の再利用性をより強化する．
4） ドキュメント出力機能をより強化する．
5） WEB 対応によりクライアントの生成機能を強化する．

また，今回の経験や成果物を今後の他のプロジェクトにも適用できるよう LUCINA for HYPRPRODUCE II として集大成する予定である．

7. お わ り に

初めて使う開発環境ということで，はたして HYPERPRODUCE II を開発環境として選択して良かったのだろうかという不安が何度も胸を過ぎったものである．

しかし，いくつかの課題はあるにせよ，われわれは HYPERPRODUCE II を使用して，大規模なアプリケーションである LeaseCreation を完成させることができた．

ただし，このことは，一概に開発環境を何にしたかということだけではなく，夜遅くまで残ってコードを記述したり，暑い中土日に出勤してデバッグを行なってくれたプロジェクトのメンバーおよび各協力会社の方々のご尽力によるものである．各人に深く感謝の意を表したい．

＊ 1　実際には，クライアントの機能が重かったため，クライアントの開発は P II 比 2 倍近くにはなっていた．今後クライアント開発環境が強化され生産性が向上することが望まれる．
＊ 2　ソースコードのコピー/ペーストは可能であるがペーストした時点で別のコードとして扱われる．不具合等があった場合，コピーした個数だけ修正作業が必要になる．
＊ 3　ここでは，1 次作成の分を HYPERPRODUCE II を初めて使用するリスク工数として見ている．

執筆者紹介　北　川　克　也（Katsuya Kitagawa）
1982 年佐賀大学理工学部物理学科卒業．同年日本ユニバック（現日本ユニシス）入社．パソコンの OS および Basic 言語のメーカリエゾン，端末エミュレータ等の開発に従事し，1999 年よりリースパッケージの IT 系リーダを担当，現在に至る．