



ICカード発行システム事例紹介

東芝ソシオエンジニアリング(株)
ソフトウェア統括部

SpecCへの取組目的

ハード、ソフトを一体としたシステム物件のJOBが多い。
上流から一貫した設計手法が欲しい。

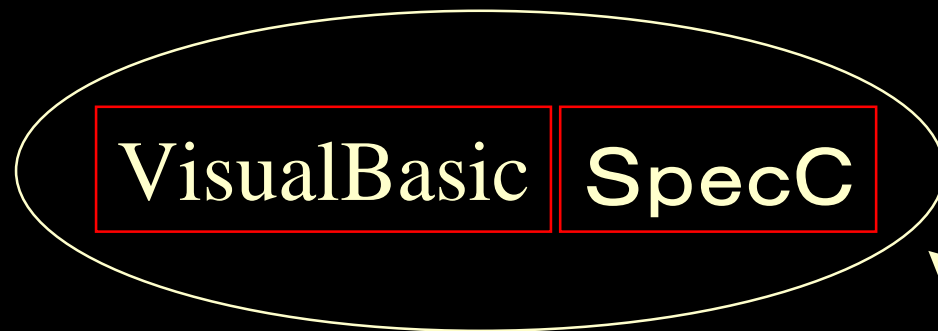
SpecCに注目

理解と習得の為、既存システムの設計で記述実験

ICカード発行システムへ適用
イベントドリブン型ソフトの記述法の工夫

SpecCの適用事例

GUI



武蔵工業大学

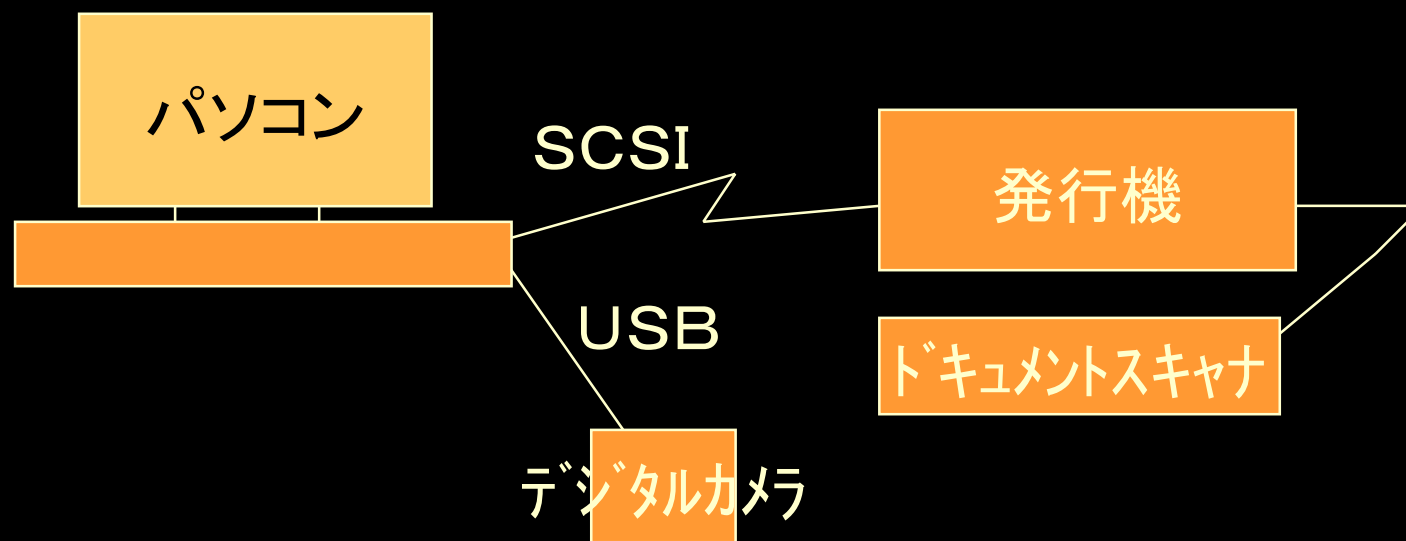
データベース・外部装置



東芝ソシオエンジニアリング(株)

ICカード発行システムとは

従業員証の様なカード発行を行うシステム。
(IC、磁気エンコード、バーコード印字も可能)



ハードウェア構成

ICカード発行システムとは

新規作成
開く
印刷

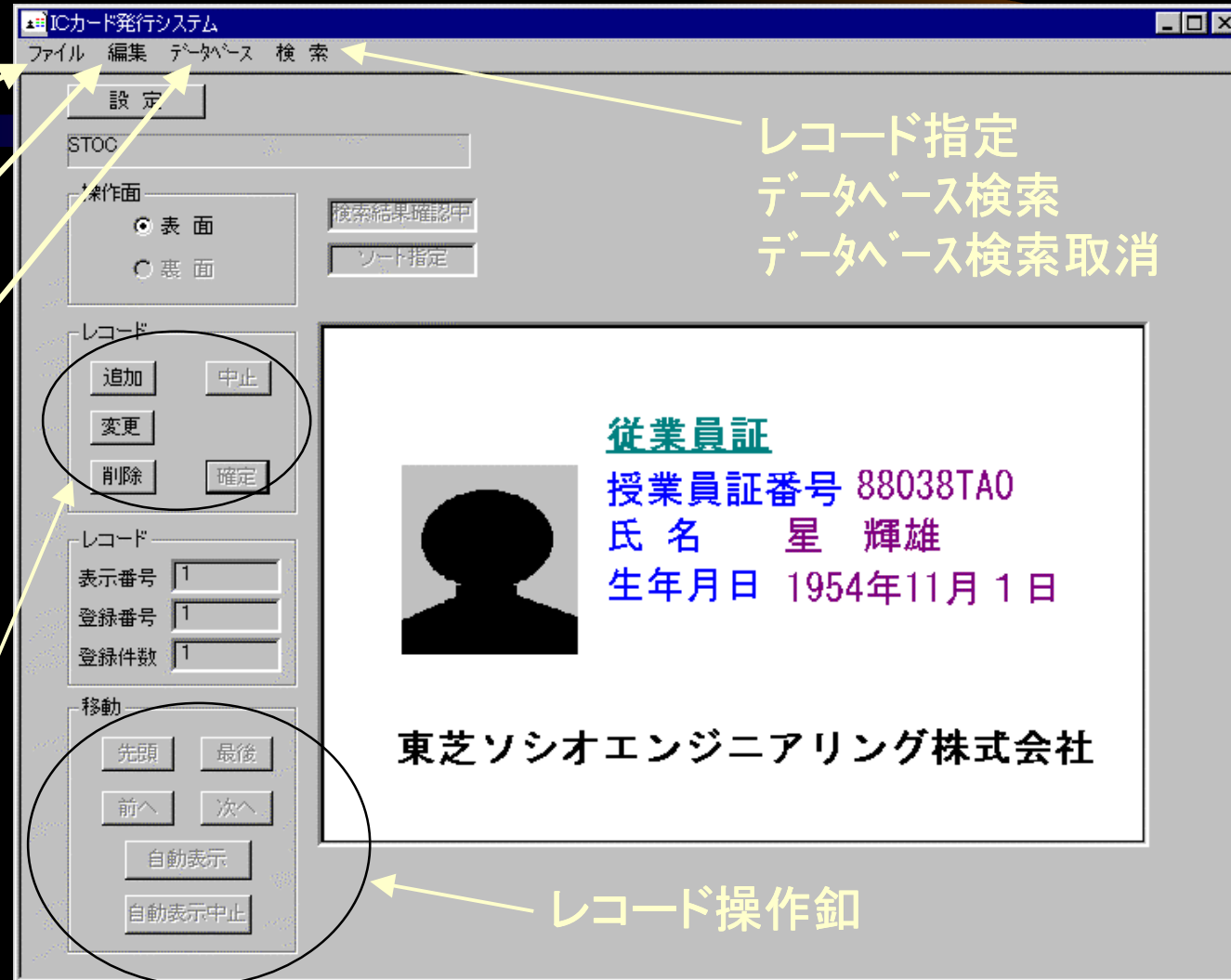
デザイン編集

自動追加
ソート
コピー
一括修正
一括削除
一覧表示

レコード作成釦

レコード指定
データベース検索
データベース検索取消

レコード操作釦



マンマシン メイン画面

ICカード発行システムとは

デザイン編集

操作面
 表面
 裏面

デザイン種別
 アイテム
 オーバーコート

追加中アイテム

追加アイテム

アイテム

外部結合

未印刷領域設定

データベースリンク

完了

取消し

カード領域

従業員証

従業員証番号 番号:入力フィールド

氏名 氏名:入力フィールド

生年月日 生年月日:入力フィールド

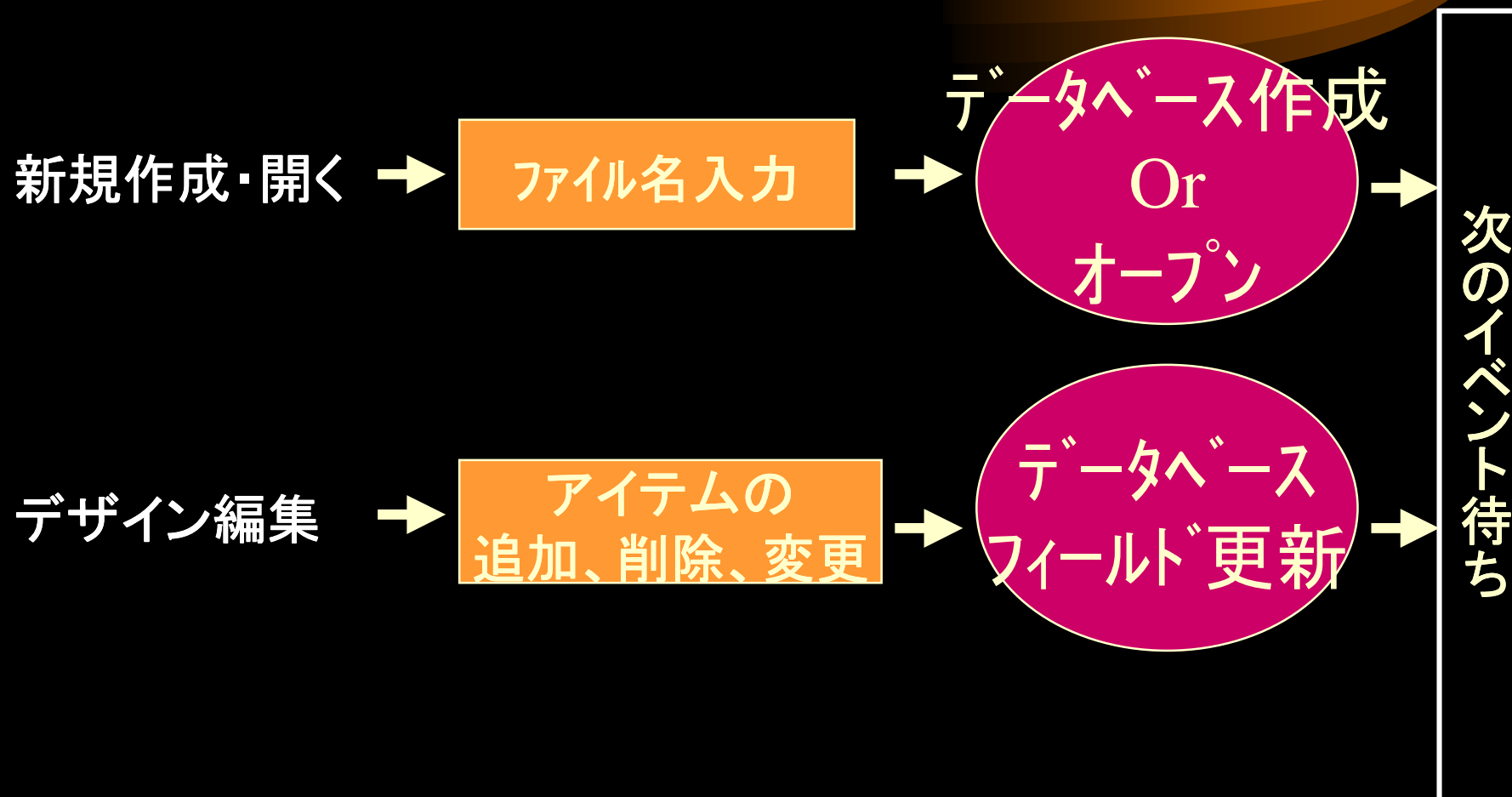
東芝ソシオエンジニアリング株式会社

マンマシン デザイン編集画面

主なマンマシン機能説明

メニュー、釦選択によるイベントドリブン。状態で活性化。

処理中はメニュー・釦の状態変更、完了後元へ戻す



主なマンマシン機能説明

メニュー、釦選択によるイベントドリブン。状態で活性化。

処理中はメニュー・釦の状態変更、完了後元へ戻す

追加、変更



アイテムへ
データ入力



データベース
追加／変更



次のイベント待ち

削除



確認中のレコード

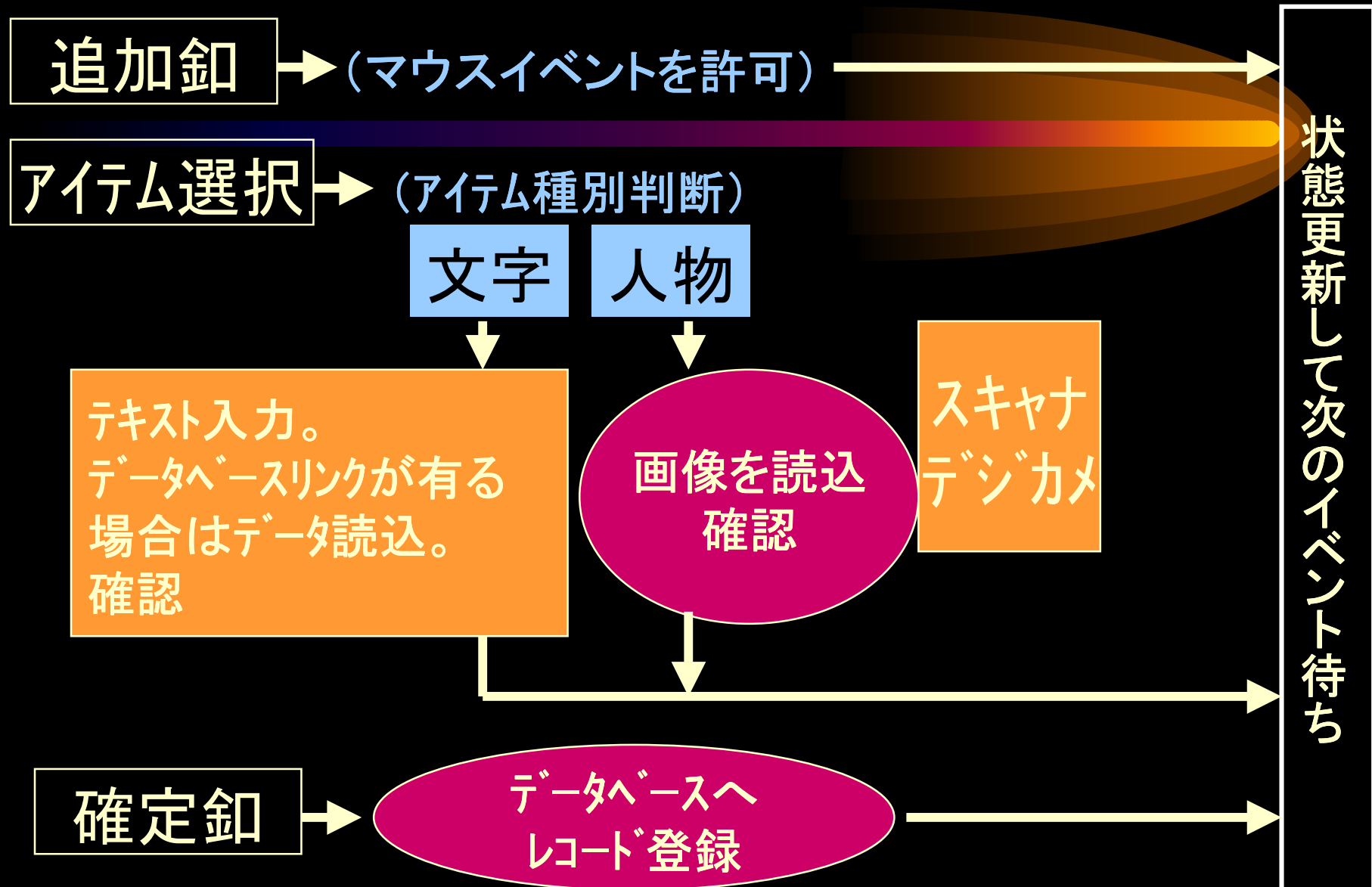


データベース
レコード削除

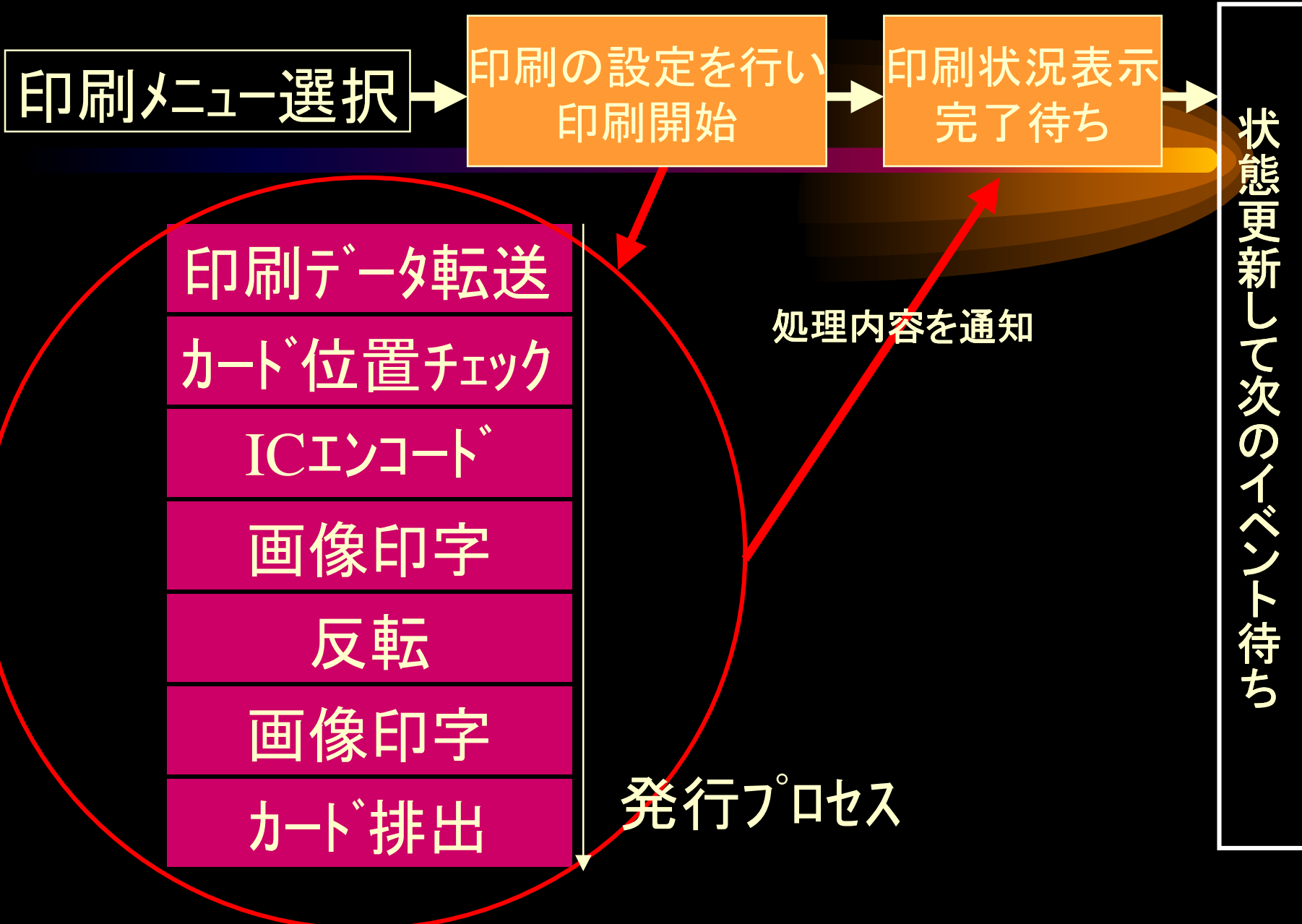


次のイベント待ち

[追加]処理説明



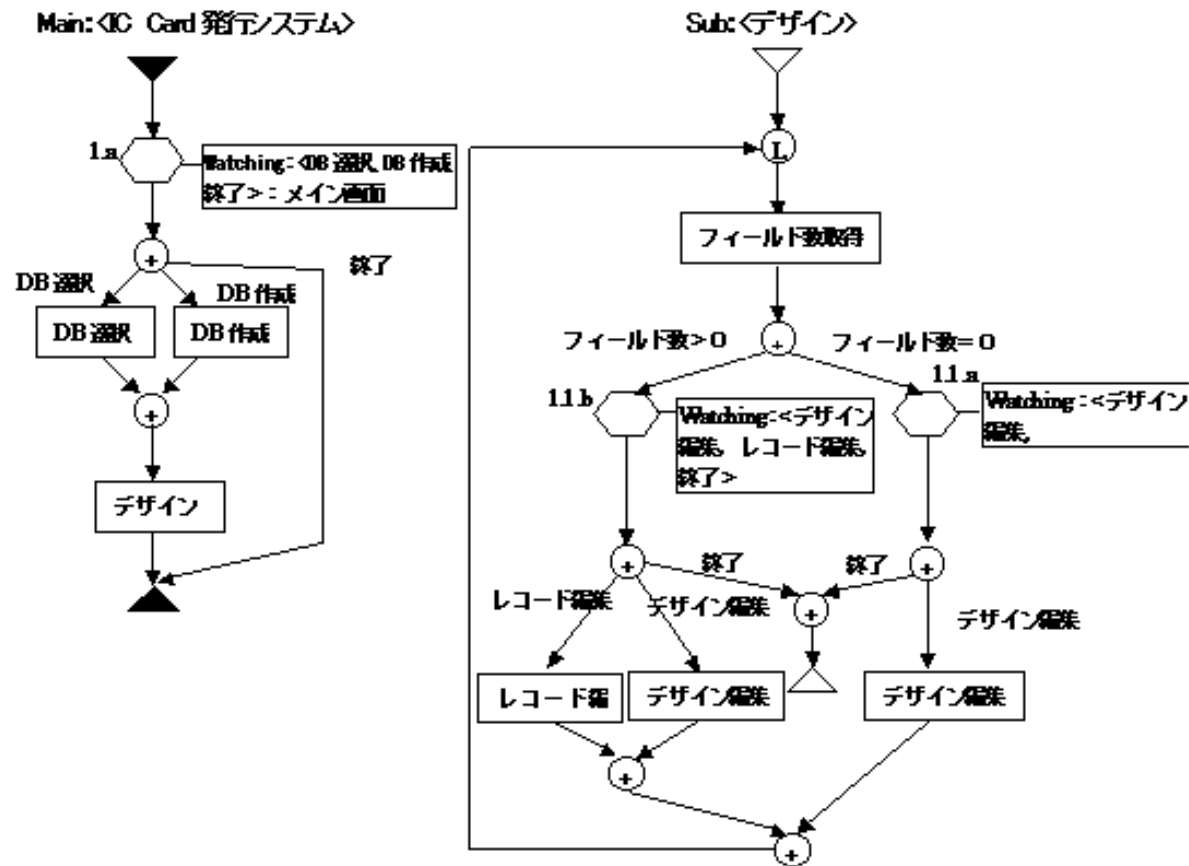
[印刷(カード発行)]処理説明



GUI部分の記述例

組込みシステムにおけるSpecCの応用

<ReactiveBehavior-Net の適用>



武蔵工業大学、SpecC記述前の設計記述(MST2000発表事例)

SpecC研究事例(記載例)

ReactiveBehavior-Net の構図規則と意味			
Esterel	ReactiveBehavior-Net	Esterel	ReactiveBehavior-Net
await S			
do stat upto S		loop stat each S	
every S do stat end		sustain S	

外部装置のSpecC記述例

The screenshot displays a SpecC development environment. On the left, a project tree shows a 'Main' component with sub-components for 'データベース::B0', '画像入力::B1', and 'カード発行::B2', along with a 'global' component. On the right, a table titled 'グローバル変数宣言' (Global Variable Declaration) lists 12 global variables of type 'int'. Each variable is associated with a comment and a set of actions (e.g., 'カード発行実行', '発行', '準備', 'データベース', '画像入力', '磁気エンコード').

	型	変数名	コメント	ユーティリティ
1	int	cardAllow	カード発行処理内の行先決定用	カード発行実行
2	int	cardICEncode	カード発行でICエンコード有無	発行 カード位置
3	int	cardFace	カード発行で両面印刷有無	発行 画像転送
4	int	cardOutBusu	カード発行の部数	次発行
5	int	cardOutStart	カード発行開始位置	発行 画像転送
6	int	cardOutEnd	カード発行終了位置	発行 次発行
7	int	cardOutCount	カード発行の部数カウンタ	準備 次発行
8	int	cardFaceFlg	カード発行の印刷面のフラグ	発行 画像転送
9	int	dbAllow	DB実行での行先決定	データベース DB実行
10	int	imgAllow	画像入力での行先決定	画像入力 IMG実行
11	int	cardMGEncode	カード発行で磁気エンコード有無	発行 磁気エンコ
12				

データベース、画像入力、カード発行を並列に記述
状態遷移の判断用データをグローバル宣言

共通サブルーチン部分をチャンネルで

Channel View : CShared2

CShared2

I/Oポ-ト宣言 ロー加変数宣言 実行コ-ト

外部イテ-ィク

```
typedef struct DBDataArea_Tag {
    char rcvBuff[256];
    char DBName[36]; //データベース名
    int FieldCnt; //フィールド件数
    char FieldTable[20]; //フィールド定義有無 0:無し、1:有り
    char FieldName[20][36]; //フィールド名
    //
    int RecordCount; //登録コード件数
    struct {
        char FieldVal[20][64]; //フィールド内容
    } Record[100];
    //
} DATABASE_AREA;

DATABASE_AREA dbArea;
int dbOpenFlg = 0;

int FieldIndex (char #name);
int AsciiBin (char #data);
void TestMsg (int val, char #msg);

void dbAreaRW (int cmd, int #size, char #data)
{
    int i,j,k,m,n, val, ix;
    char buff[128];
    FILE #fp;
    char #src, #dst;

    switch (cmd) {
    case 0:
        for (i=0; (i < 256)&&(*(data+i)); i++) dbArea.rcvBuff[i] = *(data+i);
        dbArea.rcvBuff[i] = 0;
        break;
    case 1: //DB登録
        for (i=0; (i < 36)&&(dbArea.rcvBuff[i+2] != ':')&&(dbArea.rcvBuff[i+2]); i++)
            dbArea.DBName[i] = dbArea.rcvBuff[i+2];
        dbArea.DBName[i] = 0;
        dbArea.FieldCnt = 0;
        for (i=0; i<20; i++) {
            dbArea.FieldTable[i] = 0;
```

コマンドは状態遷移で振り分け

Behavior View : DB実行

DB実行

	型	変数名	コメント	R/W
1	CShared	dbChanel1	DB 応答データ R/W	ローカ
2	CShared2	dbChanel2		ローカ
3	int	cardAllow	カード発行処理内の行先決定用	グローバル
4	int	cardICEncode	カード発行でICエンコード有無	グローバル

変数接続の編集

I/Oポート宣言 追加変数宣言 ポート接続関係 実行コード **状態遷移**

開始 DB受信::B0

	タイプ	From	To	条件
1	condition	DB受信::B0	DB登録::B1	dbAllow == 1
2	condition	DB受信::B0	フィールド登録::	dbAllow == 2
3	condition	DB受信::B0	DBオープン::B	dbAllow == 3
4	condition	DB受信::B0	フィールド名取得	dbAllow == 4
5	condition	DB受信::B0	フィールド値追加	dbAllow == 5
6	condition	DB受信::B0	フィールド値更新	dbAllow == 6
7	condition	DB受信::B0	フィールド削除::	dbAllow == 7
8	condition	DB受信::B0	登録件数取得::B	dbAllow == 8
9	condition	DB受信::B0	フィールド値取得	dbAllow == 9
10	condition	DB受信::B0	DB終了::B10	dbAllow == 99
11	condition	DB受信::B0	DBクローズ::B	dbAllow == 10
12	condition	DB受信::B0	レコード削除::B	dbAllow == 11
13	condition	DB受信::B0	フィールド数取得	dbAllow == 12
14	condition	DB受信::B0	全フィールド名取	dbAllow == 13

Main

- データベース::B0
 - DB実行::B2
 - DB受信::B0
 - DB登録::B1
 - フィールド登録::B2
 - DBオープン::B3
 - フィールド名取得::B4
 - フィールド値追加::B5
 - フィールド値更新::B6
 - フィールド削除::B7
 - 登録件数取得::B8
 - フィールド値取得::B9
 - レコード削除::B13
 - DBクローズ::B12
 - フィールド数取得::B14
 - 全フィールド名取得::B15
 - 全フィールド値取得::B16
 - 一括フィールド値追加::B18
 - 一括フィールド値更新::B17
 - DB終了::B10
 - DB返信::B11
 - DBLOOP::B3
- 画像入力::B1
- カード発行::B2
- global

チャンネルを利用するポート宣言

フィールド登録

	型	変数名	コメント	R/W
1	DB_Write	db_Port1		RW
2	DB_Area	db_Port2		RW
3	int	cardAllow	カード発行処理内の行先決定用	
4	int	cardICEncode	カード発行でICエンコード有無	

変数接続の編集

I/Oポート宣言

	I/O	型	ポート名	コメント
1	inout	DB_Write	db_Port1	
2	inout	DB_Area	db_Port2	
3				

フィールド登録::B2

Main

- データベース::B0
 - DB実行::B2
 - DB受信::B0
 - DB登録::B1
 - フィールド登録::B2
 - DBオープン::B3
 - フィールド名取得::B4
 - フィールド値追加::B5
 - フィールド値更新::B6
 - フィールド削除::B7
 - 登録件数取得::B8
 - フィールド値取得::B9
 - レコード削除::B13
 - DBクローズ::B12
 - フィールド数取得::B14
 - 全フィールド名取得::B15
 - 全フィールド値取得::B16
 - 一括フィールド値追加::B18
 - 一括フィールド値更新::B17
 - DB終了::B10
 - DB返信::B11
 - DBLOOP::B3
- 画像入力::B1
- カード発行::B2
- global

画像入力の状態遷移

Behavior View : 画像入力

型 変数名 コメント R/W 定

1	int	imgErrorCode			0-加
2	int	imgSleepVal			0-加
3	int	cardAllow	カード発行処理内の行先決定用		0-加
4	int	cardICEncode	カード発行でICエンコード有無		0-加

変数接続の編集

I/O宣言 0-加変数宣言 接続関係 実行コード 状態遷移

開始 IMG初期化::B0

	タイプ	From	To	条件
1	completion	IMG初期化::E	IMG受信::B1	
2	completion	IMG受信::B1	IMG実行::B2	
3	condition	IMG実行::B2	IMG受信::B1	imgAllow == 0
4	completion	IMG実行::B2	IMG返信::B3	
5	condition	IMG返信::B3	IMG受信::B1	imgAllow != 99
6	completion	IMG返信::B3	EXIT	
7				

Behavior View : 画像入力

Main

- データベース::B0
- 画像入力::B1
 - IMG初期化::B0
 - IMG受信::B1
 - IMG実行::B2
 - IMG前処理::B0
 - IMG画像入力::B1
 - IMG終了::B2
 - IMG返信::B3
- カード発行::B2
 - 実行::B0
 - CARD受信::B2
 - 準備::B0
 - 発行::B1
 - CARD終了::B3
 - CARD返信::B4
 - CARDLOOP::B1

global

カード発行の状態遷移

Main

- データベース::B0
- 画像入力::B1
- カード発行::B2
 - 実行::B0
 - CARD受信::B2
 - 準備::B0
 - 発行::B1
 - CARD前処理::B8
 - 画像転送::B0
 - カード位置確認::B1
 - ICエンコード::B2
 - 印字処理::B3
 - 磁気エンコード::B7
 - 反転::B4
 - カード排出::B5
 - 次発行::B6
 - CARD終了::B3
 - CARD返信::B4
 - CARDLOOP::B1

型	変数名	コメント	R/W	定義場所
CARD_Area	card_Port1			ポート
int	cardAllow	カード発行処理内の行先決定用	R	グローバル変数
int	cardICEncode	カード発行でICエンコード有無	R	グローバル変数
int	cardFace	カード発行で両面印刷有無	R	グローバル変数

開始	タイプ	From	To	条件
CARD前処理::B8	condition	CARD前処理::	画像転送::B0	((cardFace & cardFaceFlg) != 0)
	condition	CARD前処理::	カード位置確認::	(cardICEncode==1)&&(cardFaceFlg==1)
	condition	CARD前処理::	印字処理::B3	((cardFace & cardFaceFlg) != 0)
	condition	CARD前処理::	磁気エンコード::	(cardMGEncode==1)&&(cardFaceFlg==1)
	condition	CARD前処理::	反転::B4	(cardFace & 2)&&(cardFaceFlg==1)
	completion	CARD前処理::	次発行::B6	
	condition	画像転送::B0	印字処理::B3	((cardICEncode == 0) cardFaceFlg == 2)&&(cardA
	condition	画像転送::B0	次発行::B6	(cardAllow == -1)
	completion	画像転送::B0	カード位置確認::	
	completion	カード位置確認::	ICエンコード::	
	condition	ICエンコード::	印字処理::B3	((cardFace & cardFaceFlg) != 0)&&(cardAllow != -1
	condition	ICエンコード::	磁気エンコード::	(cardFaceFlg==1)&&(cardMGEncode==1)&&(cardAllow !
	condition	ICエンコード::	反転::B4	(cardFace & 2)&&(cardFaceFlg==1)&&(cardAllow != -
	condition	ICエンコード::	次発行::B6	(cardAllow == -1)
	completion	ICエンコード::	カード排出::B5	
	condition	印字処理::B3	磁気エンコード::	(cardFaceFlg==1)&&(cardMGEncode==1)&&(cardAllow !
	condition	印字処理::B3	反転::B4	(cardFace & 2)&&(cardFaceFlg==1)&&(cardAllow != -
	condition	印字処理::B3	次発行::B6	(cardAllow == -1)
	completion	印字処理::B3	カード排出::B5	

カード発行／画像転送での実行コード

ヒューマンウェア チャット インターフェース IP 画像転送

Main

- データベース::B0
- 画像入力::B1
- カード発行::B2
 - 実行::B0
 - CARD受信::B2
 - 準備::B0
 - 発行::B1
 - CARD前処理::B8
 - 画像転送::B0**
 - カード位置確認::B1
 - ICエンコード::B2
 - 印字処理::B3
 - 磁気エンコード::B7
 - 反転::B4
 - カード排出::B5
 - 次発行::B6
 - CARD終了::B3
 - CARD返信::B4
 - CARDLOOP::B1

- global

	型	変数名	コメント
1	CARD_Area	card_Port1	
2	int	cardAllow	カード発行処理内の行先決定用
3	int	cardICEncode	カード発行でICエンコード有無
4	int	cardFace	カード発行で両面印刷有無

変数接続の編集

I/Oポート宣言 ロー加変数宣言 ポート接続関係 **実行コード** 状態遷移

外部データ

```

void main( void ) {

    int pos, i, dev;
    char buff[12];

    //動作チェック
    pos = cardFaceFlg - 1;
    card_Port1.cardAreaRW (13, &pos, (char *)0);
    if (pos != 0) { //異常
        cardAllow = -1;
        return;
    }
    //実行有無判断
    // if ((cardFace & cardFaceFlg) == 0)
    // return;

    //状況送信
    buff[0] = (char)(cardFaceFlg-1) + '0';
    dev=10000;
    pos= cardOutStart;
    for (i=1; i<6; i++) {
        buff[i] = (char)(pos / dev) + '0';
        pos %= dev;
        dev /= 10;
    }
    buff[6] = 0;
    i = VS_SendMessage("発行応答", buff);
    waitfor(1000);
}

```

画像転送::B0

今後の予定

GUI開発での活用方法をICカード発行システムで検証し使用方法の確立を計る。

メカ動作を含むシステムへの適用を検討する。

製品組込型のソフトウェア開発に於いてSpecCで設計を行い評価。



以上、発表をおわります。