



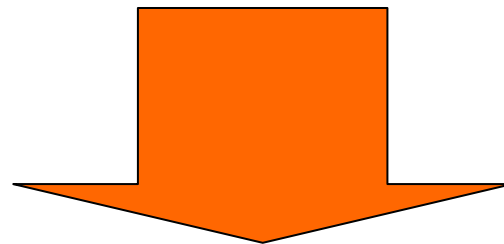
データベースによる多言語対応とGUI開発の効率化

エンサーク株式会社
テクノロジー・エバンジェリスト 高橋君夫



多言語対応とGUIの問題点

- ◆ レガシーなコードではソースにGUIのテキストが埋め込まれている
- ◆ 開発がGUI用のテキストをメンテナンス
 - 実装だけでなくメンテナンスも開発者の負荷
 - GUIがリッチになれば表示テキストの量も膨大になる
- ◆ GUI用テキストの開発・メンテがアプリケーションと分離できない
- ◆ シミュレーターの有効活用があまり出来ていない

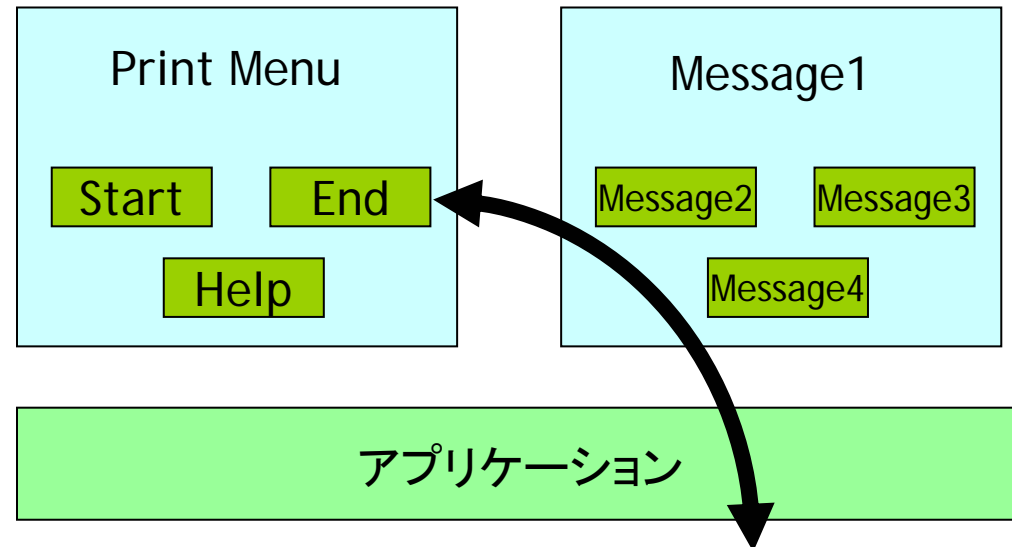


作業負荷の増大と生産効率の低下

より効率的なGUI開発の為に

◆ GUI用テキスト・画像とGUIのフォームを分離

- アプリケーションは常に同じフォームをアクセス
- フォームデータは言語毎のデータと関連付け



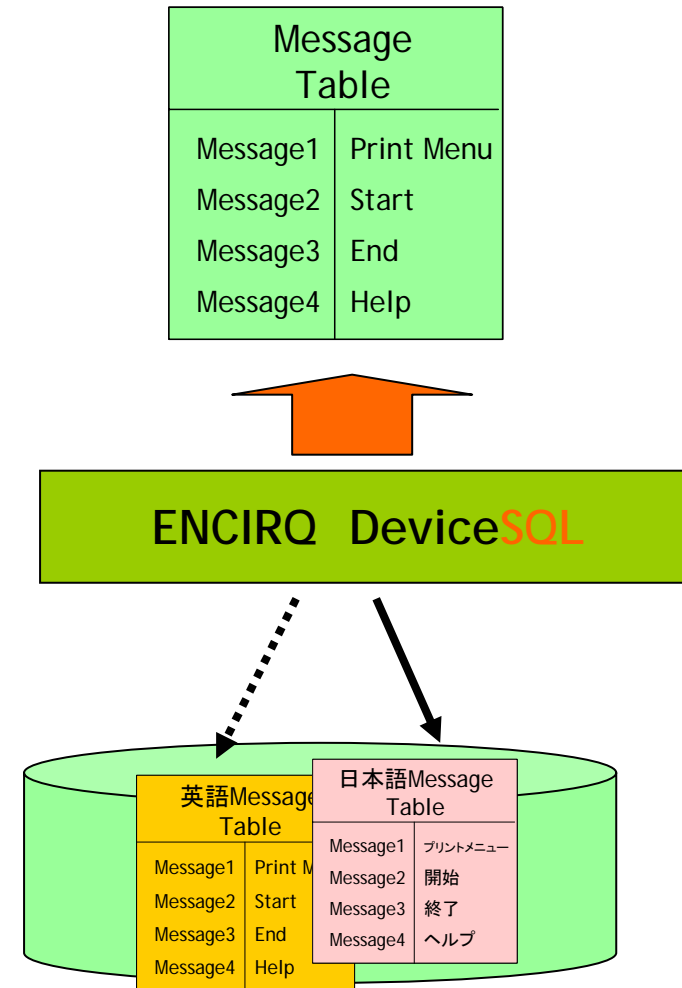
効率上がるが誰がこの仕組みを作るのか？

日本語Message Table	
Message1	プリントメニュー
Message2	開始
Message3	終了
Message4	ヘルプ

英語Message Table	
Message1	Print Menu
Message2	Start
Message3	End
Message4	Help

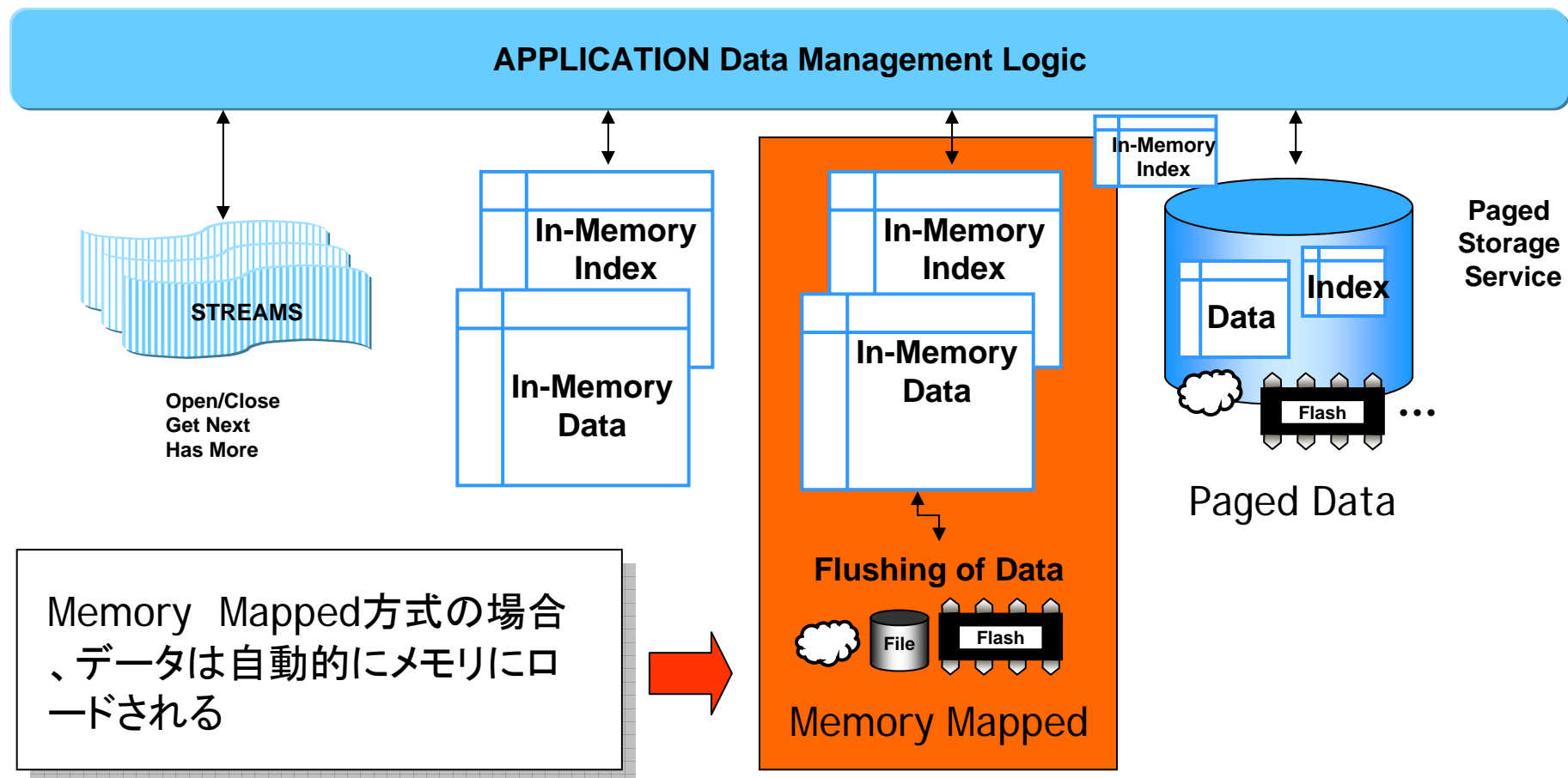
GUIテキストの分離にDBを活用

- ◆ 各言語のテキストデータを独立したDBファイルとして作成
- ◆ コマンドによる言語DBの切り替え
 - 自動的にメモリ中のテキスト用データをDBの機能で入れ替える
 - アプリケーションはDBを意識せず開発できる



DeviceSQLの幅広いデータ参照(格納)方式

- ◆ Memory Mapped方式を活用してテキストデータのメモリへの読み込みをデータベースにまかせる



Memory Mapped方式の場合、データは自動的にメモリにロードされる

SM_OPEN/SM_CLOSEによるDB切替

- ◆ Demoで言語を切り替えているコードはこれだけ

```
-- 英語DBへの切り替え  
CREATE PROCEDURE SwitchToEng  
AS  
  EXPORT NAME "db_SwitchToEng";  
BEGIN  
  SM_CLOSE('ErrMsg');  
  SM_OPEN('ErrMsg', 'ErrMsgEng.edb');  
END;/
```

```
-- 日本語DBへの切り替え  
CREATE PROCEDURE SwitchToJpn  
AS  
  EXPORT NAME "db_SwitchToJpn";  
BEGIN  
  SM_CLOSE('ErrMsg');  
  SM_OPEN('ErrMsg', 'ErrMsgJpn.edb');  
END;/
```

