Byz-Morphosis

Reforming the Lifecycle of Business Transaction Systems (BTS)

through Meta-Application Reuse and Time-induced Metamorphism

Postgraduate Research by Chris Green (supervised by Dr. Ezra Mugisa)
Component-Based Software Engineering – The MORRESA PROJECT
www.mona.uwi.edu/dmcs/research.htm
University of the West Indies, Mona Campus (Jamaica)

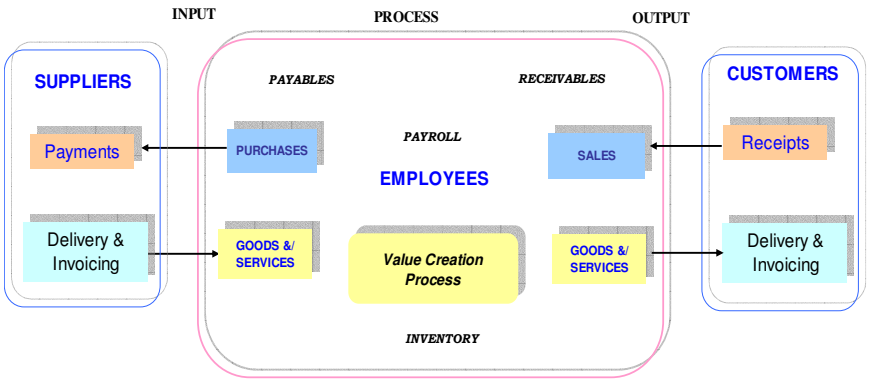
Chris Green is chief architect of Krys Financial Software
Krys Financials
www.krysglobal.com

… please keep an open mind.

ByzMorphosis: Meta-Application Reuse and Time-induced Metamorphism
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Enterprise solutions not Evolvable?

It's a Myth: Businesses cannot Evolve through their lifecycle stages with Static IT systems.



The Problem statement

Within the context of this research, a formal problem statement would be presented as:

Provide a software engineering approach (architecture and framework) that utilizes object-oriented methodologies within the domain of evolving business requirements to:

- **Enable developers to create just-in-time BTS solutions**
  (application time-to-market and agility)
- **Improve the adaptability and interoperability of BTS solutions relative to the recurrent changes in business requirements**
- **Extend the useful lifespan of BTS solutions to match organizations' lifespan**
  (application reusability)
- **Reduce the negative impacts of *Software Evolution* on BTS solutions**
  (time-induced application scalability to meet evolving requirements)

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Evolution: crisis, attempts & proposal

CRISIS:

Despite years of IT innovations, developers have failed persistently to establish systems that effectively fulfill the **evolving requirements** of organizations. [Duggan 2004]

Reason: *Systems Analysis (a Waterfall method) uses "point requirements" to produce "point solutions" which are not adaptable to changes, resulting in* ***"static IT"****.*

Evolution: crisis, attempts & proposal

ATTEMPTS:

By 2008-2010, widespread **automated** application development and program **generation tools**, will significantly reduce the dependency on programmers and coding by hand. [Gartner 2005]

Analysis: *a faster creation of "point solutions" does not adequately address the requirements evolution problem.*

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Evolution: crisis, attempts & proposal

This research "Pattern Mining" principle: *If summer is normally a dry season then storing sufficient water represents a predictive solution in pursuit of this recurrent challenge.*

PROPOSAL:

This research abstracts **patterns of evolving requirements** in the form of **reusable** objects and apply configurations **(meta-data)** to control the variations of object instantiation and adaptability.

Resolution: *refractor patterns of business evolution via Domain Engineering to produce "holistic, reusable and adaptable solutions".*

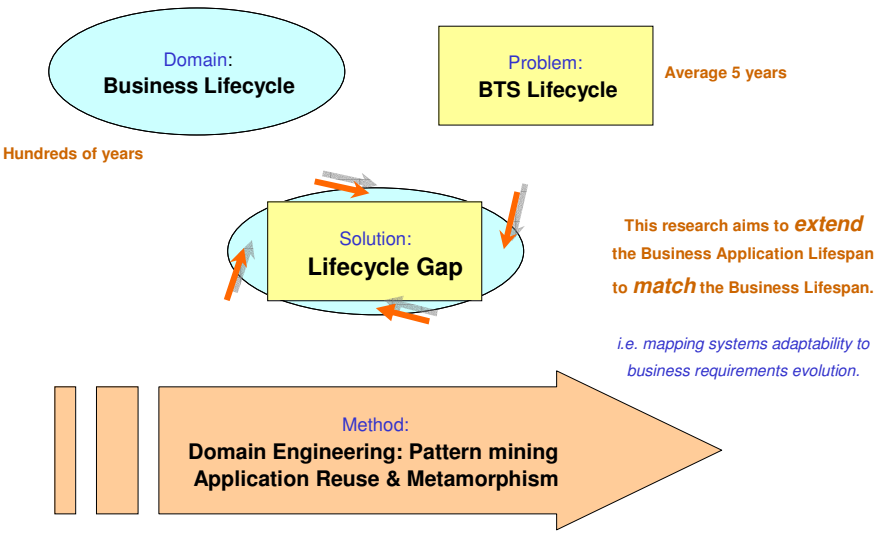
Evolution: crisis, attempts & proposal

RE-POSITION (Contribution):

Leveraging the proposed "Adaptable solution" combined with Systems Analysis being applied at the points of adaptability, transition and re-configuration results in ***"evolvable IT"****.*

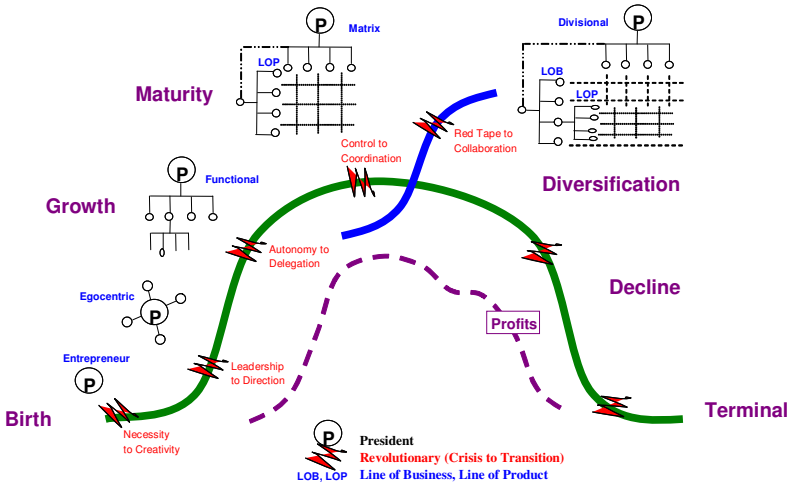
ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*

The Research Roadmap



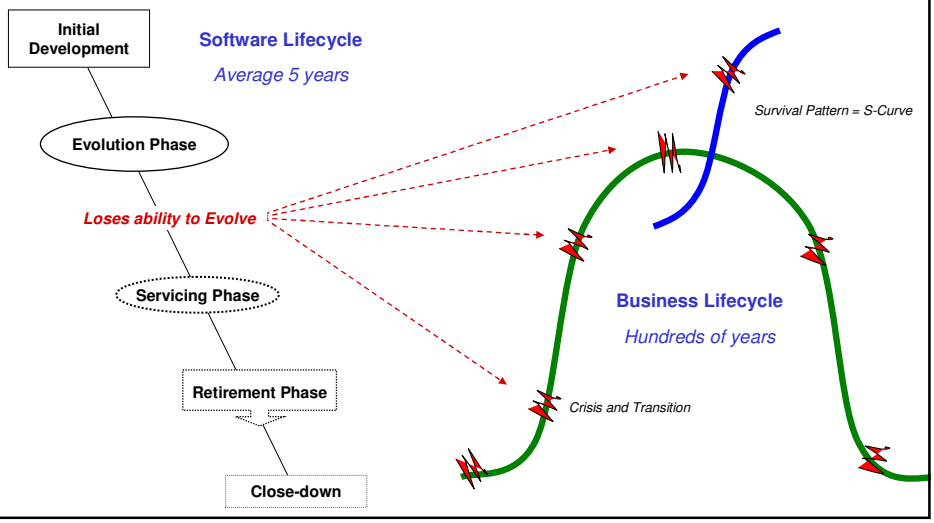
Mapping the Lifecycle patterns

The Survival pattern of an organization resembles a re-cycle of S-shaped curves.



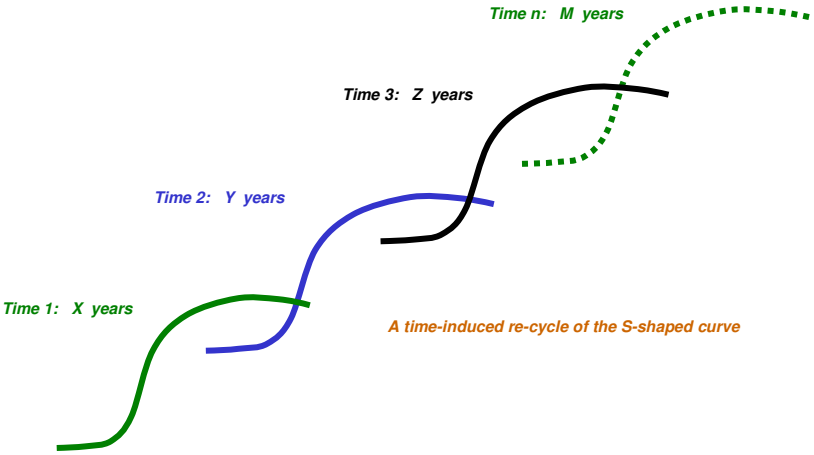
Identifying the Lifecycle Gap

BTS loses its ability to evolve at points of crisis and transition within the business lifecycle.



S-Curve survival patterns

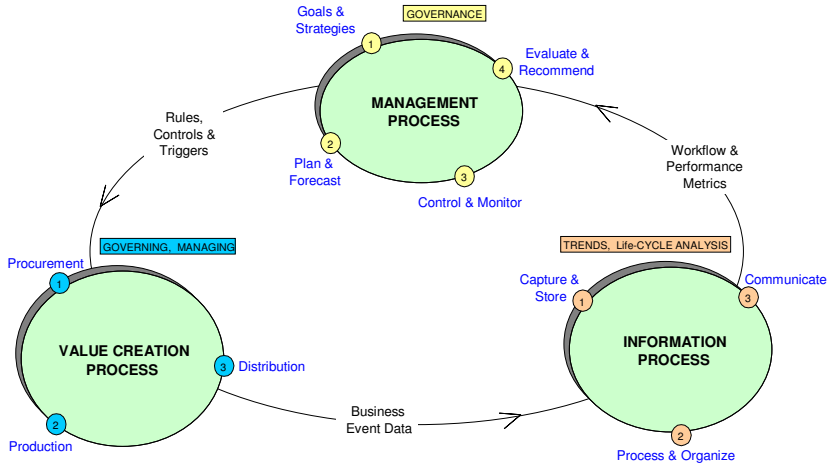
Abstracting and encapsulating the S-shaped curve factored by variants of time represents a fundamental basis for mapping systems adaptability to business requirements evolution.



Demystify the Business

Organizations are coordinated systems of human activities.

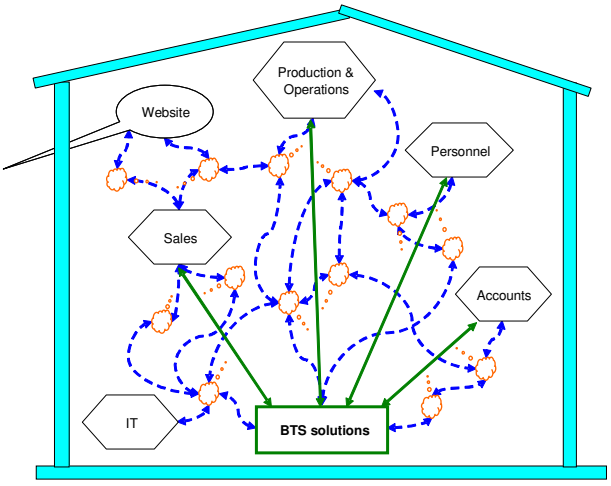
Business is a re-cycle of a finite set of patterns /processes.



The Gap = ?flow culture

Business don't change. People in business change their work-task flows (routes, schedules, strategies)

Unfortunately, most BTS solutions have hard-coded work-task flows (fixated on the Money-Flow).



MONEY-FLOW

WORKFLOW CHANGES
- Routes
- Authorizations
- Strategies
- Regulations
- Customer Notices

SUPPORTED BY
- Spreadsheets
- Word Processors
- Rubber Stamps
- Filing Cabinets

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Software Engineering challenge

The software development lifecycle has *two inherent limitations* that impede the creation of *readily available* systems to meet business changes.

1.) The time it takes to deliver *solutions are often late* due to the frequency and number of changes in business.

2.) Difficulties in predicting future requirements, often cause developers to *rely on end-user's dated information* and "point requirements" to build systems for future business challenges.

These limitations triggers *-ve Software Evolution* and magnifies the cost of *software decay*.

This research "Pattern Mining" principle: *If summer is normally a dry season then storing sufficient water represents a predictive solution in pursuit of this recurrent challenge.*

Design Methodologies

Application Reuse

A Meta-Application that *Role-plays in both the Applications and Business Domains*

Metamorphism

Meta-Data (Role Descriptions) *induced morphing, reuse, inheritance and polymorphism*

Byz-Morphosis

BTS Metamorphosis: *time-induced Metamorphism* (applications Role transition and mutation)

Object-orientation

Provides the framework for building and reusing the Meta-Application

(Encapsulating domain patterns and abstractions can improve software composition, maintenance and reuse)

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Meta-Data driven Independence

Database Systems foster *Data Independence* in the application domain.

1. *We can make changes to the Database System (platform or indexes) without changing the application*
2. *One common Database (data-model) can service several instances of an applications*

Encapsulating Meta-Data (application descriptions) fosters different types of Domain Independences.

Meta-Data Types	Domain Independences
Business Rules	both Applications and Rules Engines Independences
Process Flow Logics	both Applications and Flow Engines Independences as well as Applications Process Task Independence
Referential Integrity Rules	both Applications and Database Systems Independences
System State Configurations	Applications State Configuration Independence
User Interface Specifications	both Applications and Interface Systems Independences

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*

Data-Model Independence

BTS solutions are all about representing business transactions processes.

By definition Business Transactions don't Change – they may have different execution paths.

Supported by studies in "Narrative data", this research explores an Atomic Data-Model for business data.

Document Types

Invoices

Payments

Letters

Receipts

* * *

Data Model Reuse

Barter Trading System

(the first form of trade: exchange of goods for goods)

Data-Model Entities

Parties – suppliers, consumers, employees

Products – items of value being traded

Transactions – contracted actions & terms

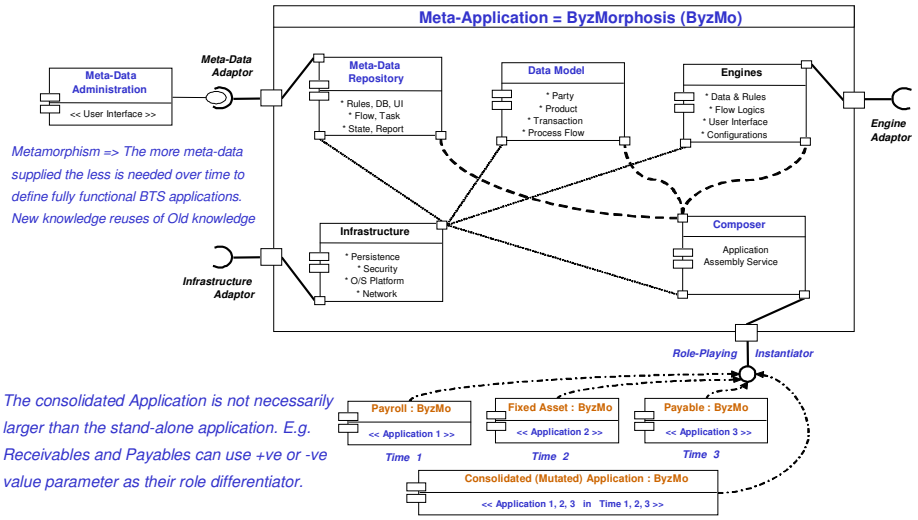
Process Flows – execution path & sequence

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*

Demystify the Meta-Application

A Meta-Application encapsulates several applications (i.e. Application domain independence).

Real-time Meta-Data driven application development provides Just-In-Time BTS solutions.

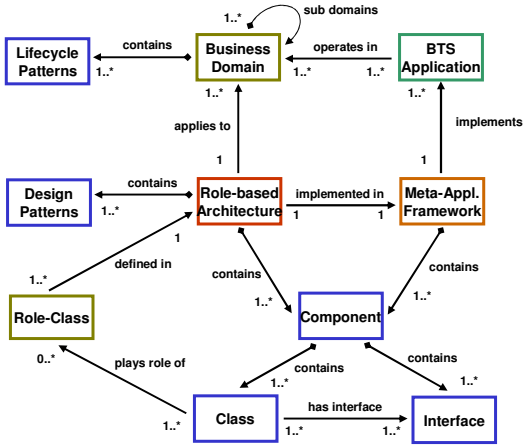


Metamorphism => The more meta-data supplied the less is needed over time to define fully functional BTS applications. New knowledge reuses of Old knowledge

The consolidated Application is not necessarily larger than the stand-alone application. E.g. Receivables and Payables can use +ve or -ve value parameter as their role differentiator.

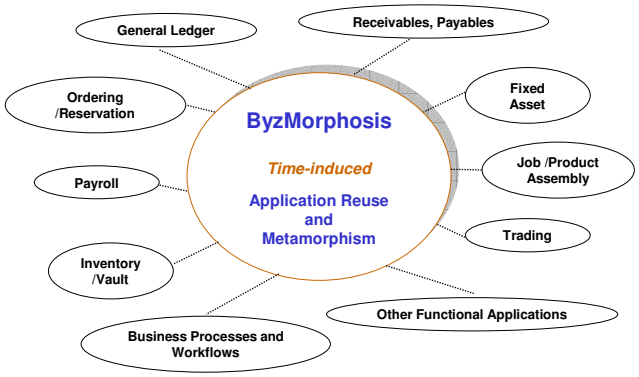
The object-oriented approach

Object-orientation provides an architectural style or framework for building and reusing the Meta-Application. Encapsulating domain patterns and abstractions can improve software composition, maintenance and reuse.



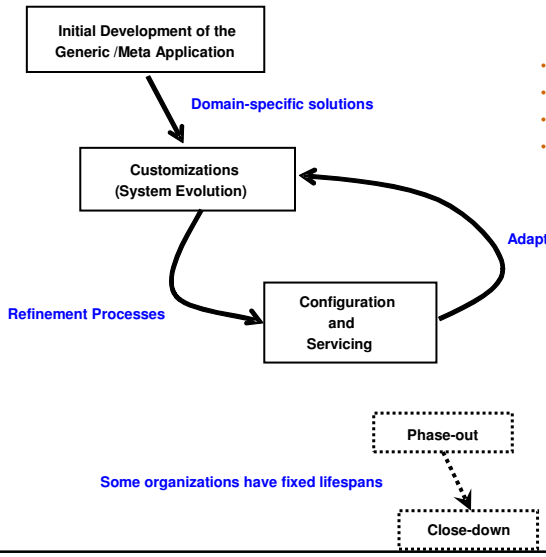
The solution Resolution

The choices of selected solutions are driven by the stages of change, growth or development within the Evolutionary Business Lifecycle of organizations.



The solution Contribution

The Revised Software Lifecycle



- *Just-in-time BTS solutions*
- *Adaptability to changing business requirements*
- *BTS useful lifespan match organizations' lifespan*
- *Reduce the negative impacts of Software Evolution*

ByzMorphosis: *Meta-Application Reuse and Time-induced Metamorphism*
Copyright © 2007 Chris Green, Krys Financial Software. All rights reserved.

Questions and comments