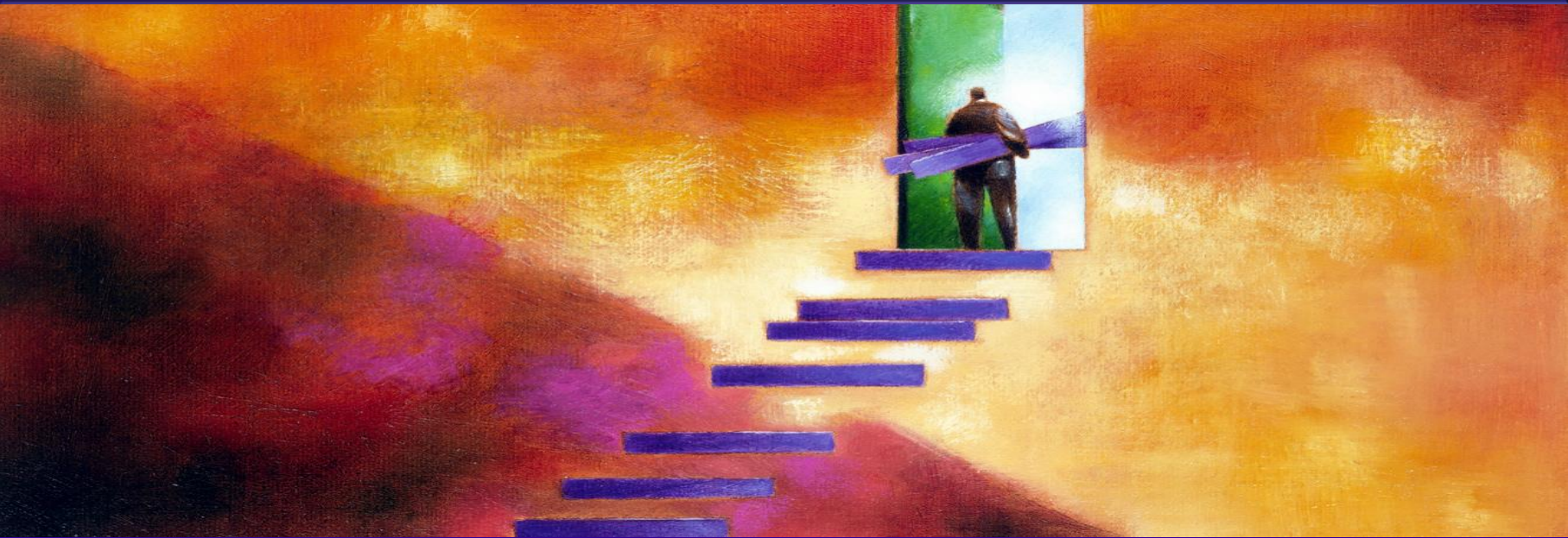


Advanced software technologies for breakthrough applications



Business Intelligence with InterSystems DeepSee: Data modeling, exploration and presentation

Dr. Josef Němeček, Sales Engineer, InterSystems Švýcarsko

INTERSYSTEMS

Agenda



❖ Introduction

- Embedded and real-time business intelligence
 - Differentiation from our competitors
 - Advantages for our application partners
- InterSystems DeepSee for our Application Partners
 - Architecture and design of InterSystems DeepSee
 - Improvements inside InterSystems DeepSee
 - Best practices and common recommendations

❖ Demonstration of InterSystems DeepSee

What is «Business Intelligence»?



- ❖ Howard Dresner (Gartner Group, 1989):
“Concepts and methods to improve business decision making by using fact-based support systems”
- ❖ Wikipedia (2010):
“Skills, processes, technologies, applications and practices used to support decision making”
- ❖ Popular explanation:
“The art of turning numbers into decision”

Who are the Decision Makers?



- ❖ Traditional model: Executives only
 - Make-or-buy, out/in-sourcing, capacity management
 - Financial status, cash management, resources
 - Analysis based on historical, consolidated summaries
 - Strategic decisions based on «slow» data
- ❖ New model: Everybody within his responsibility
 - Priorities in workflows, tasks, preferences
 - Backorder and stock status, customer VIP levels
 - Analysis mainly based on current data and expectations
 - Tactical decisions based on «fast» data

Embedded and Real-Time BI



- ❖ Real-time Business Intelligence
 - Analysis bases on current operational data
 - Drill down to the transaction level possible
 - Short response time

- ❖ Embedded Business Intelligence
 - Integrated into the application
 - Same terminology as the application
 - Automates internal application processes, workflows
 - Integration into the application user interface
 - Good argument for an application update

Competitive comparison



Traditional BI

- ❖ Data extracted, consolidated, and loaded periodically
- ❖ Historical summaries only
- ❖ Specialist users with tool and data scheme expertise
- ❖ Separate-system effect
- ❖ Expensive project
- ❖ High TCO

Embedded Real-Time BI

- ❖ Data accessed in place
- ❖ History & current, detailed
- ❖ All application users that need decision information
- ❖ Uses available system
- ❖ Part of the application
- ❖ Cost-effective, as Caché is

Advantages for our Application Partners



- ❖ BI functionality integrated in Caché & Ensemble
 - BI can be part of a product upgrade
 - BI can be created by Caché developers
 - BI workflows can use BI data as suggestions to users
- ❖ BI functionality integrated into your product
 - Customer does not need a separate BI installation
 - Customer optimizes workflows with BI
 - Managers have BI data from your system
- ❖ BI enabled applications can trigger integration
 - Use other application's data for your BI too

Why next-generation DeepSee?



- ❖ Our customers appreciate DeepSee
 - High customer interest, embedding BI is useful
 - High performance, ease of use, thin client
 - Complex data can easily be handled
- ❖ Some improvements are requested
 - Even higher performance
 - Platform-independent and flexible user interface
 - Partners prefer use of modern Caché functionality

What has changed in DeepSee 2?



- ❖ Data models are implemented as classes
 - Simplification of design and development
 - Independent from the source class code
- ❖ Query execution engine redesigned
 - Access to pivot tables with ResultSet API
 - Pivot tables can be accessed with MDX queries
 - Multi-level caching, multithreading, native indices
- ❖ Platform-independent ZEN-based user interface
 - User portal with dashboards, reports, queries
 - Dashboards can interact with application logic

When is DeepSee 2 available?



- ❖ DeepSee 2 is currently available in a beta program
 - Contact your Sales Engineer for details
 - DeepSee 2 comes as an installable project XML file
 - Limited documentation and training material available
- ❖ DeepSee 2 will be commercially available in 2011
 - Roadmap of feature set not yet defined
 - Pricing will be similar to the existing model

DeepSee Key Capabilities



Architect

Define data models

Analyzer

Explore and display data

Designer

Create dashboards

Connector

Use external data

DeepSee Architect: Data Modeling



- ❖ Data model inside DeepSee: Cube
 - Contains measures and hierarchies of dimensions
 - Compilation creates fact tables with indices
- ❖ Subject area: “Filtered cube”
 - Contains subsets of cube dimensions and measures
 - Filters data inside the cube (e.g. for user specific data)
 - Different names (e.g. user group dependent)
- ❖ Key Performance Indicator (KPI)
 - Function of measures and dimensions
 - Used for Balanced Score Cards (BSC)

DeepSee Architect: Data Modeling



Menu | Home | About | Help | Logout | DeepSee > Architect

HoleFoods Server: michael-c03a10d Namespace: HOLEFOODS Switch
 User: UnknownUser Licensed to: InterSystems Sales Engineering Instance: E20102

DeepSee by InterSystems

View: [New] [Save] [Compile] [Build] Change Subject Area **Architect**

Data Source [Change](#) Dimensions [Add](#) [Undo](#)

▼ HoleFoods.Transaction
 Actual
 AmountOfSale
 Channel
 DateOfSale
 Discount
 Outlet
 Product
 UnitsSold

HoleFoods		cube	HoleFoods.Transaction	
▼ Measures				
Amount Sold	SUM	AmountOfSale		✖
Units Sold	SUM	UnitsSold		✖
▼ DateOfSale time dimension DateOfSale ✖				
Actual hierarchy ✖				
YearSold	level	Year		✖
MonthSold	level	MonthYear		✖
DaySold	level	DayMonthYear		✖
QuarterSold	level	QuarterNumber		✖
▼ Product dimension ✖				
P1 hierarchy ✖				
Product Category	level	Product.Category		✖
Product Name	level	Product		✖
Name	property	Product.Name		✖
UnitPrice	property	Product.Price		✖
PriceSquared	property	(expression)		✖
SKU	property	Product.SKU		✖
▼ Outlet dimension ✖				
H1 hierarchy ✖				
Region	level	Outlet.Country.Region.Name		✖
Country	level	Outlet.Country.Name		✖
City	level	Outlet		✖
Name	property	Outlet.City		✖

Details [Help](#) <<

Cube Disabled

Name
HoleFoods

Display name

Description

Source Class
HoleFoods.Transaction

Default listing
%ID,DateOfSale,Outlet->City,%EXTERNAL(

Default listing order
DateOfSale,%ID

Default measure

Default member

Owner

Caption
HoleFoods Sales

Bucket size Precompute

How BI can become a bad advisor



- ❖ **Bad data:** Bad data due to broken business rules
 - Missing adjustments of historical data
- ❖ **Overcooked data:** The “brain/gut” distance
 - Incorrect separation of overheads, temporal separation
- ❖ **Missing details:** No transparency
 - No detail listing available
- ❖ **Experts’ disease:** Bending results
 - No “public” access for better coverage/agreement

How BI can become a good advisor



- ❖ Use basic transaction classes as BI source
 - Separate sources create inconsistencies
- ❖ Transparently show data consolidation
 - Make filters visible and easy to understand
- ❖ Use application terms and names
 - New concepts unnecessarily puzzle users
- ❖ Have the users in focus, not the experts
 - Complexity can swamp and/or distract the user

How to create a good data model



- ❖ Look at the available data (bottom-up)
 - Think about what is possible to show
- ❖ Look at the requirements (top-down)
 - Find out what is missing in the available data
- ❖ Close the gap by adding attributes / classes
 - BI creates new application requirements
- ❖ Look for early adopters and business analysts
 - Avoid customers that already have a BI system

DeepSee Analyzer: Explore Data



- ❖ Basic element of Analyzer: Pivot table
 - Column and row headers contain dimensions
 - Cells contain aggregations of measures
- ❖ Representation by MDX queries
 - Standard “defined by” Microsoft
 - Separation of query engine and user interface
- ❖ New look of DeepSee Analyzer
 - Utilization of ZEN technology
 - New “Stop Query” button

DeepSee Analyzer: Explore Data



Menu Home | About | Help | Logout DeepSee > Analyzer
Server: CACHE20101 Namespace: HOLEFOODS Switch
User: _SYSTEM Licensed to: InterSystems Sales Engineering

HoleFoods DeepSee
by InterSystems

View: Analyzer

Open Save Change Subject Area Chart Options

Measures

- Count
- Amount Sold**
- Units Sold

Dimensions

- Channel
- DateOfSale
 - All DateOfSale
 - ▶ YearSold
 - ▶ MonthSold
 - ▶ DaySold
- Discount
- Outlet
- Product
 - All Product
 - ▶ Product Category
 - ▶ Product Name
- UnitsPerTransaction

Rows: Product Category Columns: YearSold Measures: Amount Sold Filters: Drop filter here

Drop row here Drop column here Drop measure here

Filters

	2004	2005	2006	2007	2008
Candy	181.15	540.28	1,322.60	3,694.88	9,305.78
Cereal	1,811.59	4,317.46	10,935.09	26,663.46	71,750.81
Dairy	475.45	1,174.67	2,974.78	6,810.19	17,444.80
Fruit	2,061.36	4,732.31	11,393.54	27,056.09	72,957.56
Pasta	1,423.84	3,573.27	8,467.93	22,446.85	58,435.69
Snack	5,037.62	13,496.33	32,489.14	83,091.38	210,463.16
Vegetable	1,722.06	5,371.32	13,655.25	31,936.37	78,795.60

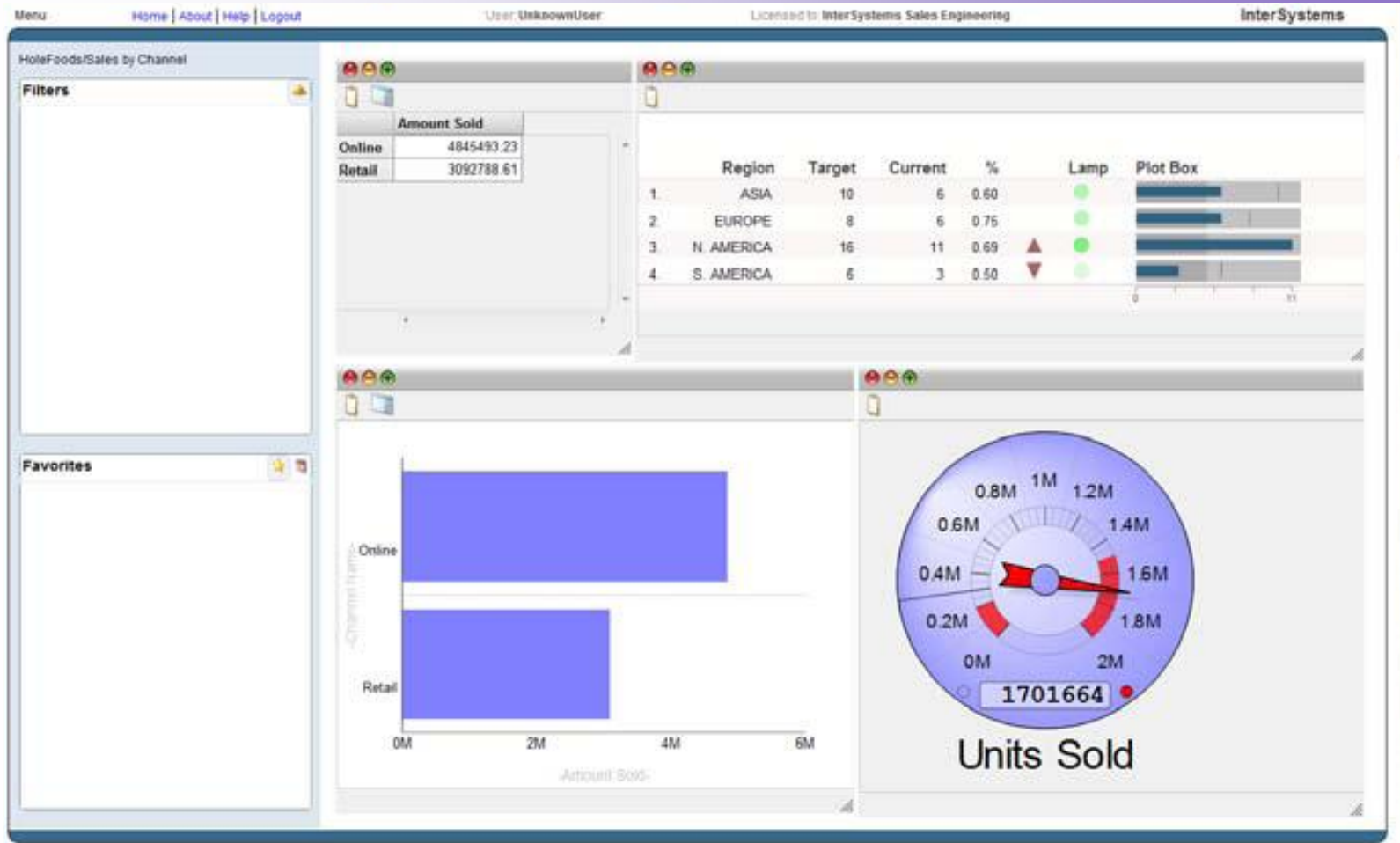
Cells: 35 Total: 848,009.66

DeepSee Designer: Create Portals



- ❖ Presentation of data using dashboards
 - Prepared set of linked tables, graphs, speedometers, ...
 - Global, per user group, per user, private etc.
- ❖ Interactive part of Business intelligence
 - Data can be filtered by user, drilled down, ...
 - Dashboard content can be altered by the user
- ❖ Light-weight user interface client
 - Portal is a ZEN-based web client
 - Platform independent, adaptive, highly interactive

DeepSee Designer: Create Portals



Updating Indices for DeepSee



❖ Real-time (“immediate”) updates

- Add triggers and call-back functions to the base class, which call the fact update functions
 - `do ##class(%DeepSee.Utils).%ProcessFact(...)`
 - `do ##class(%DeepSee.Utils).%DeleteFact(...)`

❖ Scheduled (“deferred”) updates

- Add class parameter “DSTIME” with value “auto” to the base class to create an update index
- Call the synchronization function for updating
 - `do ##class(%DeepSee.Utils).%SynchronizeCube(...)`