



Composable Analytics | dataflow intelligence

# *Handling Variability through Composability*

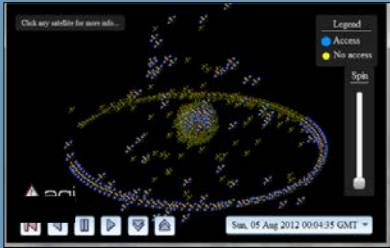
---

North East Database Day 2017

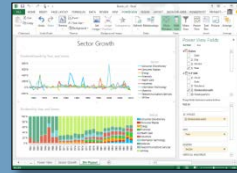
# About Composable Analytics, Inc.

MIT Lincoln Laboratory advanced data analytics spin-off company

## Our Background & Expertise:



**Big Data Analytics**



**Enterprise Information Systems & Business Intelligence**



**Serious Gaming**



**Multi-Agency Distributed Command & Control**

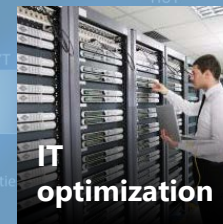
## Our Mission:

Give enterprises the power to consume, manage, explore and harness massive amounts of data

## Our Customers' Domains:



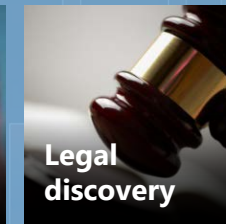
**Network flow analysis**



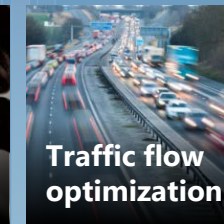
**IT optimization**



**Social network analysis**



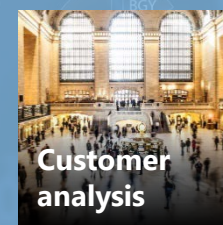
**Legal discovery**



**Traffic flow optimization**



**Churn analysis**



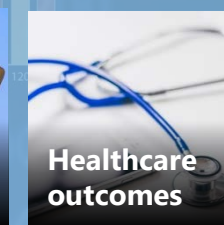
**Customer analysis**



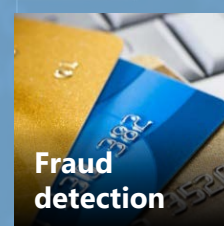
**Natural resource exploration**



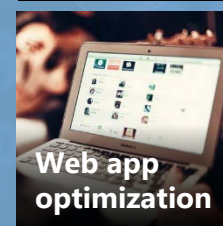
**Defense systems**



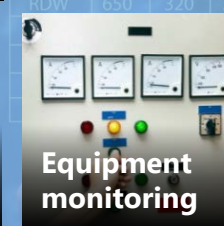
**Healthcare outcomes**



**Fraud detection**



**Web app optimization**



**Equipment monitoring**



**Advertising analysis**



**Financial market analysis**





**Application Variability** has not yet been addressed in a coherent way

# Application Variability

## Characteristics

- Variable workloads
- Unanticipated questions
- New clients and accounts
- Mergers and Acquisitions
- Hypothesis Testing
- Integration of large sets
- 10,000s of small data sets
- Real-time vs. Periodic
- Stream vs. Batch
- Varying departments
- Multiple personas

## Requirements

### Rapid Analysis in a Controlled Environment

• *Infinitely Configurable*

• *Pluggable, Agile, Extensible*

• *Continuous Analytics*

Minimal deployment costs,  
Infrastructure-agnostic

• *No Monolithic Data Model*

• *No Single Data Warehouse*

• *Repeatable process*



Revenue growth divisions.

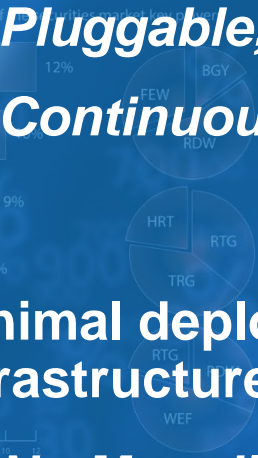
TYU division

FRT division

Distribution marketing participation in the securities market.



Distribution of...



	TYU division			FRT division		
GHT	254	550	254	274	154	415
RDW	650	320	754	273	825	154
RTG	254	550	874	657	125	274
WEF	124	752	741	759		

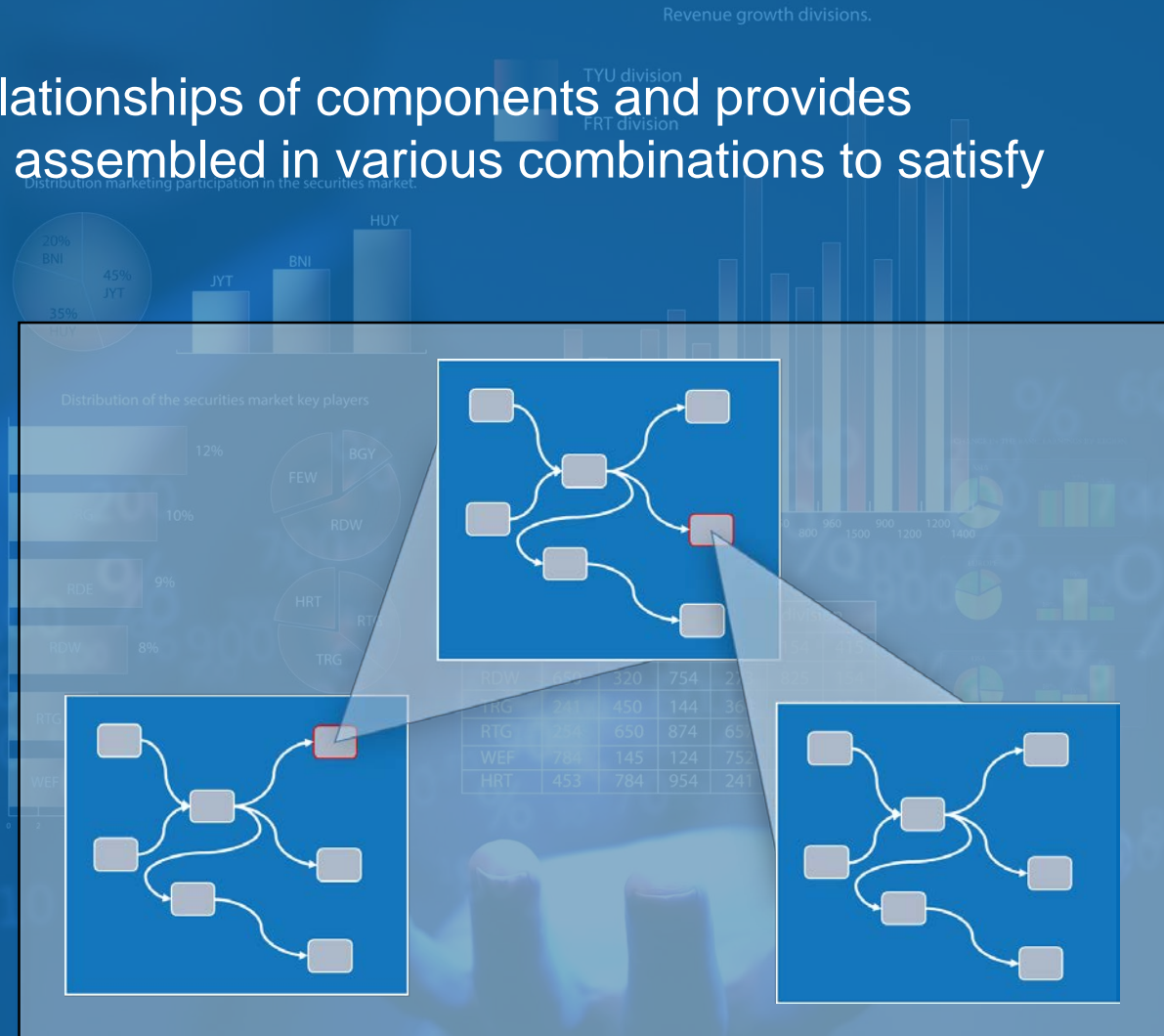


# Composability

System design principle that deals with the inter-relationships of components and provides recombinant components that can be selected and assembled in various combinations to satisfy specific application requirements

## Advantages:

- Pluggable / Reusable components
- Testability
- Late-binding configuration
  - Blurring the lines between code and configuration
  - End-User behavior modification
- Unanticipated need and requirements accommodated
- Carry structured analysis into execution phase
- Just-in-Time coding and analytics
- Define dependencies, not execution order



# Common Threads Across Applications

- *ETL*
- *Analytics*
- *Blending*
- *Reports*

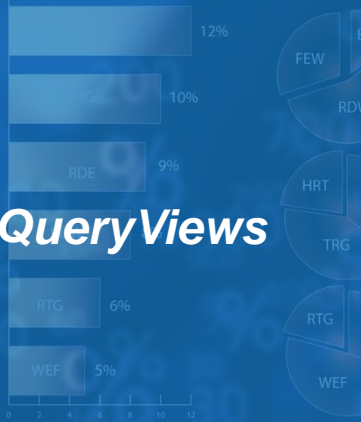


- *Dataflows*

Distribution marketing participation in the securities



Distribution of the securities market key players



- *Data Exploration*
- *Distributed queries*
- *Slicing & Dicing / Drill Downs*
- *Data Exports*



Projected sales of main products in 2013



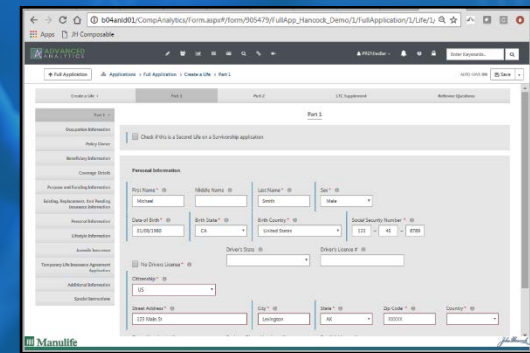
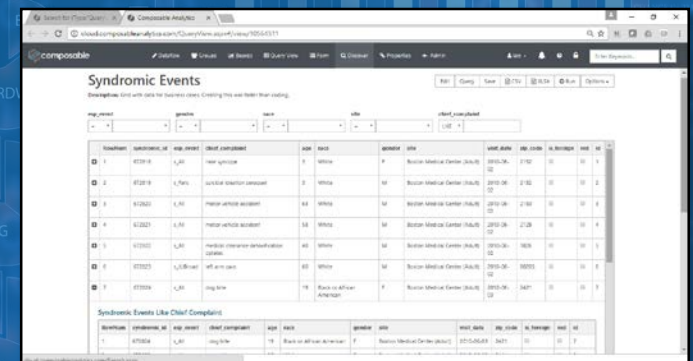
Share of market activity



Changes in the activity of the active and passive market is uncertain. Established positive trends in various market segments.



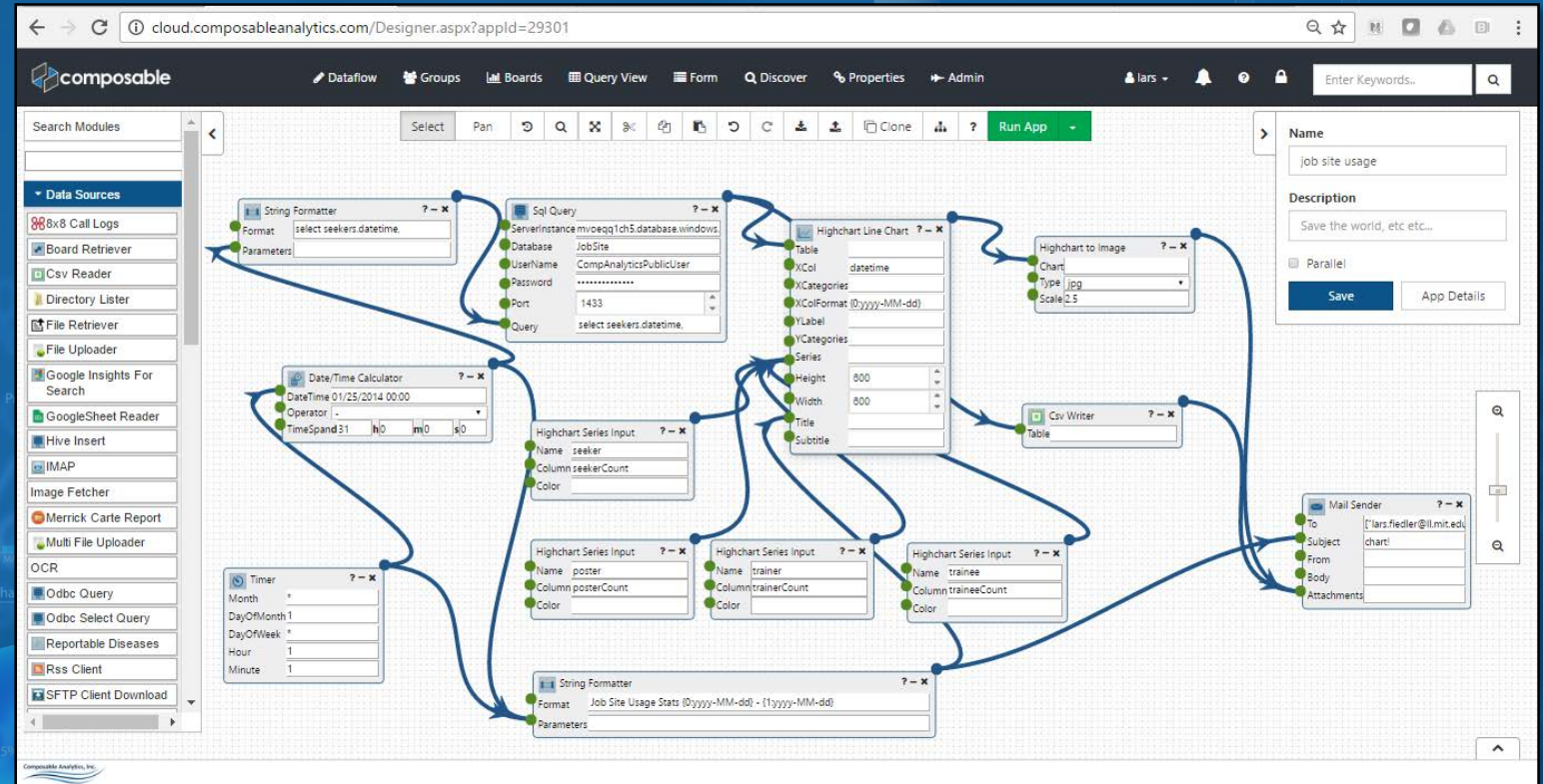
- *Data modeling*
- *Web/business layers*
- *User interfaces*





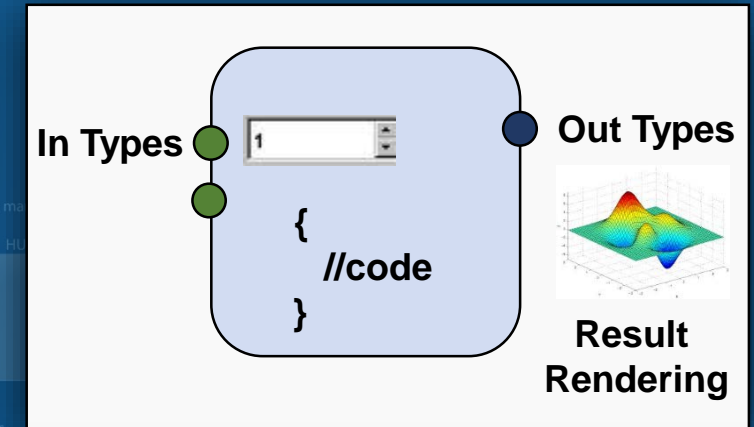
# Composable Dataflows

- ModuleType
  - Arguments
    - Support collections and streams
    - Name, Description, Type
    - Inputs
    - Outputs
- Module
  - Instantiation of ModuleType in an application
- Dataflow
  - Multiple Modules
  - Connections between arguments
  - Can be a Module
    - Externalized inputs and outputs
    - Call other apps or itself



# Module Extensibility Points

- Type System for inputs and outputs
  - i.e. UInt32, String, Boolean, Chart, Table, List<>, Object.
  - Conversions through IConvertible, Casting Operators, Inheritance
- Module execution routines (building blocks)
  - Transform inputs into outputs
- Input Control
  - Define how a user can set an input if not coming in from a connection. “hard-coded” in the app
  - reusable across modules
  - i.e. spin control, dropdown list, region chooser.
- Output Rendering - linked to type system
  - i.e. chart, map, tree view.
- Versioning



```
11 namespace CompAnalytics.Execution.Modules
12 {
13     [ModuleType(Name = "Calculator", Namespace = "edu.mit.ll.companalytics",
14         Category = ModuleCategory.Operator, Icon = "Icons.calculator.png")]
15     [Description("Performs numerical operation on two numbers")]
16     17 references | Jim, 238 days ago | 3 authors, 3 changes | 1 work item
17     public class CalculatorModuleExecutor : ModuleExecutor
18     {
19         [Description("First numerical input")]
20         15 references | 0/6 passing | 0 changes | 0 authors, 0 changes
21         public ModuleInput<double> Param1 { get; set; }
22
23         [IsNotNull]
24         [ComboBoxControl("...", "+", "-", "/", "%", "^")]
25         [Description("Numerical operation")]
26         8 references | 0/2 passing | 0 changes | 0 authors, 0 changes
27         public ModuleInput<string> Operator { get; set; }
28
29         [Description("Second numerical input")]
30         11 references | 0/5 passing | 0 changes | 0 authors, 0 changes
31         public ModuleInput<double> Param2 { get; set; }
32
33         [Description("Output of operation")]
34         20 references | 0/6 passing | 0 changes | 0 authors, 0 changes
35         public ModuleOutput<double> Result { get; set; }
36
37         222 references | 0/6 passing | Taymon A. Beal, 541 days ago | 1 author, 1 change
38         public override void Execute(IExecutionContext context)
39         {
40             double param1 = this.Param1.Get(context);
41             string op = this.Operator.Get(context);
42             double param2 = this.Param2.Get(context);
43             double result = this.Operate(param1, param2, op);
44             this.Result.Set(context, result);
45         }
46     }
47 }
```



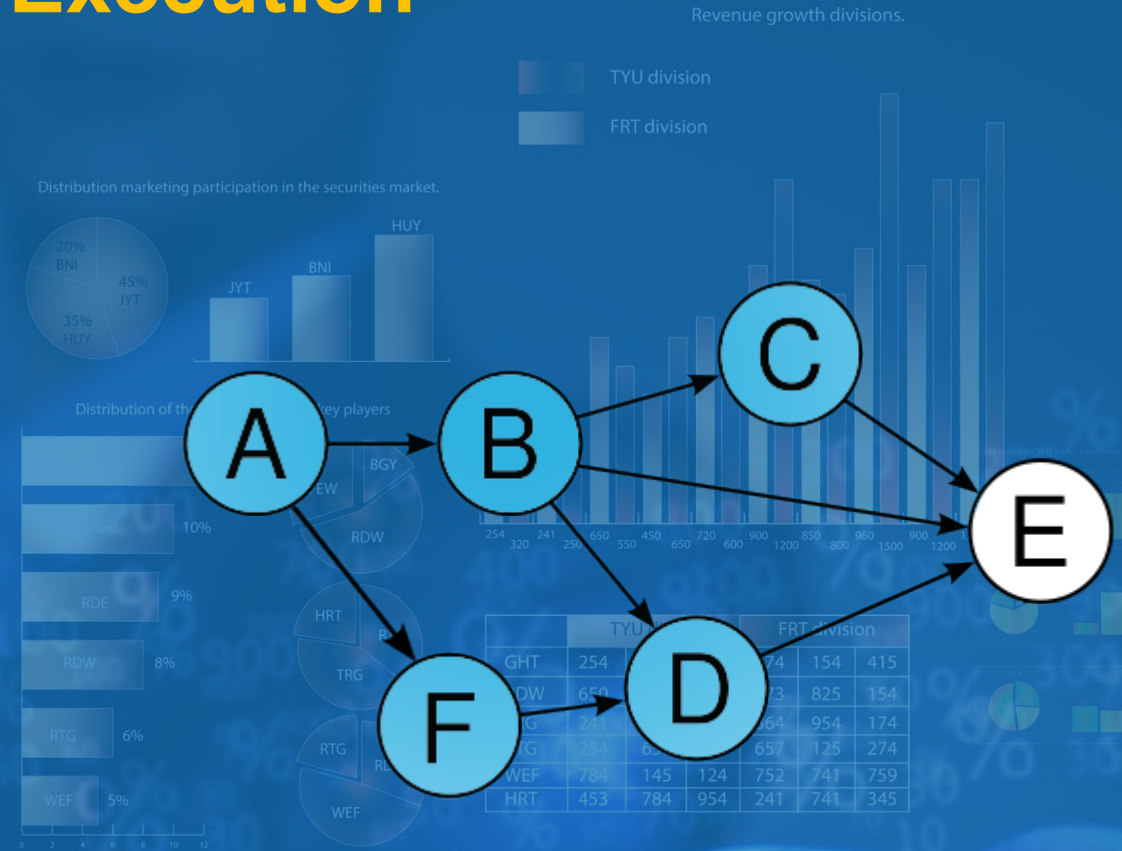
# Composable Dataflows – Execution

- Execution Strategies

- Topological
- Parallel and asynchronous
- Central vs Distributed execution
- Streams vs Staging
- Support for Looping and Branching

- Types

- Object oriented type system
- Serializable
- References for large data object (FileReference, Table)
- Streams for continuous data and large tables



# Composable QueryViews

Simplifies generation of a web-based grid-views on datasets

- **Model**
    - **Inputs**
      - Knobs – search / literal expressions
    - **Actions / Links**
    - **Child QueryViews**
  - **Recursive nature**
    - QueryViews calling other QueryViews
  - **Templating Preprocessor**
  - **Paging / sorting / filtering**
  - **Interplay across Dataflows and QueryViews**
    - Dataflows can call QueryViews— query view inputs become module inputs
    - QueryViews can be backed by Dataflows— for those data-blending scenarios
- Support for Exploration, deep dives into data, filtering / sorting / pivoting

cloud.composableanalytics.com/QueryView.aspx#/view/10564511

composable Dataflow Groups Boards Query View Form Discover Properties Admin

Enter Keywords...

### Syndromic Events

Description: Grid with data for business cases. Creating this was faster than coding.

Edit Query Save CSV XLSX Run Options

esp_event	gender	race	site	chief_complaint
=	=	=	=	LIKE

RowNum	syndromic_id	esp_event	chief_complaint	age	race	gender	site	visit_date	zip_code	is_foreign	red	id
1	672618	s_All	near syncope	3	White	F	Boston Medical Center (Adult)	2010-06-02	2152			1
2	672619	s_Falls	suicidal ideation seroquel	3	White	M	Boston Medical Center (Adult)	2010-06-02	2152			2
3	672620	s_All	motor vehicle accident	63	White	M	Boston Medical Center (Adult)	2010-06-02	2150			3
4	672621	s_All	motor vehicle accident	58	White	M	Boston Medical Center (Adult)	2010-06-02	2128			4
5	672622	s_All	medical clearance detoxification opiates	48	White	M	Boston Medical Center (Adult)	2010-06-02	1826			5
6	672623	s_ILIBroad	left arm pain	60	White	M	Boston Medical Center (Adult)	2010-06-02	80205			6
7	672624	s_All	dog bite	19	Black or African American	F	Boston Medical Center (Adult)	2010-06-03	2421			7

#### Syndromic Events Like Chief Complaint

RowNum	syndromic_id	esp_event	chief_complaint	age	race	gender	site	visit_date	zip_code	is_foreign	red	id
1	672624	s_All	dog bite	19	Black or African American	F	Boston Medical Center (Adult)	2010-06-03	2421			7

cloud.composableanalytics.com/Search.aspx



# Composable Forms

Simplifies construction of Model, Database, Web UI for CRUD operations

- **Container**
  - Fields
  - Layout
  - Hierarchical structure of Containers
    - 1-to-many, 1-to-1 relationships
- **Junction Containers**
  - Relating containers across hierarchical structure (many-to-many)
- **Field**
  - Name, Description, Type, ControlType, Constraints

The screenshot displays the 'Composable Analytics' web application interface. The browser address bar shows the URL: `localhost/CompApp/Form.aspx#/form/844897/FullApp_Hancock_Demo/1/FullApplication/1/Life/1/TAPP_P1_CONTAINER/1`. The application header includes the 'composable' logo, a navigation menu with icons, a user profile 'admin', and a search bar. The main content area is titled 'Full Application' and shows a breadcrumb trail: 'Applications > Full Application > Create a Life > Part 1'. The form is divided into two main sections: a left sidebar with a list of containers (Part 1, Occupation Information, Policy Owner, Beneficiary Information, Coverage Details, Purpose and Funding Information, Existing, Replacement, And Pending Insurance Information, Personal Information, Lifestyle Information, Juvenile Insurance, Temporary Life Insurance Agreement Application, Additional Information, Special Instructions) and a main form area. The 'Personal Information' container is selected, showing fields for First Name (Michael), Middle Name (M), Last Name (Smit), Sex (Male), Date of Birth (01/09/2017), Birth State (ID), Birth Country (United States), Social Security Number (123-14-5678), Driver's State (CA), Driver's License #, Citizenship (US), Street Address (123 Main St), City (Lexington), State (MA), Zip Code (02421), and Country (United States). There is also a checkbox for 'Check if this is a Second Life on a Survivorship application' and a 'No Drivers License' checkbox.

# Hitting the Sweet Spot with Composability

- Objects are atomic, yet high-level at the same time
- Give enough flexibility, yet Composite Objects should be easy to create
- Decomposing a problem into independent components is hard
- Decomposition can be more time consuming at first
  - but allows for tremendous amount of variation

How do you know you've succeeded?

- Minimum time to create a new composite
- Large number of varying composites developed
- Do you support a recursive nature to your language?
- Can composite objects look like atomic objects?
- Users create composites you never even thought were possible