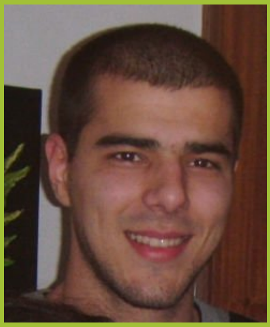
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

DEPARTAMENTO DE INFORMÁTICA

Dynamic Updates on Web Applications

SOFTWARE SYSTEMS / PLASTIC Team

CENTER FOR INFORMATICS AND INFORMATION TECHNOLOGIES



Miguel Domingues

(PhD Student)

Advised by:
João Costa Seco

Research focus on reactive dataflow programming languages and software evolution.

Objectives

Web applications are built on top of a data repository (e.g., a database) and commonly developed using an imperative operational style. The natural flow of application data between repositories and interface is hidden in the application code and its side-effects.
Data is incrementally gathered and modified until it reaches the desired content and format. This hinders the evolution of software systems and require a deep understanding of the implicit data flows by the developer to avoid making mistakes during application updates.

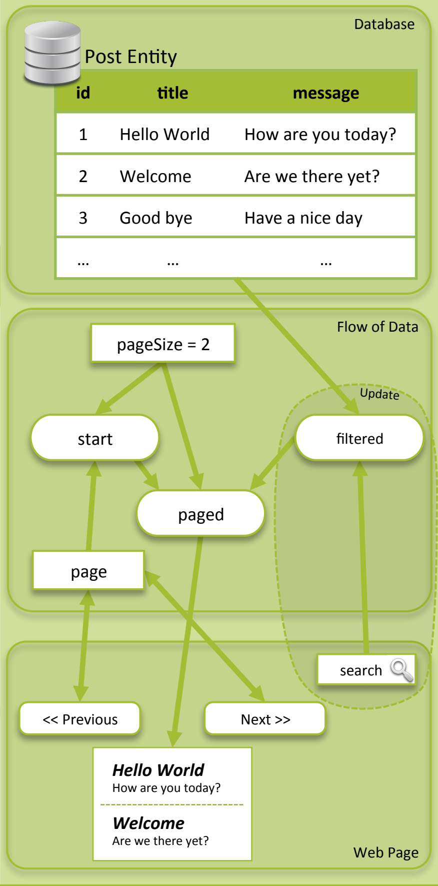
We aim to support the design of modular and reactive data-centric software systems focusing on how data flows from data sources to user interface and back.

```
def Post = entity(id, title, message)
def page = 0
def pageSize = 2
def prev = action { page ← page-1 }
def next = action { page ← page+1 }
def start = page * pageSize
def paged = { p |
    p in Post [start..start+pageSize] }
```

```
update {
  def search = input null
  def filtered = { p | p in Post ∧
          p.message contains search }
  def paged = { p|
   p in filtered [start..start+pageSize] }
}
```

Methodology

We propose a reactive dataflow based programming language capable of expressing most scenarios of data-centric systems, and is amenable to updates in a type safe way (crash free way). Flows of data are explicit in our language easing application updates.
Our reactive core language is equipped with update operators that allow seamless introduction of changes and smooth code and data evolution. Changes in data are seamlessly propagated through the application, providing a dynamic and reactive programming model. Changes in code require the application to evolve to a new state where soundness and safety is still guaranteed.
Our language is defined using formal techniques like operational semantics and type systems, amenable to mechanical verification. We aim at ensuring that updates are performed safe and seamlessly.



Post Entity

id	title	message
1	Hello World	How are you today?
2	Welcome	Are we there yet?
3	Good bye	Have a nice day
...	...	...

Expected Results

- A reactive dataflow core language for data-centric systems.
- Formal definition of the language (operational semantics and type system).
- Support for code and data evolution by means of dynamic updates.
- Static checking that well typed systems are kept structurally sound on updates.
- Compilation to operational code to be executed in the different application layers.
- Type based access control and information flow based security analysis over application data and code.
- A development tool supporting textual and visual programming, and live programming of applications.

Funding:

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

PEst-OE/EEI/UI0527/2011

INTER FACES