

Effect Handler Oriented Programming for Data Processing Applications

ALI BAŞARAN (a.basaran@student.tudelft.nl) – supervised by JARO REINDERS, CASPER POULSEN, CAS VAN DER REST

1. Background

- Effect handler oriented programming (EHOP)
 - New programming paradigm
 - Aims to provide separation of concerns
- With EHOP the user
 - Defines operations as effects
 - In one section operations are called which emit effects
 - In another section the emitted effects are handled via handlers
- EHOP needs to be analyzed and assessed against traditional paradigms to see if it beneficial to use while developing applications

```
// Define emit operation as effect
effect fun emit(msg : string) : ()

// Call emit in application logic
fun hello()
    emit("hello world!")

// Handle emit in separate handler logic
pub fun handle-emit()
    with handler
        fun emit(msg) println(msg)
    hello()
```

Example code that uses EHOP.

2. Research Question

- Main Q: How does using EHOP for implementing a data processing application affect the **modularity, readability and maintainability** of code compared to **traditional** paradigms?
- Traditional: Commonly used programming paradigms (Ex: functional programming, object oriented programming...)
- Sub Qs: How does EHOP compare against traditional paradigms in:
 - Readability
 - Development and maintenance experience
 - Runtime
 - Memory characteristics

3. Method

- Implement data processing application: **MiniExcel**, command line version of Excel which implements basic Excel functionality
- With MiniExcel, analyze EHOP on:
 - Readability** by analyzing objective measures of readability and subjective aspects of EHOP that produce readable code. Objective measures are:
 - Entropy**: Degree of complexity, disorder
 - Lines of code**: Total number of lines of code in the application
 - Halstead's Volume**: Total size of the application
 - Development experience** by taking notes during app development
 - Runtime** by measuring time it takes to execute certain commands
 - Memory characteristics** by analyzing RAM utilization
- Compare results with **functional programming** and/or **object oriented programming**
 - Application-wise comparisons are done with a similar application implemented using functional programming, referred to as **BSC**.

4. Results

4.1 Readability

- Lines of code has a positive correlation with readability, while Halstead's Volume and Entropy have a negative correlation with readability

	Lines of code	Halstead's Volume	Entropy
MiniExcel	267	12286	6.18
BSC	315	13582	6.57

Objective readability measurements of **MiniExcel** (EHOP) and **BSC** (functional programming)

- Subjectively, compared to functional programming:
 - EHOP produces more readable in-code documentation
 - EHOP, in terms of style, is closer to object oriented programming which is the most widely known programming paradigm. Therefore, for most developers, code written using EHOP will be more readable than code written using functional programming

4.2 Development and Maintenance Experience

- EHOP achieves separation of concerns which allows for seamless updates to application code
- EHOP's ability to define operations as effects without the need of implementing them allows for quick prototyping, similar to interfaces in object oriented programming
- The logic behind EHOP is significantly different than previously known paradigms which entails a learning curve
- Lack of development in the EHOP ecosystem requires developers to implement libraries that are built-in for languages that support traditional paradigms

4.3 Runtime and Memory

- After executing sets of commands on average:
 - MiniExcel** was **two times slower** than **BSC**
 - MiniExcel** utilized **less than half memory** compared to **BSC**

5. Conclusion

- Both subjectively and objectively, EHOP produces more readable code
- Development experience of EHOP had its positives and negatives. However, in the current state of the EHOP ecosystem, the lack of development makes traditional paradigms preferable.
- Functional programming code runs faster, but EHOP code utilizes less memory
- For applications that
 - expect user input EHOP is the better choice
 - aim to only execute code functional programming is the better choice
- Answer to the research question**: EHOP produces more readable, modular and maintainable code compared to functional programming.

6. Future Work

- Implement MiniExcel using a traditional paradigm from scratch rather than relying on a similar third-party application to measure more accurate results
- Extend research on EHOP over other commonly used application types such as games.