Week5 Summary

General idea of this paper:
This paper describes how to integrate interaction into a purely declarative language based on monad, which is widely used in functional programming languages. Besides, the author also compares this method with approaches including synchronous streams, continuations, linear logic and their side effects. This paper is written with examples written in Haskell.

Details about the paper:
First, the author illustrates the importance of studying monads for studying the interactive models. Monads arose in category theory and it could be applied to model a wide variety of language features. It can also serve as a basis for adding new features to a functional language.

Secondly, the author gave a tutorial of how to use monad, an abstract type to support interaction step by step. Some features of monad are introduced, including how to give a command, how to use equation reasoning, how to make commands that yield values, a combinator “>>=” (like let in Haskell), monad laws and monads and imperative programming. The imperative programming mechanism is also extended to integrate Haskell directly with C. (like ccall)

In the last part, the author told four other approaches to interaction, including synchronized streams, continuations, linear types and side effects. For stream model, it can be implemented by monads but it is very inefficient. For continuations, it is easy to define it in terms of monads, the models are equivalent in this sense. For linear states, there is no obvious way to make the converse definition of linear states in terms of monads. For side effects, it is still possible to encapsulate interaction with a monad.

In conclusion, the author said that it is very important to develop suitable methods to interact with different languages and using monads is a very efficient possible approach. More studies should be focused on this area.

My Personal Thoughts:
This paper gave me an overview of how monads can play an important role in interacting with different kinds of languages and show how to realize these detailed examples with the functional programming language Haskell. This is an intuitive method since monad can be applied in many different conditions and this could be a possible approach to realize the interaction. This paper wrote a tutorial about how to implement monads and how to realize the interaction based on monads in different languages. In some cases, using monads could be easy but in other cases it is not true. It is a very interesting topic more features of monads are still awaiting us to explore.