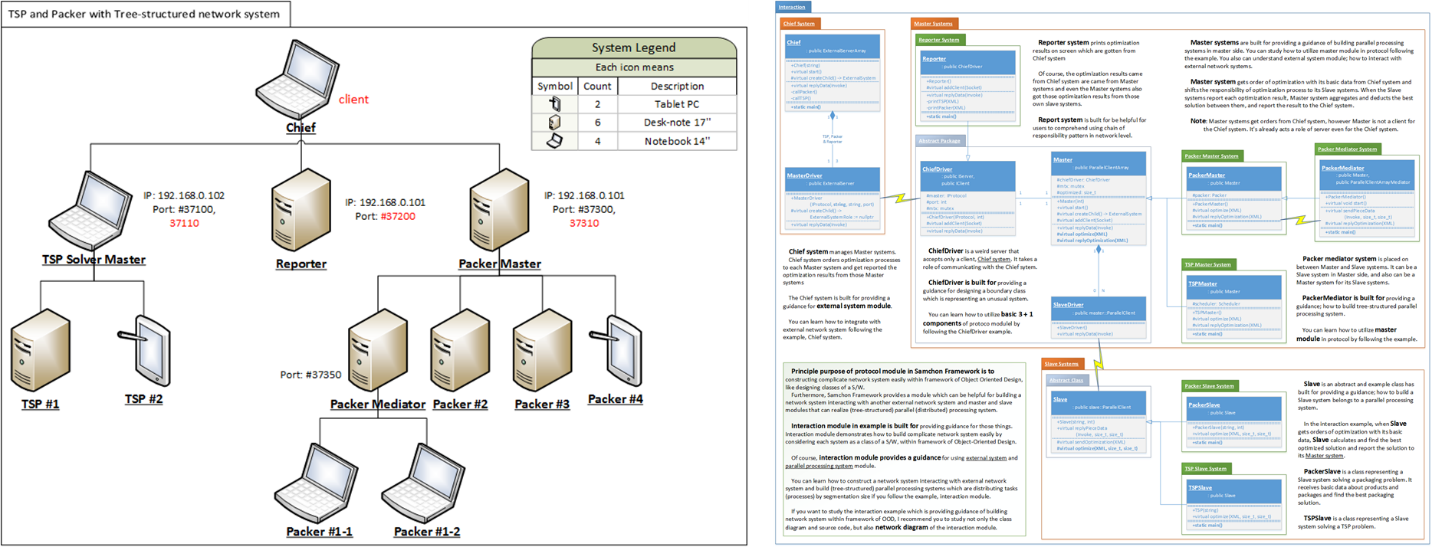
|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | |  | | **제10회 공개SW 개발자 대회 출품작 개발계획서** | |  | |

**□ 참가팀 개요**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **구 분** | | **세부내용** | | | |
| **팀 명** | | 삼촌 | | | **총 인원 ( 3 명)** |
| **팀 구 성** | | **성명** | **소속** | **부서/학과** | **직위/학년** |
| **팀 장** | | 남정호 | 히즈윌 | 연구소 | 연구원 |
| **팀 원** | | 이다빈 | LeeSet | 개발팀 | CEO |
| **팀 원** | | 이두영 | LeeSet | 개발팀 | CEO |
| **참가**  **지원** | **부문** |  | |  | |
| **분야** |  |  |  |  |
|  |  |  |  |
|  | |  | | | |
| **출품작명** | | Samchon Framework v2.0 | | | |
| **출품작 소개**  **(요약)** | | SDN (Software Defined Network) Framework wth C++ and TypeScript.  Samchon Framework는 네트워크 시스템을 S/W, 객체지향적인 관점에서 구현할 수 있는 SDN 프레임워크이다. 클라우드 및 분산처리 시스템 구현에 유용한 여러가지 모듈을 제공하고 있다.  특히 v2.0 업데이트에서는 TypeScript (JavaScript) 가 추가되어 C++ 와 JavaScript를 연동한 네트워크 시스템을 구현할 수 있게 되었다.   * GitHub: <https://github.com/samchon/stl> | | | |

1. **개별배경 및 목적**

*Samchon Framework*는 네트워크 시스템을 S/W, 객체지향적인 관점에서 구현하기 위하여 만든 SDN (Software Defined Network) 프레임워크로써, 구체적으로는 다음과 같은 목적을 위해 제작되었다.



* **네트워크 시스템을 S/W, 객체지향적인 관점에서 구현**
  + 클라우드 시스템 구현
  + 분산처리시스템 구축
  + 그 外 복잡한 구조의 네트워크 시스템을 객체적으로로 손쉽게 표현 및 구현
* **C++과 TypeScript (JavaScript) 를 연동한 통합시스템 구축**
  + **클라우드 시스템 제작**, 다음과 같은 통합이 이상적
    - C++로 서버를, TypeScript로는 UI 클라이언트를 구현
    - TypeScript with Node로 클라우드 서버를 구현
      * 분산처리시스템을 구축
        + 무겁고 연산이 많이 필요한 작업을 C++ Slave 시스템에 넘긴다
      * 물론, TypeScript Web으로 UI 클라이언트도 구현한다
  + **TypeScript-STL**
    - C++의 STL 컨테이너와 라이브러리를 TypeScript로 구현한 모듈
    - C++과의 보다 원활한 통합을 위해 만든 라이브러리. C++과 TypeScript가 기초 컨테이너, 알고리즘 레벨에서부터 공통된 인터페이스를 가져가고자 제작하였다.
    - Samchon-Collection
      * TypeScript-STL을 확장하여 만든, 컨테이너 Element I/O에 대한 이벤트를 발생시켜 이를 감지할 수 있는 모듈
* **표준화 모듈 (Protocol) 제공**
  + Entity: 엔티티 클래스 입출력에 관한 모듈
  + Invoke: Samchon Framework에서 주고받는 메시지 프로토콜에 관한 모듈
  + Basic 3 + 1 Components:
    - SDN Framework의 핵심이 되는, 네트워크 시스템을 S/W, 객체지향적인 관점에서 표현할 수 있게끔 해주는 컴포넌트 모듈
    - Samchon Framework에서 제공되는 네트워크에 관한, 클라우드 및 분산처리시스템에 관한, 일체의 모듈은 이 Basic 3 + 1 Components의 상속과 조합으로 이루어져있다.

1. **개발환경 및 개발언어**
2. C++
   * 클라우드 시스템 中 서버 역할
   * 분산처리시스템 Master 및 Slave 역할 모두 가능
3. TypeScript (JavaScript)
   * TypeScript-STL
     + C++ 시스템과의 용이한 통합을 위해, 베이스 라이브러리 레벨에서부터의 공통된 인터페이스를 가져가고자, STL (Standard Template Library) 를 TypeScript (JavaScript) 에 구현함
   * Web
     + 클라우드 시스템 中 클라이언트 역할
     + 분산처리 시스템 中 Slave client 역할
   * Node
     + 클라우드 서버 역할
     + 분산처리시스템 Master 및 Slave 역할 모두 가능
     + Node로 클라우드 서버를 만들고, 무겁고 복잡한 작업을 분산처리로 C++ Slave 에 이전하는 것이 Samchon Framework v2가 추구하는 이상적인 모형
4. ~~Flex~~
   * 클라우드 시스템 中 클라이언트 역할
   * v2.0 부터는 업데이트 중단
5. **시스템 구성 및 아키텍처**

Samchon Framework의 자랑할만한 경쟁력 중 하나는 양질의 문서 제공에 있다.

* 아키텍처
  + 클래스 다이어그램
    - C++: <http://samchon.github.io/framework/design/cpp_class_diagram.pdf>
    - TypeScript: <http://samchon.github.io/framework/design/js_class_diagram.pdf>
  + 시퀀스 다이어그램: <http://samchon.github.io/framework/design/sequence_diagram.pdf>
* API 문서
  + C++: <http://samchon.github.io/framework/api/cpp>
  + TypeScript: <http://samchon.github.io/framework/api/cpp>
* Wiki, 가이드 문서
  + Samchon Framework: 작성중
  + Library: 작성중
  + Collection: 작성중
  + TypeScript-STL: <https://github.com/samchon/stl/wiki>

1. **프로젝트 구성요소**

**내적 구성요소,** 자세히 내용은 API, 가이드 문서를 참고하거나 추후 본선에서의 설명을 들을 것

* 공통 모듈
  + library: 편의를 위한 라이브러리 모음 모듈
    - XML
    - StringUtil
    - EventDispatcher
  + protocol: 네트워크 통신에 관한 모듈
    - Basic 3 + 1 Components: SDN 을 구성하기 위한 기본 구성요소
    - Entity: 데이터 입출력에 관한 모듈
    - Invoke: Samchon Framework의 네트워크 메시지 프로토콜에 관한 모듈
  + cloud: 클라우드 서버 구현을 위한 모듈
  + master: 분산처리시스템 中 마스터 시스템
  + slave: 분산처리시스템 中 슬레이브 시스템
* C++
  + library
    - Critical Section
    - DBMS, SQL 인터페이스 및 드라이버
    - Genetic Algorithm
* TypeScript
  + Typescript-STL: TypeScript에 구현된 C++의 STL.
  + collection: Elements I/O detectable containers.

**외적 구성요소**

* GitHub Repository, 소스 코드 일체
  + Samchon Framework: <https://github.com/samchon/framework>
  + Sub-modules
    - TypeScript-STL: <https://github.com/samchon/stl>
    - Samchon-Library: <https://github.com/samchon/library>
    - Samchon-Collection: <https://github.com/samchon/collection>
  + Examples
    - 3D Bin Packing with multiple Boxes: <https://github.com/betterwaysystems/packer>
    - 추후 계속 추가될 예정
* Architecture Design: 오픈소스 참여자의 개괄적 이해를 돕기 위해 공개
* API 문서: 사용자의 미시적 이해를 돕기 위해 작성됨
* Wiki, 가이드 문서: 초보자도 쉽게 따라하며 이해하기 위해 작성되는 가이드 문서

1. **프로젝트 추진 일정**

* 6월
  + 팀 결성 및 역할 분배
* 7 ~ 8월
  + TypeScript Node 부문 구현
  + 예제 프로젝트 구현
* 9월 ~ 10월
  + 문서화. API 및 Wiki 가이드 문서 작성
* 11~12월
  + 본선진행 및 유지보수

1. **기대효과 및 활용분야**

**활용분야**

* 클라우드 및 분산처리시스템 구축
* C++과 TypeScript (JavaScript) 의 통합

**그간의 활용**

1. Samchon Simulation: 주가 시뮬레이션 및 자동거래에 관한 솔루션
2. OraQ: 클라우드 PACS 지원시스템
3. OrderCrawler: 클라우드 타사 주문기록 크롤러
4. Packer, 3D Bin Packing with multiple Boxes: Cloud 3D Bin Packing 솔루션

**앞으로의 활용**

1. Samchon Framework v3.0 with TypeScript-Flex

클라우드 시스템 구축의 주된 목적 중 하나는 웹 어플리케이션 구현에 있다. Flash player에서 구동되는 Flex를, TypeScript에 구현하여, HTML5 환경에서 구동할 수 있게끔하여 위와 같은 웹 어플리케이션 구현을 용이하게 만들 것이다.

TypeScript-Flex로 구현한 UI와 종래의 Samchon Framework를 통해 만든 클라우드 서버와의 통합을 염두로 제작하게 되는 프로젝트이다. 이와 같은 구현을 통하여 Samchon Framework를 명실상부한 하나의 개발 솔루션으로 만드는 것이 내년의 목표이다.

1. Samchon Framework v4.0 with UML

Samchon Framework는 몇 개 안 되는 컴포넌트 (Basic 3 + 1 Components) 로, 객체지향적인 관점에서 모든 종류의 네트워크 시스템을 구현할 수 있는 장점을 지녔다. 이러한 장점을 더욱 강화하기 위하여, Samchon Framework에 맞는 Network 및 Class Diagram을 그릴 수 있는 UML을 만들 것이다.

위 TypeScript-Flex와 Samchon Framework의 클라우드 모듈을 통해, 클라우드 UML을 제작, Samchon Framework의 강점을 극대화하는 게 차후년의 목표이다.

**오픈소스로써의 영역 확대**

위의 소주제들은 어디까지나 Samchon Framework에 대한 개인적인 활용 사례 및 계획에 지나지 않는다. Samchon Framework는 문서화에 많은 공을 들였다. 문서화에 공을 들인만큼 외부 이용자들이 사용하기 편할 것이라 믿고 Samchon Framework가 오픈소스로써 그 영역을 확대해나갈 수 있도록 여러 종류의 PR 활동을 해 볼 생각이다.

1. **기타 (출품작에 대한 추가설명 및 PT자료 첨부 가능)**

* <https://github.com/samchon/framework>
* <https://github.com/samchon/stl>