

ezSpec#：將嵌入式行為驅動開發工具 ezSpec 移植至 .Net 平台

專題編號：113-CSIE-S005

執行期限：112 年第 1 學期至 113 年第 1 學期

指導教授：鄭有進

專題參與人員： 110820054 鍾承皓

110820058 萬祥瑞

110820059 林于喬

一、摘要

ezSpec 是一個以 Java 語言撰寫的行為驅動開發 (BDD) 開源軟體[1]。現有的 BDD 工具（例如 Cucumber、SpecFlow 和 JBehave）要求開發人員分開撰寫功能文件與步驟實作，而 ezSpec 採用開發人員友善的嵌入式 Gherkin 語法，將功能文件與步驟實作整合到同一份檔案中，並支援測試步驟錯誤後持續執行、同步執行測試步驟以及報表的輸出[2]。本次專題之目的為將 ezSpec 完整移植到 .NET 平台，以 C# 重新開發 ezSpec#，提供 .NET 平台開發人員一個具備 ezSpec 功能的 BDD 工具。

關鍵詞：行為驅動開發 (Behavior-Driven Development, BDD)、Gherkin 語法 (Gherkin syntax)、SBE (Specification by Example)、TDD (Test-Driven Development)、.NET Framework

二、緣由與目的

ezSpec 是我們在實驗室參與開發的一個以 Java 語言撰寫的嵌入式行為驅動開發工具，其目的是解決現有 BDD 工具需要分開撰寫功能文件與步驟實作的問題[1]。透過嵌入式的設計，ezSpec 將功能文件和測試步驟整合到同一個程式碼檔，以簡化撰寫測試案例的流程，提高效率並降低學習曲線和維護成本。

由於我們希望這個工具能夠支援更多的開發者，所以選擇將 ezSpec 移植到 .NET 平台並以 C# 撰寫，其中.NET

是一個主流的開發框架，又以 C# 最常被使用。以此提高 ezSpec 的跨平台性，並為 .NET 開發者提供一個更加直觀、高效的 BDD 工具。

三、研究報告內容

(一) 研究範圍

- (1) Gherkin：一種用於 BDD 的領域特定語言，它使用自然語言的風格來描述商業邏輯和用例。
- (2) BDD (Behavior-Driven Development)：一種軟體開發方法，強調從使用者的角度思考，以驗證軟體功能是否符合需求。
- (3) SBE (Specification by Example)：一種軟體開發方法，強調利用實例來討論需求，以確保結果符合預期。

(二) 使用技術方法

- (1) Test-Driven Development (TDD)

(三) 程式架構

- (1) Domain Model (詳見圖 1)
- (2) Class Diagram (詳見圖 2)

(四) 與 ezSpec 的差異

ezSpec# 修改了部分程式架構，使其更能表達 Gherkin 中每個關鍵字的關聯。在 Domain Model 上我們修改了兩個地方，移除 Bakground 與 Scenario 的繼承關係，以及讓 Scenario Outline 繼承 Scenario[1]。（ezSpec Domain Model 詳見圖 3）

首先，在 ezSpec 中，由於 Background 與 Scenario 行為一致，所以 Background 被視作一種

Scenario[1][4]。但是我們認為，從 Gherkin 的角度出發 Background 與 Scenario 所代表的含義並不相同，因此我們認為 Background 不是一種 Scenario，從而移除 Background 與 Scenario 的繼承關係[2]。

其次，同樣從 Gherkin 的角度出發，Scenario Outline 是同一個 Scenario 執行多筆不同資料的情境，因此我們認為 Scenario Outline 是一種特殊的 Scenario，故使 Scenario Outline 繼承 Scenario[2]。

(五) 使用範例 (詳見圖 4)

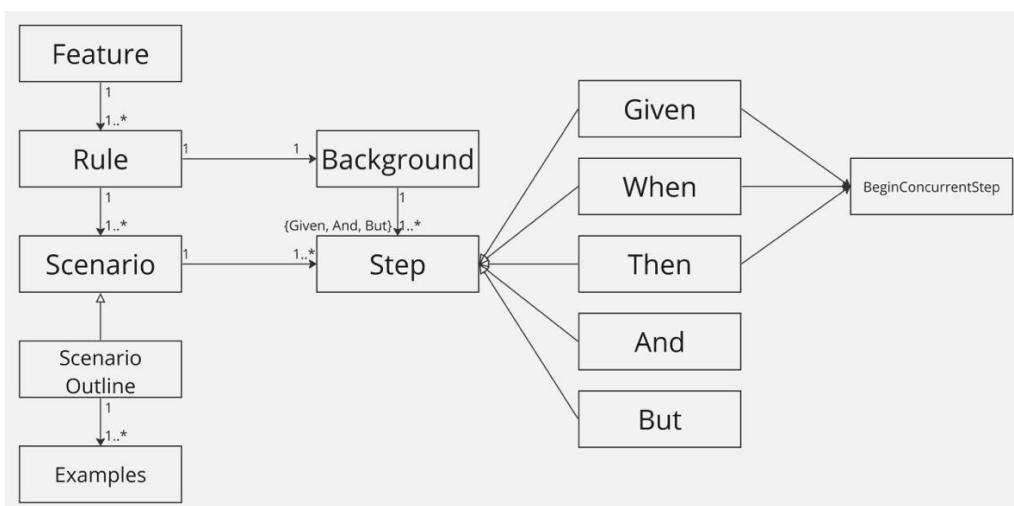
(六) 結論

這次移植的過程中，我們主要遇到了兩個問題。第一，在參考 ezSpec 的程式碼時，我們認為有一些地方是可以調整以及優化的，所以我們在撰寫 ezSpec# 的過程中，經過討論，有重新定義部分物件，並修改其之間的關係，使其更符合我們的 Domain Model。第二，由於我們對平行處理並不熟悉，因此在實作同步執行功能時花了不少時間查找資料以及除錯，最終成功實作出此功能。

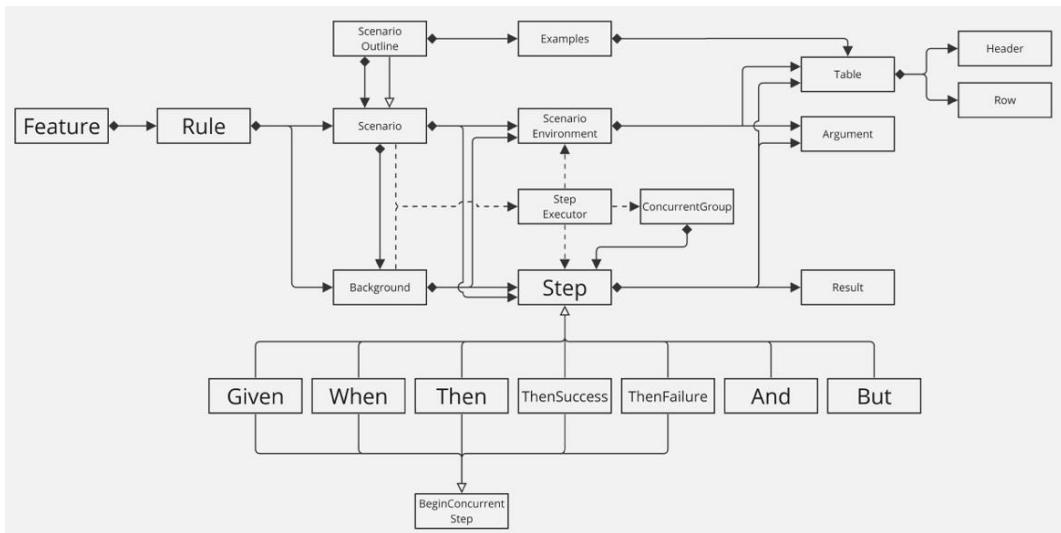
ezSpec# 功能包括使用 Gherkin 語法撰寫測試並執行，以及產生報表功能。並且已經將其套用至視窗程式設計課程專案作為使用範例。

參考文獻

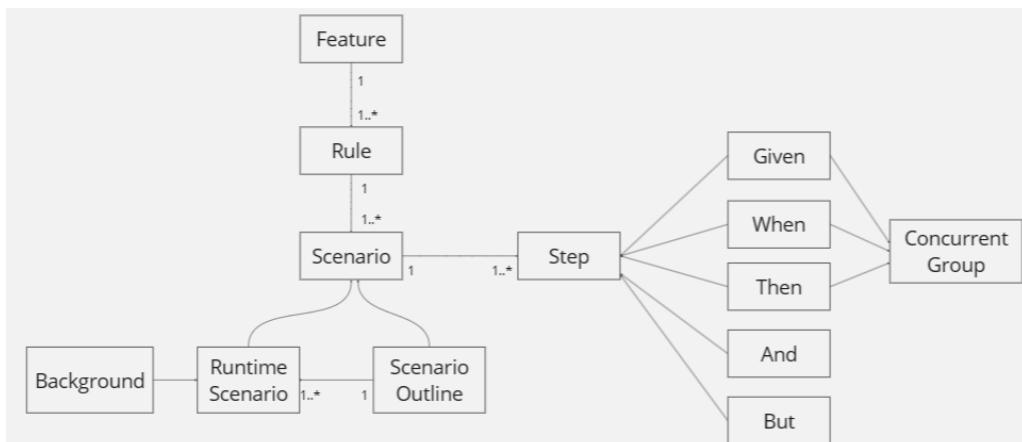
- [1] Teddy Chen, "ezspec," GitLab, [Online]. Available: <https://gitlab.com/TeddyChen/ezspec>. [Accessed: 05-May-2024].
- [2] "Gherkin Reference," Cucumber, [Online]. Available: <https://cucumber.io/docs/gherkin/reference/>. [Accessed: 05-May-2024].
- [3] Teddy Chen, "使用 ezSpec 落實行為驅動開發與實例化需求 (1)：領域模型介紹" Teddy Chen's Blog, [Online]. Available: <https://teddy-chen-tw.blogspot.com/2023/03/ezspec1.html>. [Accessed: 05-May-2024].
- [4] Teddy Chen, "使用 ezSpec 落實行為驅動開發與實例化需求 (7)：將共用步驟寫在 Background" Teddy Chen's Blog, [Online]. Available: <https://teddy-chen-tw.blogspot.com/2023/03/ezspec7background.html>. [Accessed: 05-May-2024].



(圖 1)



(圖 2)



(圖 3)

```

[TestClass]
public class PointTest {

    [TestMethod]
    public void create_point_with_single_coordinate() {
        Feature feature = Feature.New("Point");
        feature.NewRule("Create point");

        feature.WithRule("Create point")
            .NewScenario()
            .Given("value(${10.0})", env => { env.Put("value", env.GetDoubleArg(0)); })
            .When("I create a point with equal x- and y- coordinate", env => {
                Point point = new Point(env.GetDouble("value"));
                env.Put("point", point);
            })
            .ThenSuccess(env => {
                Point point = env.Get<Point>("point");
                Assert.AreEqual(10.0, point.X);
                Assert.AreEqual(10.0, point.Y);
            })
            .Execute();
    }
}
  
```

(圖 4)