CS4ZP6 Problem Statement
Ampersand Tarski Event-Condition-Action Rules

Yuriy Toporovskyy, Yash Sapra, Jaeden Guo
December 28, 2015

Ampersand Tarski is a tool used to produce functional software documents based on business process requirements. It is used by system designers, software engineers, and those involved in business management to simplify the process of designing an information system that caters to the specific needs of their company. Ampersand provides users with a dynamic model that allows them to translate real-world problems into quantifiable requirements; the requirements can be tested and used to simulate modifications to a system before they are applied.

Currently, Ampersand is readily accessible to the public through Github and it is equipped with the ability to assess logical discrepancies on sets of data based on user-specified restrictions. Logical discrepancies arise when system changes occur which violate the restrictions set forth by the user. When a system violation occurs, one of two things can happen: the change that is meant to take place is adjusted so it no longer violates the restrictions or the changes are discarded. Ampersand is used to manipulate data and generate prototypes, although there is a debugger, certain errors still slip through. Some errors remain unnoticed by the system, even though they are logical inconsistencies that prevent the system from executing the changes the user implements. These inconsistencies are persistent bugs that can distort the product that Ampersand seeks to provide. These bugs occur when the user modifies or creates new system invariants, thereby causing conflict with existing data. The system attempts to restore equilibrium by modifying the data to conform to these new restrictions. Not all sets of data can easily be adjusted to fit new restrictions created by the user, and as a result there can be data that violate the restrictions but remain in the system, distorting the outcome of modeled solutions. Data that violate the restrictions are dealt with in one of two ways: elimination or rehabilitation.

The purpose of this project is to rehabilitate existing system data while maintaining the information system according to user specifications. The job of our team involves finding a creative solution that allows Ampersand to automatically restore system invariants, rehabilitate the data that were effected by the change of restrictions, and create a program that allows Ampersand to make the most efficient choice regarding how it wishes proceed based on each individual case.

This project is important to Ampersand stakeholders (e.g. Ampersand contributors and designers) because it enhances a functional requirement and brings Ampersand closer to becoming a finished product. For other stakeholders, such as Ampersand's end users, this project means a drastic decrease in the amount of time users would normally spend searching for and correcting system inconsistencies. Moreover, this project is important to our team as it addresses a fundamental challenge that software developers often face: the ability to correctly model real world problems in an artificial system that has no natural boundaries.