Algorithmic Trading System (ATS)

Application Specification

Contents

Version:	1.1
Date:	11 March 2013

1 Introduction

1.1 Purpose of Document

This document serves as a preliminary specification of the Algorithmic Trading System (ATS). It also contains a preliminary architectural design. The full specification and design can be refined by consulting clients (course lecturer and tutor).

1.2 Background

- Students need to be familiar with the general concept of an electronic market, its participants and the way they interact with the market through transactions.
- Although an electronic market is about buying and selling anything, we will be mainly addressing the buying and selling of financial securities. Students need to have some notions of what a security is in general and in an equity (or stock) in particular since we will be focusing on equities (or stock) markets. Many of the concepts explained here can be adapted to other products.

The following terms and abbreviations will be used:

- Security: refers to the financial product being traded. In this project, the security in question is a stock with an abbreviated name. For example, “CBA” will refer to Commonwealth Bank shares.
- Bid order: represents the intention of a buyer and will contain the security involved, the name of the buyer, the quantity of shares and the price.
- Ask order: represents the intention of a seller and will contain the security involved, the name of the seller, the quantity of shares and the price.
- Orderbook: represents the list of unexecuted (i.e. pending) orders in the market.
- Bid List: represents the list of unexecuted (i.e. pending) bid orders in the market.
- Ask List: represents the list of unexecuted (i.e. pending) ask orders in the market.
- Spread: refers to the difference between the highest bid price and the lowest ask price
- Trade: represents a match between a buyer and a seller (or that a bid order is executed against an ask order). A trade has the names of the buyer and the seller, the security involved, the quantity,

The term “trading data” is used to describe three fundamental transactions types.

- Order processing: correspond to requests from the market participants (i.e. brokers) regarding the buying and selling of securities. Such transactions can correspond to submitting an order (i.e. bid or ask order), amending existing orders, or deleting existing orders. Processing orders is the main focus of this document.
- Trades: such transactions are typically generated by the trading engine once some orders have been matched with each other. They indicate details such as the volume of the trade and the price.
- Market events: represent particular events such as the opening/closing of the market and moving between different market phases.

2 Application Overview

A prototype ATS is required to be developed initially as a number of standalone modules in a programming language of your choice. Teams have the choice of running their system on two different platforms:

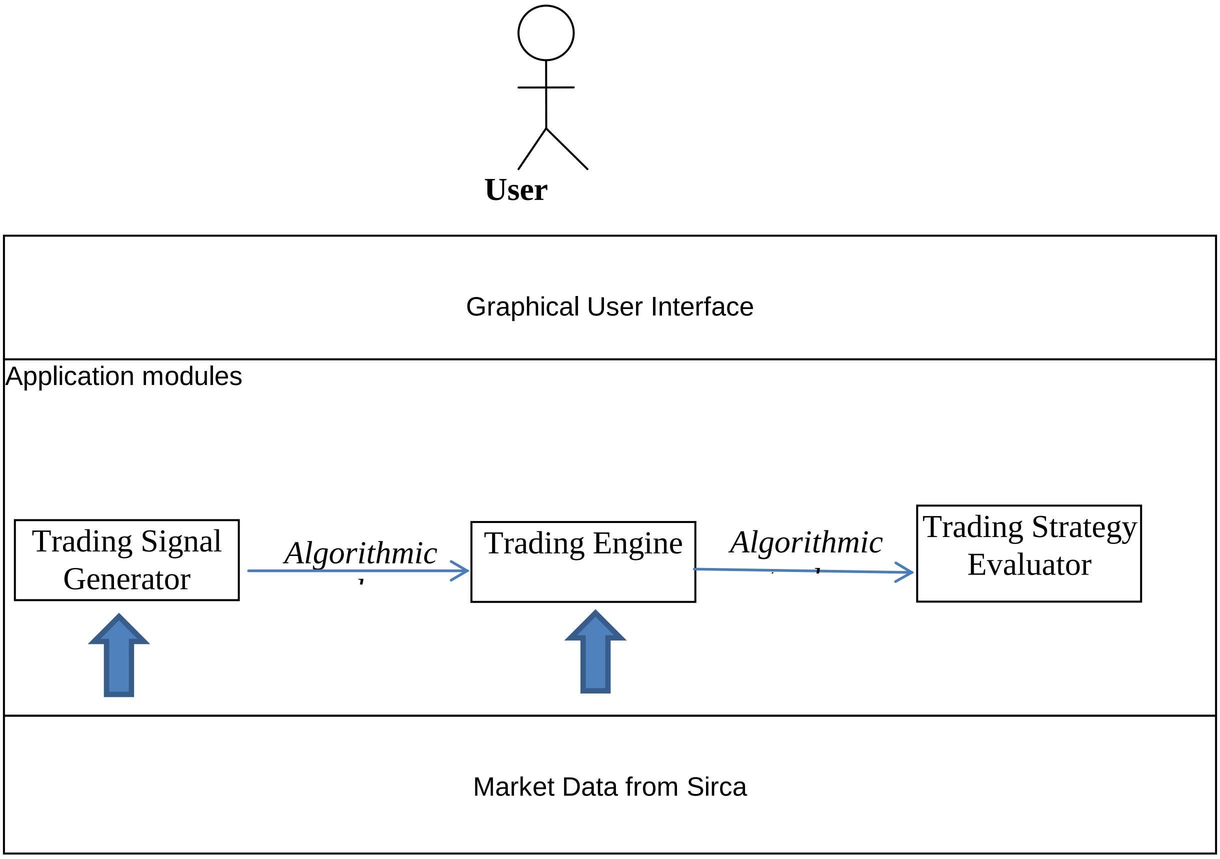
- PC running Windows
- Unix/Linux platform

In the first stage, the ATAS preliminary architecture will consist of the following modules:

- A *trading signal generator* reads market data and generates *trading signals*. The trading signals are essentially algorithmic orders that can be routed into the trading engine, which will produce an algorithmic trade.
- A *trading engine* implements the market rules and generates trades from orders which are submitted to the trading engine through some kind of communication network.
- *A trading strategy evaluator* will look at the list of algorithmic trades and give some sort of payoff (profit or loss) from using the trading strategy.

See figure below for details. The market data is the one provided by SIRCA and accessible via the Sirca portal. Trading signals are generated based on rules that produce actions based on market data. The trading signals are then recorded as orders and converted into algorithmic trades by the trading engine. A evaluation module is responsible for processing the algorithmic trades to produce results of the trading strategy.

In the second stage, these modules will be integrated within a user interface. Such a user interface can supported as a standalone or Web application.



A list of preliminary requirements is given in Section 3. Every team will be responsible for implementing as much requirements as possible within the time constraints.

3 Requirements list

3.1 Functional requirements

Id	Functional requirement	Comments
1	Reading a correctly formatted Sirca orders file (1 day only)	See “Introduction to orderbooks” document
2	Choosing an appropriate algorithmic trading strategy and setting its different parameters	List of algorithmic strategies will be provided as work progresses.
3	Generating algorithmic orders for 1 particular day	
4	Generating algorithmic trades for 1 particular day	List of techniques for generating algorithmic orders will be provided as work progresses.
5	Evaluating algorithmic trades and providing feedback to user	List of techniques for evaluating algorithmic trades will be provided as work progresses
6	Generating a strategy performance report	
7	GUI functions to control use cases (1-6) to load and execute an orders file	
8	GUI functions to visualise market data (spread, volume and depth)	

3.2 Quality requirements

Id	Quality requirement	Comments
1	Speed of execution (transactions per second)	
2	Usability of the GUI	
3	Quality of the visualisation	
4	Quality of strategy performance report	