jExchange Version: 1.3

Progetto di Laboratorio di Sistemi Operativi A.A. 2015-2016

Saverio Giallorenzo, Università di Bologna

---

1. DESCRIZIONE

jExchange è un sistema distribuito di simulazione di un mercato finanziario (chiamato anche Borsa valori o solo Borsa) la cui architettura è composta da 3 entità: i Players, il Market e gli Stocks.

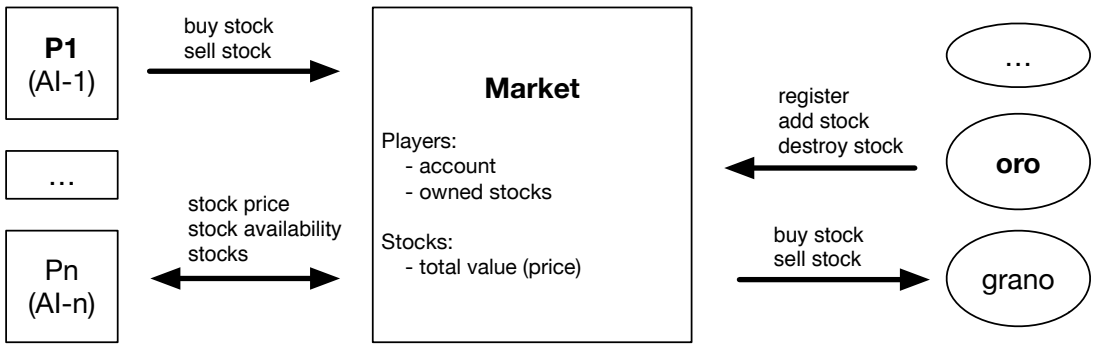

Fig. 1. Rappresentazione del sistema jExchange

1.1 Market

Il Market rappresenta il mercato. Il Market può:

registrare un nuovo Player. Quando un nuovo Player si registra, tramite il proprio nome, gli viene assegnato un Account contenente una quantità iniziale di denaro di 100 (euro). I nuovi Players non possiedono nessuna unità di Stock;

registrare un nuovo Stock. Quando un nuovo Stock si registra presso il Market, lo Stock diventa disponibile (visibile) ai Players per essere comprato e venduto;

dare informazioni sugli Stocks ai Players. I Players possono chiedere al Market:
(1) l'insieme degli Stocks disponibili nel mercato, i.e., l'insieme dei nomi degli Stocks registrati;
(2) il prezzo di uno Stock, identificato dal suo nome;
(3) la quantità di uno Stock[1], identificato dal suo nome.

confermare o negare l'acquisto di Stock. Il Market conferma o rigetta l'acquisto, da parte di un Player, di 1 unità di Stock. L'unità di Stock può essere acquistata se 1) è disponibile[1] e 2) se il Player che lo acquista ha denaro a sufficienza per pagarla. In caso positivo, in sequenza e atomicamente:
(1) l'unità di Stock disponibile viene decrementata di 1;

[1] il Market chiede questo dato allo Stock relativo.

(2) si aumenta di 1 la quantità di Stock posseduto dal Player acquirente;
(3) si decrementa l'Account del Player dell'attuale prezzo di un'unità di Stock[2]
(4) si aumenta il prezzo totale di quello Stock in maniera corrispondente a quanto riportato in § 3.

confermare o negare la vendita di Stock. Il Market conferma o rigetta la vendita, da parte di un Player, di un'unità di Stock. L'unità di Stock può essere venduta solo se è effettivamente posseduta dal Player. In caso positivo, in sequenza e atomicamente:
(1) l'unità di Stock disponibile viene incrementata di 1;
(2) si decrementa di 1 la quantità di Stock posseduto dal Player venditore;
(3) si incrementa l'Account del Player dell'attuale prezzo di un'unità di Stock[2];
(4) si diminuisce il prezzo totale dello Stock in maniera corrispondente a quanto riportato in § 3.

aggiornare il prezzo degli Stock. Quando uno Stock produce o deperisce delle proprie unità, il mercato aggiorna i prezzi coerentemente a § 1.3.

1.2 Players

I Players sono i giocatori della Borsa. Comprano e vendono autonomamente (vedere § 4) unità di Stocks speculando sul loro valore. Un Player può:

comprare. Il Player chiede al Market di comprare 1 unità di Stock;

vendere. Il Player chiede al Market di vendere 1 unità di Stock;

chiedere informazioni al Market. I Players possono chiedere le informazioni al punto "*dare informazioni sugli Stocks ai Players*" di § 1.1.

1.3 Stocks

Gli Stocks rappresentano i produttori di risorse nel mercato. Uno Stock può:

registrarsi presso il Market. Gli Stocks si registrano presso il Market col proprio nome (e.g., "Oro", "Grano", "Petrolio") e il proprio valore totale iniziale.

produrre nuove unità di Stock. A intervalli regolari, uno Stock può produrre quantità variabili di nuove unità di prodotto. Ad esempio, lo Stock "Grano" può produrre da 1 a 5 unità ogni 1 minuto. Quando uno Stock produce nuove unità, comunica al Market la quantità percentuale aggiunta sul totale precedentemente disponibile. Ad esempio, se prima della produzione c'erano 20 unità di Grano e ne vengono prodotte 2, lo Stock di Grano comunicherà al Market il dato `0.1` (corrispondente a $2/20$). Dato che è aumentata l'offerta del Grano, il Market diminuisce il prezzo totale del Grano del 10%.

far deperire unità di Stock. A intervalli regolari, una parte dello Stock può (o no, si pensi al caso dell'Oro) deperire. Ad esempio, ogni 5 minuti da 1 a 3 unità di "Grano" possono deperire. Quando uno Stock fa deperire delle unità, comunica al Market la quantità percentuale rimossa sul totale precedentemente disponibile. Ad esempio, se prima c'erano 20 unità di Grano e ne deperiscono 3, lo Stock di Grano comunicherà al Market il dato `0.15` (corrispondente a $3/20$). Dato che è diminuita l'offerta del Grano, il Market aumenta il prezzo totale del Grano del 15%.

Note ed Eccezioni

- per semplicità, in jExchange le unità di Stock possono deperire solamente se presenti nei relativi Stock. Quindi le unità possedute dai Players non deperiscono;
- **Update**: quando uno Stock arriva a **0** in seguito ad un **Deperimento**, il prezzo del bene rimane invariato, e.g., il Grano ha 1 unità con costo 40, viene Deperito di 3, la disponibilità scende a 0 ma il prezzo rimane 40.

[2] Equivalente a $valore\ totale\ Stock / totale\ unita\ di\ Stock$.

- ◇ **Update**: quando uno Stock è a **0** e c'è una **Produzione**, il prezzo del bene rimane invariato, e.g., il Grano ha 0 unità e prezzo registrato 40, ne vengono prodotte 3 unità dal relativo Stock, il Market aggiorna le unità di Grano disponibili ma il prezzo rimane 40.
- ◇ gli Stocks non possono deperire al di sotto dello 0;
- ◇ **Update** il prezzo di uno Stock non può mai scendere sotto il valore di 10.
- ◇ può esistere una sola istanza di un determinato Stock, e.g., non esistono 2 Stock chiamati "Grano".

2. OUTPUT OSSERVABILI

Dato che jExchange non ha interazioni con utenti, sia ai fini implementativi che per la valutazione finale, è bene dotare tutti i servizi coinvolti di appropriati output per seguire l'evoluzione del sistema.

Di seguito è riportato un esempio di output dei servizi: **Market** che implementa Market, **Player1** e **Player2** che implementano Player e **Oro** e **Grano** che implementano Stock.

L'output viene stampato ad ogni operazione effettuata dove → indica una vendita, ← un acquisto, ↑ una produzione e ↓ un deperimento. Acc è l'Account di un Player, + o - indicano la modifica di un valore.

Market

```
Player1 → Grano | Grano:    124.12 -
                | Oro:      210
                | Petrolio: 143.1
-----------------------------------
Player2 ← Oro   | Grano:    124.12
                | Oro:      212.12 +
                | Petrolio: 143.1
-----------------------------------
Grano ↓ 4.7%    | Grano:    129.95 +
                | Oro:      212.12
                | Petrolio: 143.1
-----------------------------------
Oro ↑ 1.4%      | Grano:    129,95
                | Oro:      209.15 -
                | Petrolio: 143.1
```

Player1

```
← Grano | Acc: 123 | Grano: 21 -
                   | Oro: 3
```

Player2

```
→ Oro | Acc: 72 | Oro: 4 +
```

Oro

```
←     | Available: 7 -
-------------------------------
↑ 1.4% | Available: 8 +
```

Grano

```
→     | Available: 89 +
-------------------------------
↓ 4.7% | Available: 85 -
```

3. PRICING

Il Market di jExchange cerca di frenare le speculazioni finanziarie, ad esempio, la creazione di bolle speculative dovuto all'acquisto ad alta frequenza di uno Stock, che ne fa lievitare il valore.

Per questo, il Market valuta il nuovo valore totale di uno Stock in funzione della frequenza di acquisto o vendita delle unità di Stock: se vengono fatti tanti acquisti o vendite entro poco tempo, e.g., nel giro di 1 secondi da un operazione all'altra, il Market aumenta o diminuisce il valore del bene di poco. Al contrario, se le operazioni sono distanziate nel tempo, e.g., oltre i 2 secondi, lo sconto o l'aumento del valore totale dei restanti beni è maggiore.

Nella tabella sotto, sono riportati gli intervalli temporali e i relativi *moltiplicatori temporali*.

Distanza tra operazioni successive	**moltiplicatore temporale**
0 – 1 secondi	0,0001
1 – 2 secondi	0,001
2+ secondi	0,01

Il differenziale per calcolare i nuovi prezzi è rappresentato da Δ, definito

$$\Delta = valore\ attuale \times moltiplicatore\ temporale$$

Quindi, in caso di vendita la formula del nuovo valore è

$$Nuovo\ Valore\ dopo\ Vendita = Valore\ Attuale - \Delta$$

analogamente, per l'acquisto, la formula del nuovo valore totale è

$$Nuovo\ Valore\ dopo\ Acquisto = Valore\ Attuale + \Delta$$

Facendo un esempio, supponiamo che non ci siano state operazione sullo Stock del Grano precedenti ai 2 secondi e che ci siano 20 unità di Grano con valore totale di 100 (euro). Un Player acquista 1 unità di Grano. Il nuovo prezzo totale del Grano viene calcolato come: $100 + 100 \times 0,01 = 101$. Ora supponiamo che entro 1 secondo ci sia un acquisto di un'altra unità di Grano: ora il nuovo valore calcolato è di $101 + 101 \times 0,0001 = 101,0101$.

4. INTELLIGENZA DEI PLAYERS

I Players di jExchange simulano degli speculatori di borsa che vogliono trarre il maggior profitto possibile dal gioco. In pratica, voglio massimizzare la quantità di denaro presente nel proprio Account.

Per farlo, ogni Player è dotato di una strategia automatica che cerca le risorse disponibili, controlla il loro prezzo e valuta cosa comprare e vendere.

Ai fini progettuali, **non** è richiesto lo sviluppo di più strategie (e.g., una conservativa, una aggressiva, etc.). Basta avere 1 tipo di Player che implementa una strategia. Il voto del progetto non valuterà "l'intelligenza" della strategia, ma solamente la sua implementazione e.g., se da adito a comportamenti imprevisti a causa di una cattiva gestione della concorrenza.

In particolare è bene riportare nel report di progetto (cf. § 6.2) le discussioni e le scelte sull'implementazione della strategia (o delle strategie, se ne sono state implementate più di una) del proprio Player, con particolare enfasi sull'eventuale (se implementato) sfruttamento del parallelismo e.g., per esplorare il mercato e cercare contemporaneamente soluzioni alternative.

5. STOCK DI DEFAULT

Di seguito, gli Stock di default con le relative quantità e valori iniziali e i tassi di deperimento e produzione.

Grano	
Quantità Iniziale	100
Valore Tot. Iniziale	100
Deperimento	1–5 ogni 5s
Produzione	3–6 ogni 3s

Oro	
Quantità Iniziale	5
Valore Tot. Iniziale	25
Deperimento	-
Produzione	0–2 ogni 10s

Petrolio	
Quantità Iniziale	50
Valore Tot. Iniziale	75
Deperimento	1–2 ogni 8s
Produzione	1–3 ogni 10s

Per la consegna di progetto basta l'implementazione di questi tre tipi di Stock, ma ogni gruppo può pensarne e definirne di nuovi nel proprio progetto.

6. CONSEGNA DEL PROGETTO

6.1 Implementazione

Il progetto deve essere sviluppato utilizzando il linguaggio Jolie.

Non ci sono requisiti riguardo ai protocolli (*protocols*) e i media (*locations*) utilizzati per realizzare la comunicazione tra Players, Market, e Stocks.

L'obiettivo dell'implementazione riguarda la gestione concorrente di risorse condivise. L'importante è dimostrare che il comportamento implementato gestisca i tipici problemi di concorrenza su risorse condivise (e.g., il tipico problema Readers-Writers).

6.1.1 *Reading-Writing.* Uno degli argomenti di discussione per lo sviluppo del progetto è la gestione delle risorse come Account, Stocks, etc. Il problema è riconducibile al Reader-Writer. È importante pensare a come sviluppare queste funzionalità tra Players e Stocks rispetto al Market che, eventualmente, tentano di produrre, acquistare, vendere o distruggere (deperire) contemporaneamente le unità di uno stesso Stock. Ai fini della valutazione è bene riportare le scelte fatte per implementare questi meccanismi.

N.B. Uno dei punti di valutazione riguarda l'uso del parallelismo nel progetto. E.g., usare `execution { sequential }` per prevenire scritture (e letture) parallele è una soluzione al problema considerato, ma rappresenta anche una notevole perdita in termini di concorrenza, incidendo negativamente sulla valutazione del progetto. Analogamente, il costrutto `synchronized( t ){ ... }` deve essere usato con raziocinio per massimizzare il grado di parallelismo del proprio progetto.

6.2 Report

Circa 8–10 pagine – singola colonna, font-size 12 – in formato PDF, dove vengono discusse:

- ◇ la struttura del progetto (la divisione delle funzionalità tra i servizi e la loro gerarchia);
- ◇ le scelte più importanti fatte sul progetto (e.g., qual'è la logica di gestione di aggiunta di Stock, deperimento, acquisto e vendita) e come sono state sviluppate le principali funzionalità (anche con snippets di codice);
- ◇ i problemi principali riscontrati, le alternative considerate e le soluzioni scelte;
- ◇ le istruzioni per eseguire il progetto (in generale) e quelle per eseguire una demo di esecuzione. La demo deve contenere almeno 2 Players, il Market e i tre Stocks di § 5. Il Market parte, gli Stocks si registrano e infine i Players si registrano e iniziano a comprare e vendere.

A questo indirizzo è disponibile un canovaccio, in markdown, con la struttura del report. (Pandoc e altri tools permettono di creare PDF da markdown). È possibile scrivere il report nel formato preferito (.doc, .tex), l'importante è che il PDF generato rispetti la struttura del succitato template.

Il report di progetto è molto importante e buoni progetti con report scadenti possono risultare insufficienti. Il report non deve essere la ripetizione delle specifiche, deve contenere le idee, le discussioni e le scelte progettuali, soprattutto per quelli considerati punti chiave.

Mantenere una documentazione completa e in linea con lo sviluppo è di estrema importanza per costituire uno storico delle scelte fatte, utile anche come knowledge-base per nuovi progetti (e.g., riuso di soluzioni, oltre che di codice).

È buona norma scrivere una bozza del report *durante* lo sviluppo del progetto, documentando le discussioni e le soluzioni adottate.

6.3 Gruppi

I gruppi possono essere costituiti da un minimo di 3 a un massimo di 5 persone.

I gruppi che intendono svolgere questo progetto, devono comunicare via email a

saverio.giallorenzo@gmail.com

entro l'**11 Maggio 2016** la composizione del gruppo. L'email deve avere come oggetto "GRUPPO LSO" e contenere il **nome del gruppo** e una riga per ogni componente del gruppo, con "cognome,

nome, matricola". Inoltre deve essere presente un **indirizzo email di riferimento** a cui mandare le notifiche al gruppo.

Inoltre, ogni componente di un gruppo deve iscriversi sul tool all'indirizzo

http://esami.int.nws.cs.unibo.it/cgi-bin/Main.php

inserendo l'appartenenza (il nome) al proprio gruppo di progetto.

L'appello è unico e indipendente dalla date di consegna del progetto e della discussione.

Chi non comunicherò la composizione del gruppo entro l'11 Maggio non potrà consegnare il progetto.

Nel caso, **chi non riuscisse a trovare un gruppo**, lo comunichi il prima possibile, entro e non oltre il **9 Maggio 2016**, a saverio.giallorenzo@gmail.com, con una mail con oggetto "CERCO GRUPPO LSO", specificando i propri dati (nome-cognome-matricola-email) ed eventuali preferenze legate a luogo e tempi di lavoro. Si cercherà di costituire gruppi di persone con luoghi e tempi di lavoro compatibili. Le persone senza un gruppo verranno raggruppate il prima possibile (e non sarà possibile modificare i gruppi formati).

6.4 Consegna e Date

Per la consegna si deve inviare una email con soggetto "CONSEGNA LSO - NOME GRUPPO" (dove NOME GRUPPO è il nome del registato del gruppo) a saverio.giallorenzo@gmail.com.

In allegato alla mail dovrà essere incluso uno archivio zip nominato `NOME_GRUPPO_LSO.zip`. L'archivio deve contenere il progetto, sotto la directory "project", e la relazione, nominata `REPORT_LSO.pdf`.

Il report va anche consegnato in forma cartacea nella casella del prof. Sangiorgi (piano terra del dipartimento di informatica, a fianco dell'ufficio).

È possibile inserire un README in "project" che riporti come lanciare gli eseguibili del progetto.

Ci sono due date di consegna:

- 23:59 di Mercoledì 1 Luglio 2016, con discussione entro lo stesso mese;
- 23:59 di Lunedì 21 Settembre 2016, con discussione entro lo stesso mese o quello successivo.

In seguito alle consegne, verranno fissate data e ora della discussione del progetto, cercando di rispettare le tempistiche sopraccitate, compatibilmente coi tempi di correzione (FCFS). La data di discussione verrà notificata ai gruppi tramite la mail di riferimento.

6.4.1 *Valutazione del progetto.*

Il progetto deve rispettare la consegna descritta nelle Sezioni 1–5.

Il progetto verrà valutato mediante una discussione di gruppo, con tutti i componenti del gruppo presenti. Non è possibile per i componenti di un gruppo effettuare la discussione in incontri separati. Al termine della discussione, ad ogni singolo componente verrà assegnato un voto in base all'effettivo contributo dimostrato nel lavoro di progetto. La valutazione del progetto è indipendente dal numero di persone che compongono il gruppo.

6.4.2 *Note Importanti*

- non si accettano email o richieste di eccezioni sui progetti con motivazioni legate a esigenze di laurearsi o di non voler pagare le tasse per un altro anno.
- chi copia o fa copiare, anche un sola parte del progetto, si vedrà invalidare completamente il progetto senza possibilità di appello. Dovrà quindi rifare un nuovo progetto l'anno successivo.

6.5 Domande sul progetto e ricevimento

Per le domande sul progetto si consiglia di usare il newsgroup del corso. Risposte corrette alle domande dei colleghi sul newsgroup verranno valutate positivamente in sede di esame. È possibile

chiedere informazioni o un ricevimento via email, specificando la motivazione della richiesta, a saverio.giallorenzo@gmail.com.