

**Ανάπτυξη Εργαλείου για τη Μοντελοποίηση, Προσομοίωση και
Αναπαράσταση Δικτύων Αξίας**

Παπαθανασίου Ευάγγελος
erapath@csd.uoc.gr

ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ, ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

Περίληψη

Πολύπλοκα συστήματα υπηρεσιών έχουν αποτελέσει τη βάση για να αναπτυχθούν δίκτυα αξίας τα τελευταία χρόνια. Στα πλαίσια της επιτυχημένης συνεργασίας των οργανισμών που απαρτίζουν αυτά τα δίκτυα έχει δημιουργηθεί η ευκαιρία να μελετηθούν και να αναλυθούν μέσω οικονομικών και τεχνολογικών μέσων. Στην παρούσα εργασία παρουσιάζεται ένα εργαλείο το οποίο απεικονίζει, μοντελοποιεί και προσομοιώνει δίκτυα αξίας.

1. Εισαγωγή

Ζούμε σε μια εποχή μεταβολών στην οποία η ανάπτυξη νέων τεχνολογικών εργαλείων και η αποτελεσματική διαχείριση μεγάλου όγκου πληροφορίας αποτελούν τη βάση για την ανάδειξη και εδραίωση πολύπλοκων συστημάτων υπηρεσιών. Για τη δημιουργία και υλοποίηση υπηρεσιών έχουν μελετηθεί δίκτυα αξίας τα οποία αποτελούνται από σύνολα κοινωνικών και τεχνικών πόρων. Οι οργανισμοί που συμμετέχουν στα δίκτυα αξιών αλληλεπιδρούν μεταξύ τους και παράγουν οικονομική αξία [1,2,3,4].

Σκοπός της εργασίας είναι ο σχεδιασμός και η δημιουργία του εργαλείου δικτύων αξίας (Value Network Tool) το οποίο απεικονίζει, μοντελοποιεί, προσομοιώνει και αναπαριστά δίκτυα αξίας. Επιπλέον έχει δημιουργηθεί ένας τρόπος απεικόνισης και διαχείρισης των δεδομένων που χρησιμοποιεί το εργαλείο. Ο σχεδιασμός και η δημιουργία του εργαλείου βασίστηκε στην μελέτη που εκπονήθηκε από ερευνητές της IBM σε συνεργασία με το εργαστήριο υπηρεσιών μετασχηματισμού (Transformation Services Laboratory) του Πανεπιστημίου της Κρήτης [1].

Η εργασία αυτή αποτελείται από δύο μέρη. Το πρώτο μέρος αφορά σε διεργασίες που θα εκτελούνται για την διαχείριση των δεδομένων του εργαλείου. Για το σκοπό αυτό υλοποιήθηκε ο πυρήνας (core) του εργαλείου στον οποίο πραγματοποιούνται όλες οι κύριες διεργασίες. Το δεύτερο μέρος αφορά στον τρόπο απεικόνισης, αποθήκευσης και ανάκτησης των δεδομένων των δικτύων αξίας μέσω ενός XML σχήματος.

Ας σημειωθεί ότι η δημιουργία του πυρήνα αύξησε τη λειτουργικότητα του εργαλείου καθώς δόθηκε η δυνατότητα υπολογισμού των οικονομικών σχέσεων ενός δικτύου αξίας, της πλήρους απεικόνισης σε xml ενός στιγμιότυπου καθώς και της σύνδεσης του δικτύου με το Websphere Business Modeler της IBM. Ένα σημαντικό χαρακτηριστικό του πυρήνα είναι η επεκτασιμότητα καθώς δίνεται η δυνατότητα να προστεθούν νέες διεργασίες και λειτουργίες

2. Τεχνολογίες υλοποίησης

Η ανάπτυξη του εργαλείου έγινε με την γλώσσα προγραμματισμού Java και συγκεκριμένα με την έκδοση 1.5. Η πλατφόρμα υλοποίησης που δημιουργήθηκε το εργαλείο VNT ήταν το NetBeans 6.0. Επίσης η δημιουργία του XML σχήματος έγινε με την βοήθεια του εργαλείου Stylus Studio 2006 [6].

Οι κυριότερες βιβλιοθήκες (API) που χρησιμοποιήθηκαν ήταν το JGraph [7-12], το οποίο είναι υπεύθυνο για το γραφικό interface του εργαλείου, και τα Xml-apis [5], Resolver, XercesSamples, XercesImpl, Serializer για την δημιουργία των XML αρχείων και την επεξεργασία τους.

Τα πλεονεκτήματα της συγκεκριμένης υλοποίησης είναι τα παρακάτω:

- Λειτουργία ανεξάρτητα από την πλατφόρμα και το λογισμικό.
- Ευκολία υλοποίησης, διαχείρισης , ελέγχου και απεσφαλμάτωσης.
- Δυνατότητα αναβάθμισης και επέκτασης.
- Συνδεσιμότητα και συνεργασία με άλλες εφαρμογές.

3. Υλοποίηση

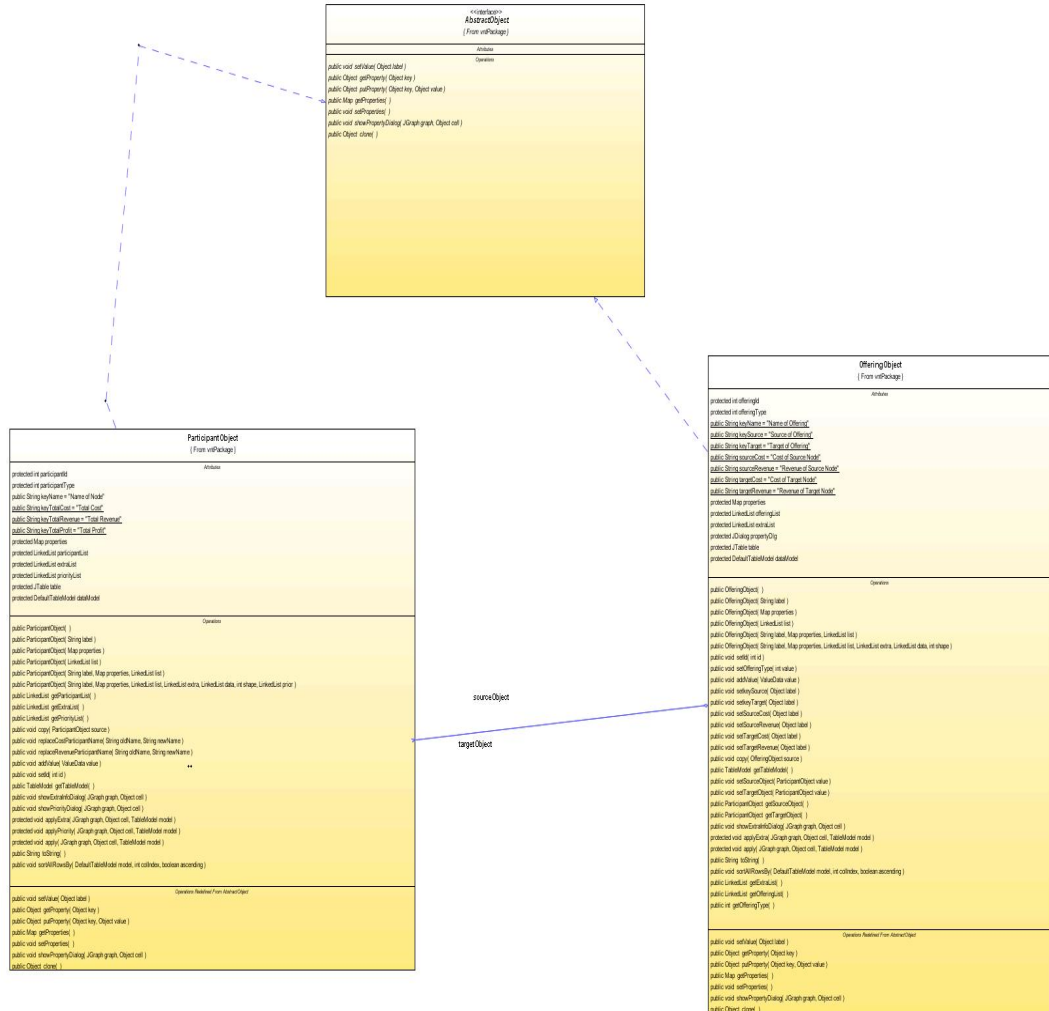
Στην παράγραφο αυτή θα περιγραφεί η σχεδίαση και η υλοποίηση όλων των κλάσεων που περιλαμβάνει ο πυρήνας και των κλάσεων οι οποίες υλοποιούν την δημιουργία ενός XML αρχείου, το μετασχηματισμό του σε HTML για την περαιτέρω κατανόηση του δικτύου, την φόρτωση του XML αρχείου στο εργαλείο και άλλες βοηθητικές διεργασίες.

3.1.Περιγραφή υλοποίησης κλάσεων δεδομένων

Ο σκοπός της υλοποίησης τέτοιων κλάσεων είναι να γίνει απεικόνιση των δεδομένων που βρίσκονται πίσω από το γραφικό περιβάλλον (GUI) του εργαλείου [13]. Οι κόμβοι-συμμετέχοντες και οι συνδέσεις-σχέσεις μεταξύ των κόμβων αποτελούν τα δεδομένα που περιγράφουν και απεικονίζουν το δίκτυο αξίας. Έτσι δημιουργήκαν δύο κλάσεις που να αντιστοιχίζουν τους συμμετέχοντες και τις σχέσεις του GUI. Οι κλάσεις αυτές είναι οι παρακάτω:

- ParticipantObject που αναφέρεται στους συμμετέχοντες του GUI.
- OfferingObject που αναφέρεται στις σχέσεις μεταξύ των συμμετοχόντων.

Αυτές οι δύο κλάσεις υλοποιούν ένα abstract interface το `AbstractObject` διότι μοιράζονται κάποιες μεθόδους με διαφορετική υλοποίηση για τις δικές τους λειτουργίες. Οι κλάσεις και οι σχέσεις του φαίνονται στην εικόνα 1.



Εικόνα-1.OfferingObject-ParticipantObject UML diagram.

Όπως προαναφέρθηκε κάθε συμμετέχοντας και σχέση περιέχουν δεδομένα. Αυτά τα δεδομένα έχουν μία ορισμένη δομή. Η δομή αυτή αποτελείται από ένα ζευγάρι τιμών. Η μία τιμή είναι για το όνομα του δεδομένου και η άλλη για την τιμή του δεδομένου. Για να γίνει αυτό δημιουργήθηκε μία αντίστοιχη κλάση. Η κλάση που τα υλοποιεί είναι η ValueData όπως φαίνεται στην εικόνα 2.

ValueData
<i>Attributes</i> protected String dataName protected Object dataValue
<i>Operations</i> public ValueData(String name, Object value) public String getValueDataName() public Object getValueDataObject() public void addValue(String name) public void addValue(Object value)

Εικόνα-2. Η κλάση που υλοποιεί τα δεδομένα.

Αφού μελετήθηκε η δομή των δεδομένων, πρέπει να περιγραφεί η αποθήκευση των συμμετεχόντων και των σχέσεων τους. Κρίθηκε σκόπιμο να αποθηκεύονται σε δύο λίστες. Η μία θα περιέχει τους συμμετέχοντες, ενώ η άλλη θα περιέχει τις σχέσεις τους. Για να γίνει αυτό υλοποιήθηκε η κλάση `RelationData`. Με αυτόν τον τρόπο κατέστη δυνατή η αναπαράσταση ενός δικτύου αξίας χωρίς τη χρήση γραφικού (GUI). Επίσης όλα τα δεδομένα του δικτύου είναι αποθηκευμένα. Η σημασία της κλάσης `RelationData` είναι ιδιαίτερη αφού ουσιαστικά είναι ο «εγκέφαλος» του πυρήνα. Οι μέθοδοι που προσφέρει διαχειρίζονται ένα μεγάλο μέρος των διεργασιών του `Value Network Tool`.

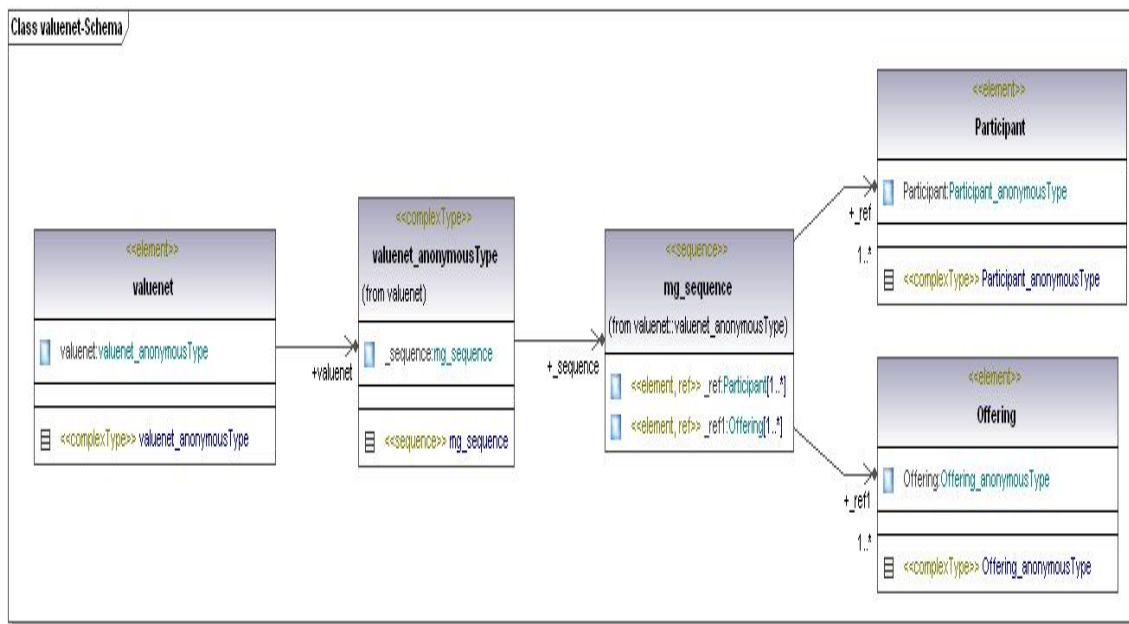
RelationData
<i>Attributes</i>
protected LinkedList nodeList
protected LinkedList arrowList
package String oldNameString
<i>Operations</i>
public RelationData()
public void clear()
public LinkedList getNodeList()
public LinkedList getArrowList()
public void addNodeValue(ParticipantObject value)
public void addArrowValue(OfferingObject value)
public void copy(RelationData source)
public void replaceNodeValue(ParticipantObject oldValue, ParticipantObject newValue)
public void replaceArrowValue(OfferingObject oldValue, OfferingObject newValue)
public void updateArrowValue(ParticipantObject oldValue, ParticipantObject newValue)
public boolean removeNodeValue(ParticipantObject value)
public boolean removeArrowValue(OfferingObject value)
public OfferingObject getArrow(OfferingObject value)
public OfferingObject getArrowWithId(int Id)
public ParticipantObject getNodeWithId(int Id)

Εικόνα-3. RelationData.

3.2. Περιγραφή XML σχήματος

Το επόμενο βήμα για το εργαλείο μας είναι να γίνει δυνατή η αποθήκευση των δεδομένων όταν κλείνει το εργαλείο και η δυνατότητα της φόρτωσης τους στο εργαλείο για περαιτέρω επεξεργασία.

Για να γίνει η παραπάνω διαδικασία, αποφασίστηκε η χρήση XML αρχείων. Ο λόγος ο οποίος οδήγησε σε αυτήν την απόφαση, ήταν η μεγάλη ευελιξία που προσφέρει η XML στην επεξεργασία της καθώς και η συχνή χρήση της στις περισσότερες νέες εφαρμογές. Έτσι ήταν επιτακτική η ανάγκη της δημιουργίας ενός XML σχήματος, τέτοιο ώστε να περιγράφει πλήρως τις ανάγκες και την μορφή των δεδομένων ενός δικτύου αξίας στο εργαλείο.

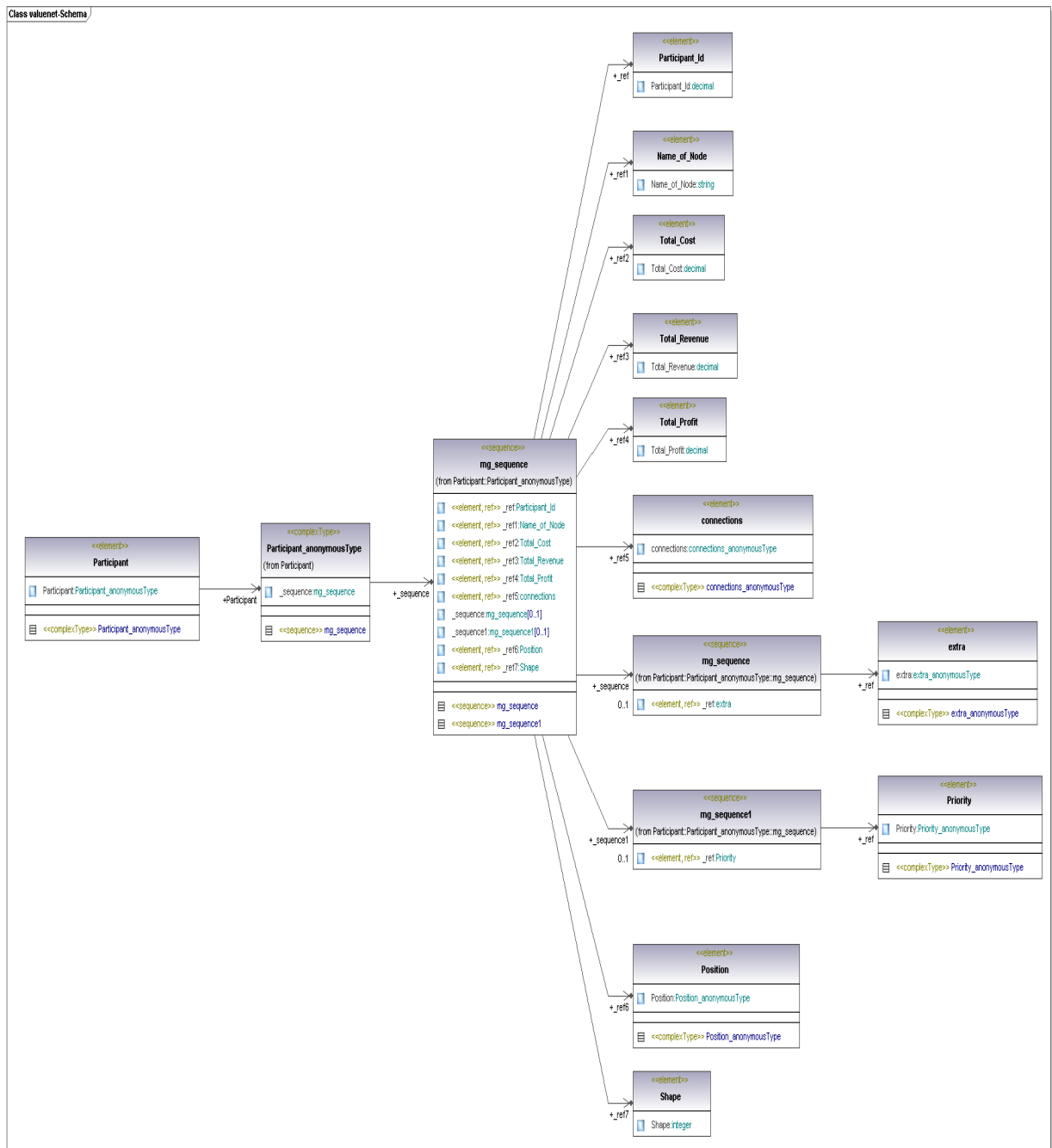


Εικόνα-4. Γενική εικόνα του valuenet σχήματος.

Το όνομα του XML σχήματος είναι valuenet και αποτελείται από δύο οντότητες το Participant που αναφέρεται στους συμμετέχοντες, και το Offering που αναφέρεται στις σχέσεις μεταξύ των συμμετεχόντων όπως φαίνεται στην εικόνα 4.

Αυτές οι δύο οντότητες έχουν κάποια ξεχωριστά χαρακτηριστικά, τέτοια ώστε να περιγράφουν καλύτερα τα δεδομένα. Θα γίνει η επεξήγηση αρχικά της οντότητας του Participant και των χαρακτηριστικών που περιλαμβάνει.

Ο Participant αποτελεί την οντότητα που περιλαμβάνει όλα τα χαρακτηριστικά που αφορούν τους συμμετέχοντες του δικτύου αξίας (βλ. Εικόνα 5).

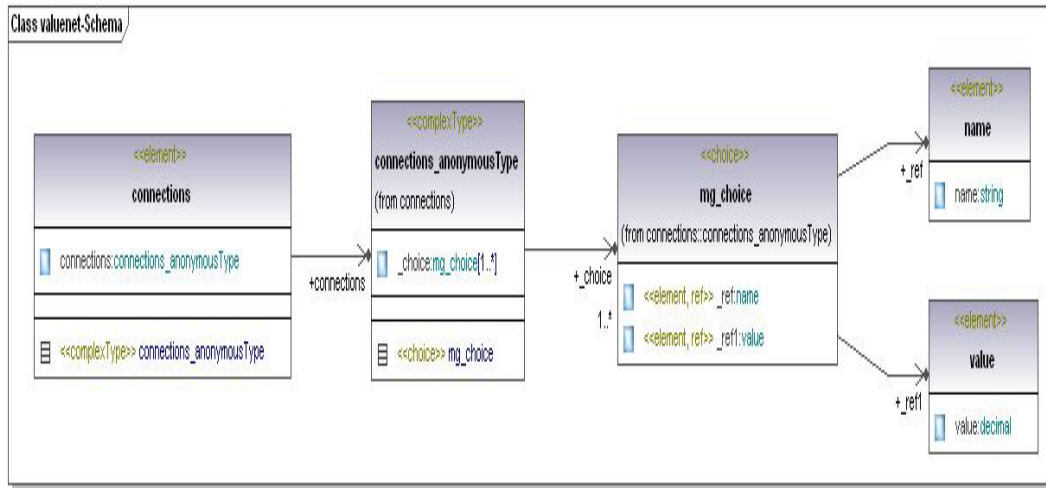


Εικόνα-5. Οντότητα του Participant.

Τα χαρακτηριστικά είναι τα εξής:

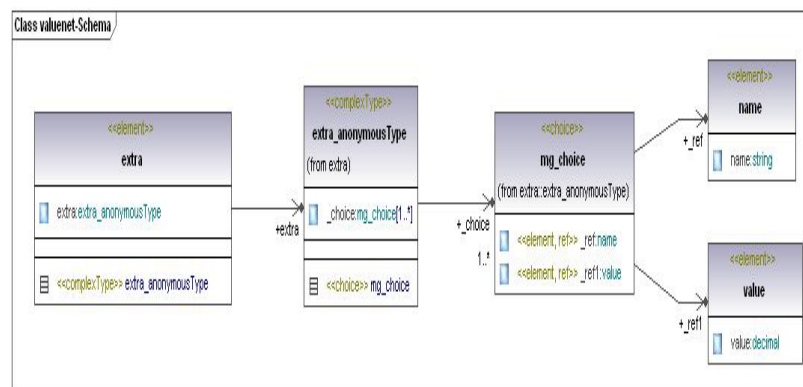
- Participant_Id: Αποτελεί ένα αύξοντα αριθμό που δημιουργείται όταν φτιάχνεται ένας συμμετέχοντας και χρησιμοποιείται κυρίως για την διαχείριση του εργαλείου.
- Name_of_Node: Είναι τό όνομα του συμμετέχοντα.

- **Total_Cost:** Συνολικό κόστος του συμμετέχοντα που προέρχεται από τους υπολογισμούς που γίνονται μέσα στο εργαλείο.
- **Total_Revenue** και **Total_Profit:** Αφορούν τα έσοδα και το κέρδος του συμμετέχοντα.
- **Connections:** Περιέχουν τις συνδέσεις που έχει ο κάθε συμμετέχοντας και περιέχει εσωτερικά ένα ζευγάρι τιμών **name** και **value**. Μια πιο λεπτομερής απεικόνιση των συνδέσεων φαίνεται στην εικόνα 6.



Εικόνα-6. connections.

- **Extra:** Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό και περιέχει όποια πληροφορία είναι χρήσιμη σύμφωνα με τον χρήστη του δικτύου (βλ. Εικόνα 7).
- **Priority:** Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό και περιέχει πληροφορία που είναι χρήσιμη για την σειρά με την οποία εμφανίζεται κάθε συμμετέχοντας στο δίκτυο.

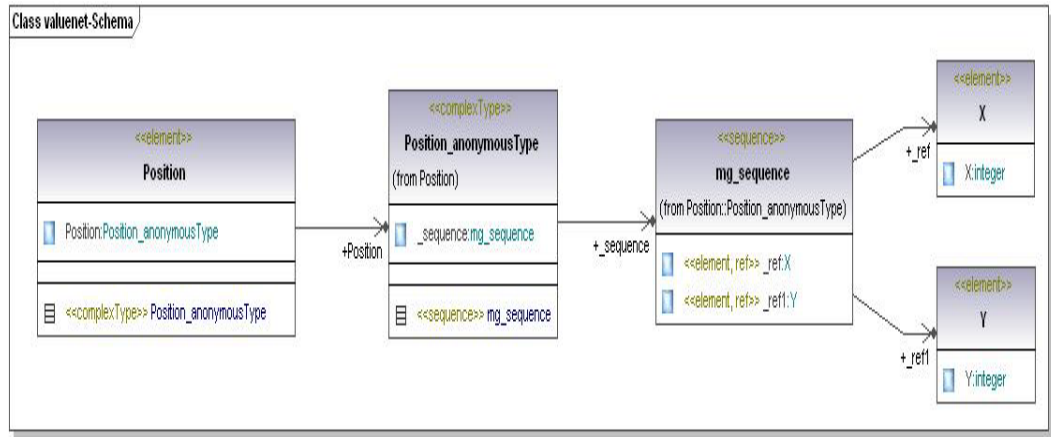


Generated by UModel

www.altova.com

Εικόνα-7. extra.

- Position: περιέχει τις συντεταγμένες του συμμετέχοντα πάνω στο GUI του εργαλείου και αποτελείται από δύο τιμές το X και το Y (βλ. Εικόνα 8).



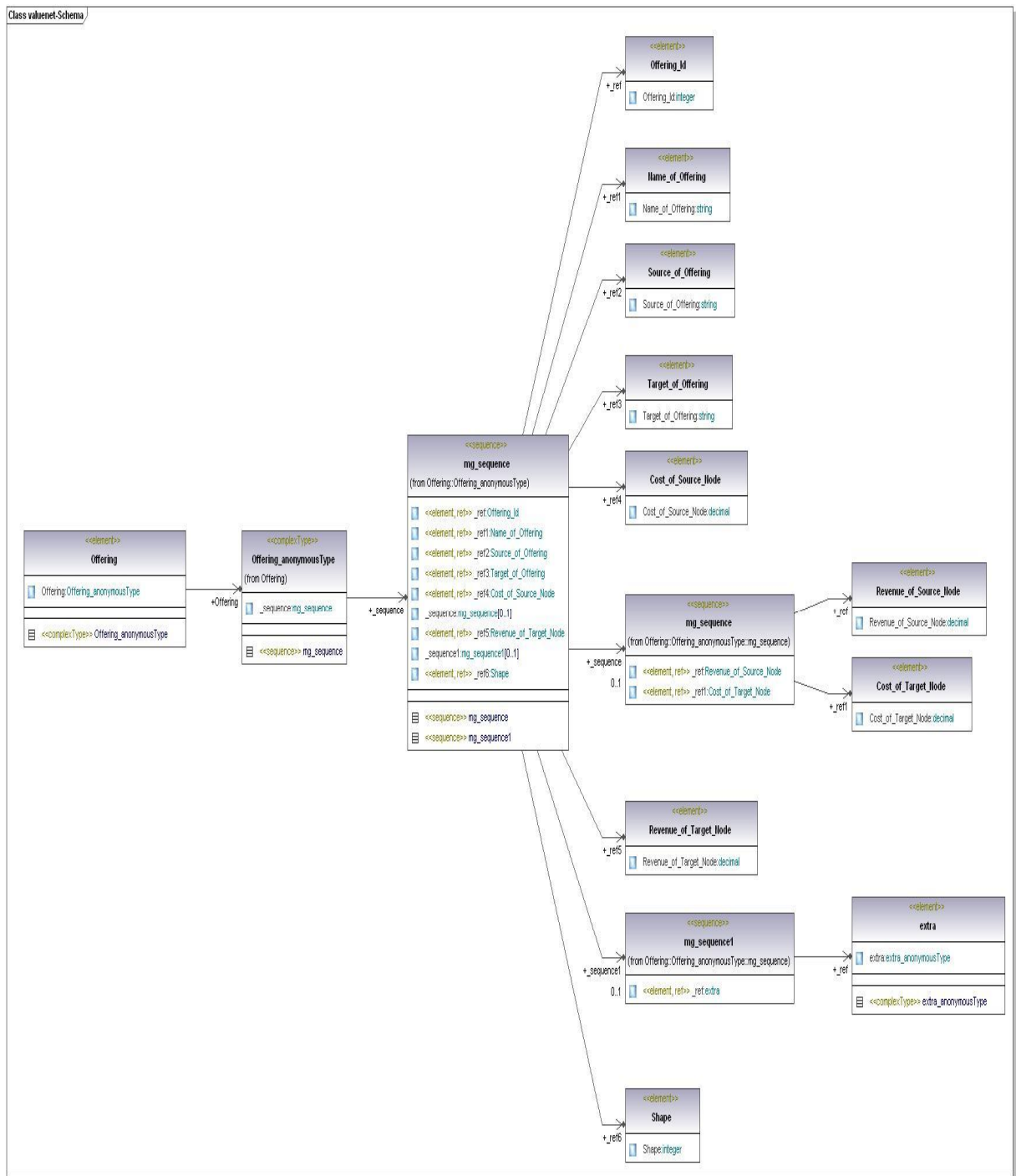
Generated by UModel

www.altova.com

Εικόνα-8. Position.

- Shape: αποτελεί πληροφορία για το σχήμα του συμμετέχοντα αν είναι τετράγωνο ή κύκλος.

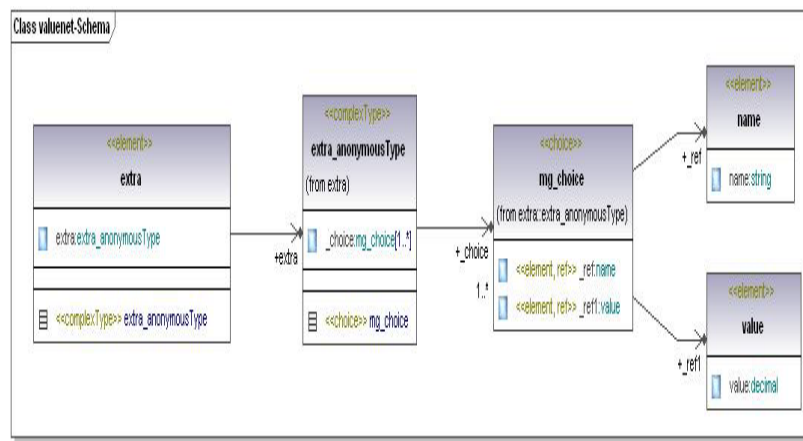
Το Offering αποτελεί την οντότητα που περιλαμβάνει όλα τα χαρακτηριστικά που αφορούν τις συνδέσεις μεταξύ των συμμετεχόντων του δικτύου αξίας (βλ. Εικόνα 9).



Εικόνα-9. Offering.

Τα χαρακτηριστικά είναι τα εξής:

- **Offering_Id:** Αποτελεί ένα αύξοντα αριθμό που δημιουργείται όταν φτιάχνεται μια σχέση και χρησιμοποιείται κυρίως για την διαχείριση του εργαλείου.
- **Name_of_Offering:** Είναι το όνομα της σχέσης.
- **Source_of_Offering** και **Target_of_Offering:** είναι τα ονόματα του αρχικού συμμετέχοντα και του τελικού συμμετέχοντα που είναι στην σχέση.
- **Cost_of_Source_Node, Cost_of_Target_Node, Revenue_of_Source_Node, Revenue_of_Target_Node:** Ανάλογα με την σχέση που υπάρχει, αυτές οι τιμές χαρακτηρίζουν τα έσοδα και τα κόστη που μεταφέρονται στους συμμετέχοντες και ανάλογα με το αν υπάρχει μονή ή διπλή σύνδεση εμφανίζονται. Όλα για την διπλή και **Cost_of_Source_Node** και **Revenue_of_Target_Node** για απλή(μονή) σύνδεση.
- **Extra:** Το συγκεκριμένο πεδίο δεν είναι υποχρεωτικό και περιέχει όποια πληροφορία είναι χρήσιμη σύμφωνα με τον χρήστη για το δίκτυο (βλ. εικόνα 10).



Generated by UModel

www.altova.com

Εικόνα-10. extra.

- **Shape:** αποτελεί πληροφορία για το αν η σχέση είναι απλή (μονή) ή διπλή.

Ο κώδικας το XML σχήματος (xsd) είναι ο παρακάτω:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://vnt.tsl.gr/schemas/"
xmlns="http://vnt.tsl.gr/schemas/"
elementFormDefault="qualified">
  <xs:element name="valuenet">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="Participant"/>
        <xs:element maxOccurs="unbounded" ref="Offering"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Participant">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Participant_Id"/>
        <xs:element ref="Name_of_Node"/>
        <xs:element ref="Total_Cost"/>
        <xs:element ref="Total_Revenue"/>
        <xs:element ref="Total_Profit"/>
        <xs:element ref="connections"/>
        <xs:sequence minOccurs="0">
          <xs:element ref="extra"/>
        </xs:sequence>
        <xs:sequence minOccurs="0">
          <xs:element ref="Priority"/>
        </xs:sequence>
        <xs:element ref="Position"/>
        <xs:element ref="Shape"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Name_of_Node" type="xs:string"/>
  <xs:element name="Total_Cost" type="xs:decimal"/>
  <xs:element name="Total_Revenue" type="xs:decimal"/>
  <xs:element name="Total_Profit" type="xs:decimal"/>
  <xs:element name="Participant_Id" type="xs:decimal"/>
  <xs:element name="connections">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="name"/>
        <xs:element ref="value"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="Priority">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="name"/>
        <xs:element ref="value"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
```

```

        </xs:complexType>
    </xs:element>
    <xs:element name="Position">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="X" />
                <xs:element ref="Y" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="X" type="xs:integer" />
    <xs:element name="Y" type="xs:integer" />

<xs:element name="Offering">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Offering_Id" />
            <xs:element ref="Name_of_Offering" />
            <xs:element ref="Source_of_Offering" />
            <xs:element ref="Target_of_Offering" />
            <xs:element ref="Cost_of_Source_Node" />
            <xs:sequence minOccurs="0">
                <xs:element ref="Revenue_of_Source_Node" />
                <xs:element ref="Cost_of_Target_Node" />
            </xs:sequence>
            <xs:element ref="Revenue_of_Target_Node" />
            <xs:sequence minOccurs="0">
                <xs:element ref="extra" />
            </xs:sequence>
            <xs:element ref="Shape" />
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="Name_of_Offering" type="xs:string" />
<xs:element name="Source_of_Offering" type="xs:string" />
<xs:element name="Target_of_Offering" type="xs:string" />
<xs:element name="Cost_of_Source_Node" type="xs:decimal" />
<xs:element name="Revenue_of_Source_Node" type="xs:decimal" />
<xs:element name="Cost_of_Target_Node" type="xs:decimal" />
<xs:element name="Revenue_of_Target_Node" type="xs:decimal" />
<xs:element name="Offering_Id" type="xs:integer" />
<xs:element name="name" type="xs:string" />
<xs:element name="value" type="xs:decimal" />
<xs:element name="extra">
    <xs:complexType>
        <xs:choice maxOccurs="unbounded">
            <xs:element ref="name" />
            <xs:element ref="value" />
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="Shape" type="xs:integer" />
</xs:schema>

```

Τα πλεονεκτήματα αυτού του σχήματος είναι η προσθήκη του χαρακτηριστικού extra το οποίο προσφέρει ευελιξία και επεκτασιμότητα στο εργαλείο αφού μπορεί να περιγράψει οποιοδήποτε δίκτυο αξίας με επιτυχία.

Για να είναι έγκυρο το δίκτυο αξίας σύμφωνα με το σχήμα πρέπει απαραίτητα να έχει συμμετέχοντες με σχέσεις, αλλά για λόγους λειτουργικότητας το εργαλείο υποστηρίζει και μη έγκυρα XML αρχεία, για την φόρτωση τους στο εργαλείο.

3.3. Περιγραφή υλοποίησης κλάσεων δημιουργίας και ανάγνωσης XML.

Μετά την περιγραφή του XML σχήματος, θα μελετηθεί η κλάση που υλοποιεί τα παραπάνω. Η κλάση η οποία είναι υπεύθυνη για την πλήρη διαχείριση των XML είναι η XMLCreation (βλ. Εικόνα 11).

XMLcreation	
<i>Attributes</i>	
protected LinkedList	parsedCells
package	Document dom
<i>Operations</i>	
public XMLcreation()	
public void exportXML(String fileName, String filePath, VNT app)	
public void transformHtml(String filepath)	
public void parseXmlFile(String filename, String filePath, RelationData data, VNT app)	
public void SelectedExport(String fileName, String filePath, VNT app)	
public void SelectedImport(String fileName, String filePath, VNT app)	

Εικόνα-11. XMLCreation.

Εμφανίζονται συνολικά πέντε μέθοδοι που δίνουν αντίστοιχα πέντε δυνατότητες. Αυτές είναι οι εξής:

- ExportXML(αποθηκεύει τα δεδομένα) και η ParseXML(φορτώνει τα δεδομένα).
- SelectedExport η οποία αποθηκεύει τα επιλεγμένα δεδομένα.
- SelectedImport η οποία φορτώνει μόνο τα επιλεγμένα δεδομένα.
- transformHtml η οποία μετασχηματίζει το XML αρχείο σε HTML με την βοήθεια ενός XSL αρχείου.

Αναλυτικότερα, η ExportXML κάνει τα εξής:

1. Αυτή η μέθοδος χρησιμοποιεί τις λίστες που της δίνει η κλάση RelationData και τις συντεταγμένες του κάθε συμμετέχοντα που δίνονται από την κλάση του JGraph. Το τελικό αποτέλεσμα είναι ένα XML αρχείο.
2. Επιπλέον η μέθοδος transformHtml καλείται μετασχηματίζει το XML αρχείο του εργαλείου σε μία HTML σελίδα. Για να γίνει αυτό πρέπει να έχει δημιουργηθεί και να δοθεί το κατάλληλο XSL αρχείο. Στην προκείμενη περίπτωση το XSL έχει δημιουργηθεί.

Αναλυτικότερα η ParseXML κάνει τα εξής:

1. Η μέθοδος αυτή διασχίζει το XML αρχείο του εργαλείου.
2. Κρατάει τα δεδομένα σε κάποιες βοηθητικές λίστες.
3. Ανακατασκευάζει το δίκτυο αξίας χρησιμοποιώντας συνδυασμένα της μεθόδους που προσφέρουν οι κλάσεις ParticipantObject, OfferingObject και JGraph.

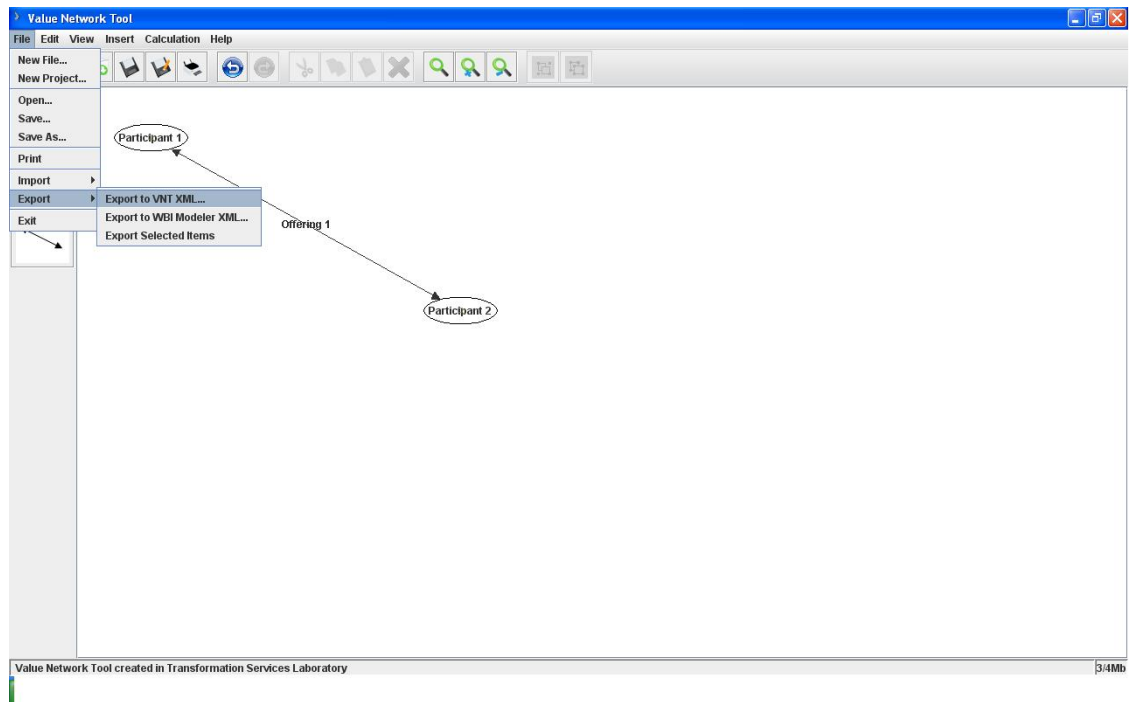
Αντίστοιχα οι κλάσεις SelectedExport και SelectedImport συμπεριφέρονται με τον ίδιο τρόπο όπως η ExportXML και η ParseXML με την μόνη διαφορά ότι το κάνουν μόνο για ένα επιλεγμένο κομμάτι του δικτύου αξίας.

4. Αναλυτική Περιγραφή Χρήσης

Σε αυτή την παράγραφο περιγράφεται η χρήση όλων των λειτουργιών που δημιουργήθηκαν κατά την υλοποίηση. Δηλαδή η αποθήκευση, η μορφή της αποθήκευσης, η φορτώση και τα αποτελέσματα της φόρτωσης των δεδομένων στο εργαλείο.

4.1. Αποθήκευση Δικτύου Αξίας

Η λειτουργία της αποθήκευσης του δικτύου αξίας γίνεται ως εξής: Εφ' όσον έχει σχεδιάσει το δίκτυο γίνεται επιλογή File → Export → Export to VNT XML (βλ. Εικόνα 12).



Εικόνα-12. Export to VNT XML.

Η παραπάνω ενέργεια θα οδηγήσει στην δημιουργία ενός XML αρχείου που περιγράφει το δίκτυο καθώς και μίας HTML σελίδας που δείχνει τα δεδομένα του δικτύου (βλ. εικόνα 13 και 14 αντίστοιχα).

```
<?xml version="1.0" encoding="UTF-8"?>
<valuenet xmlns="http://vnt.tsl.gr/schemas/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://vnt.tsl.gr/schemas/
    http://vnt.tsl.gr/schemas/valuenet.xsd">
  <Participant>
    <Participant_Id>1</Participant_Id>
    <Name_of_Node>Participant 1</Name_of_Node>
    <Total_Cost>0.0</Total_Cost>
    <Total_Revenue>0.0</Total_Revenue>
    <Total_Profit>0.0</Total_Profit>
    <connections>
      <name>Cost for Participant 2</name>
      <value>0.0</value>
      <name>Revenue from Participant 2</name>
      <value>0.0</value>
    </connections>
    <extra>
      <name>costs</name>
      <value>200</value>
    </extra>
    <Position>
      <X>42</X>
      <Y>42</Y>
    </Position>
    <Shape>0</Shape>
  </Participant>

  <Participant>
    <Participant_Id>2</Participant_Id>
    <Name_of_Node>Participant 2</Name_of_Node>
    <Total_Cost>0.0</Total_Cost>
    <Total_Revenue>0.0</Total_Revenue>
    <Total_Profit>0.0</Total_Profit>
    <connections>
      <name>Cost for Participant 1</name>
      <value>0.0</value>
      <name>Revenue from Participant 1</name>
      <value>0.0</value>
    </connections>
    <extra>
      <name>extra</name>
      <value>1000</value>
      <name>insurance</name>
      <value>20</value>
    </extra>
    <Position>
      <X>396</X>
      <Y>240</Y>
    </Position>
    <Shape>0</Shape>
  </Participant>

  <Offering>
    <Offering_Id>1</Offering_Id>
    <Name_of_Offering>Offering 1</Name_of_Offering>
    <Source_of_Offering>Participant 1</Source_of_Offering>
    <Target_of_Offering>Participant 2</Target_of_Offering>
    <Cost_of_Source_Node>0.0</Cost_of_Source_Node>
    <Revenue_of_Source_Node>0.0</Revenue_of_Source_Node>
    <Cost_of_Target_Node>0.0</Cost_of_Target_Node>
    <Revenue_of_Target_Node>0.0</Revenue_of_Target_Node>
    <extra>
      <name>funds</name>
      <value>40</value>
    </extra>
    <Shape>0</Shape>
  </Offering>
</valuenet>
```

Εικόνα-13. XML αρχείο

Valuenet

Participant

Name_of_Node	Total_Cost	Total_Revenue	Total_Profit	connections name	connections value	extra name	extra value
Participant 1	0.0	0.0	0.0	Cost for Participant 2	0.0	costs	200
				Revenue from Participant 2	0.0		
Participant 2	0.0	0.0	0.0	Cost for Participant 1	0.0	extra	1000
				Revenue from Participant 1	0.0	insurance	20

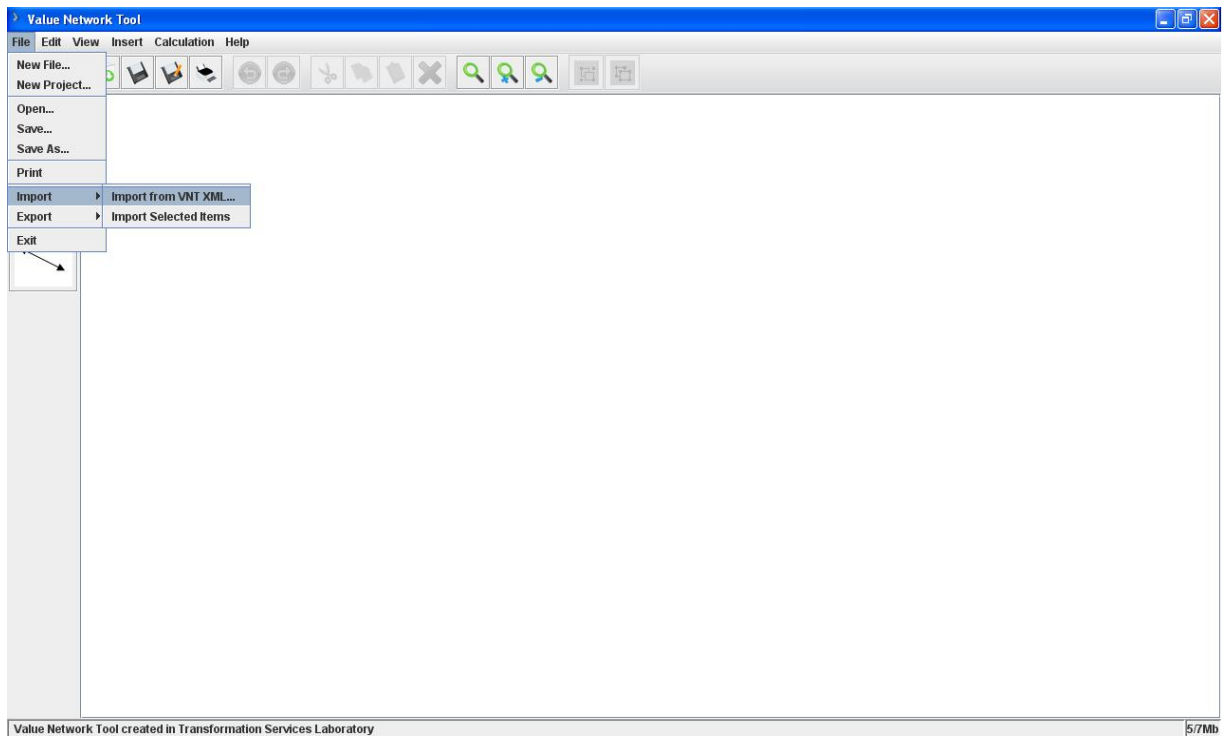
Offering

Name_of_Offering	Source_of_Offering	Target_of_Offering	Cost_of_Source_Node	Revenue_of_Source_Node	Cost_of_Target_Node	Revenue_of_Target_Node	extra name	extra value
Offering 1	Participant 1	Participant 2	0.0	0.0	0.0	0.0	funds	40

Εικόνα-14. HTML σελίδα.

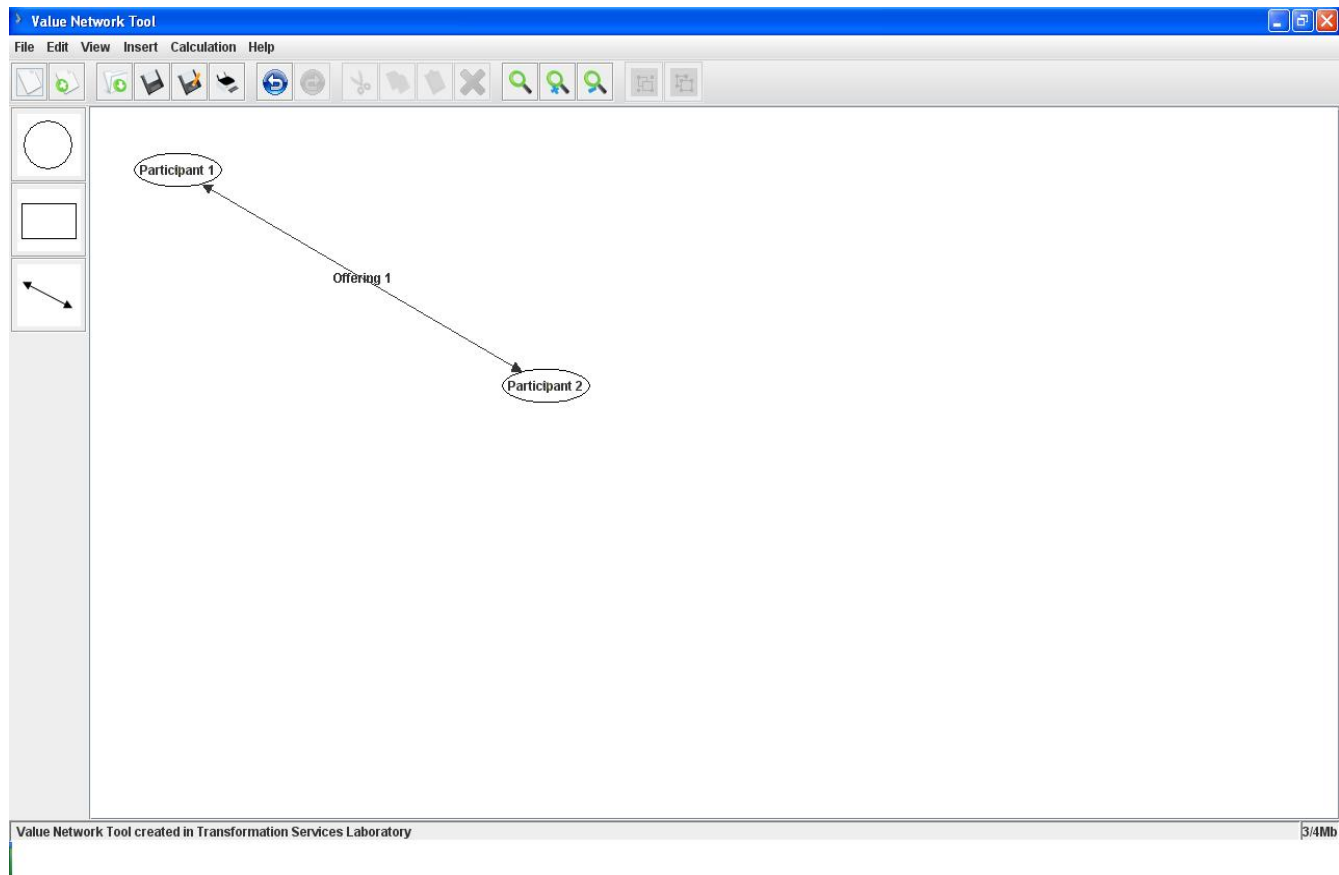
4.2. Φόρτωση Δικτύου Αξίας

Η λειτουργία της φόρτωσης του δικτύου αξίας γίνεται ως εξής: File → Import → Import from VNT XML. Γίνεται η επιλογή του αρχείου για φόρτωση (βλ. Εικόνα 15).



Εικόνα-15. Import XML.

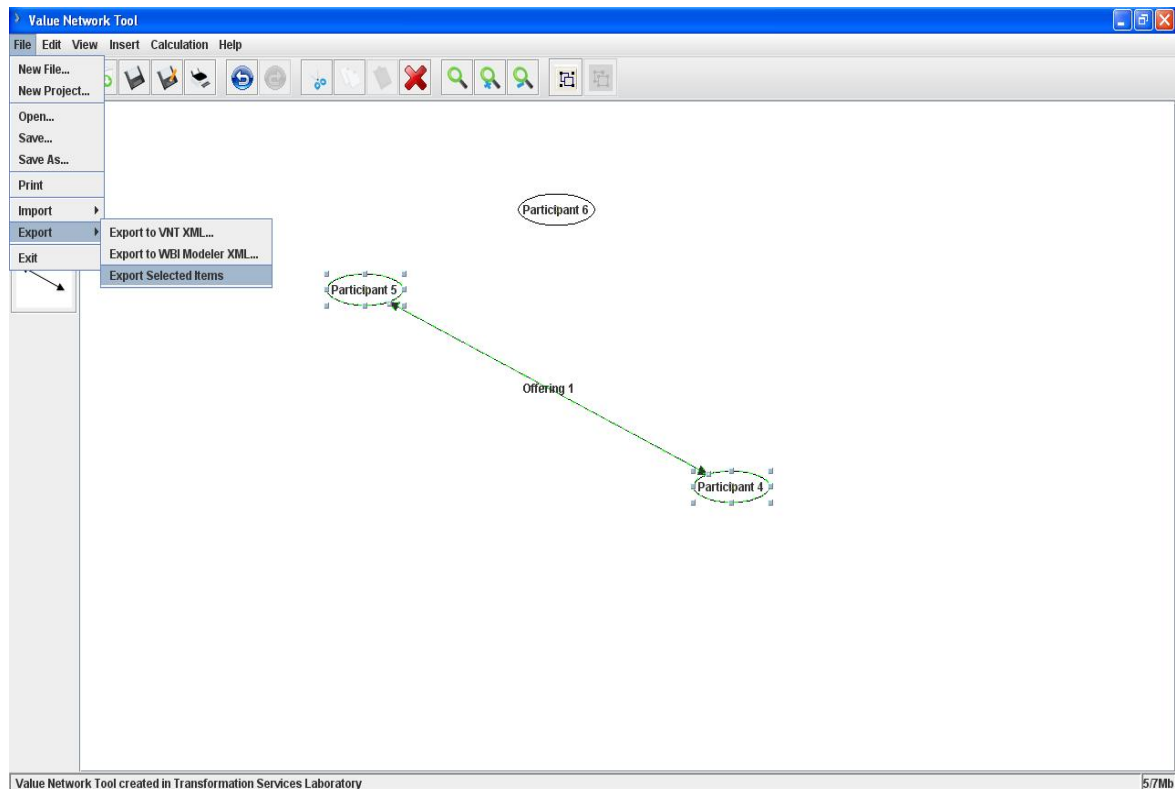
Το αποτέλεσμα της παραπάνω ενέργειας είναι η δημιουργία του δικτύου που απεικονίζει το XML αρχείο (βλ. εικόνα 16).



Εικόνα-16. Δίκτυο από φόρτωση.

4.3. Αποθήκευση επιλεγμένου μέρους δικτύου

Η λειτουργία της αποθήκευσης επιλεγμένου μέρους του δικτύου αξίας, γίνεται ως εξής: Εφ' όσον έχει σχεδιαστεί το δίκτυο γίνεται επιλογή του μέρους που θέλει ο χρήστης και γίνεται η αποθήκευση κάνοντας File → Export → Export Selected Items (βλ. εικόνα 17).



Εικόνα-17. Export Selected Items.

Αυτό έχει ως αποτέλεσμα τη δημιουργία ενός default XML αρχείου που έχει τα επιλεγμένα δεδομένα του δικτύου αξίας (βλ. εικόνα 18).

```
<?xml version="1.0" encoding="UTF-8"?>
<valuenet xmlns="http://vnt.tsl.gr/schemas/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://vnt.tsl.gr/schemas/
    http://vnt.tsl.gr/schemas/valuenet.xsd">
  <Participant>
    <Participant_Id>4</Participant_Id>
    <Name_of_Node>Participant
4</Name_of_Node>
    <Total_Cost>0.0</Total_Cost>
    <Total_Revenue>0.0</Total_Revenue>
    <Total_Profit>0.0</Total_Profit>
    <connections>
      <name>Cost for Participant 5</name>
      <value>0.0</value>
      <name>Revenue from Participant
5</name>
      <value>0.0</value>
    </connections>
    <Position>
      <X>672</X>
      <Y>360</Y>
    </Position>
    <Shape>0</Shape>
  </Participant>

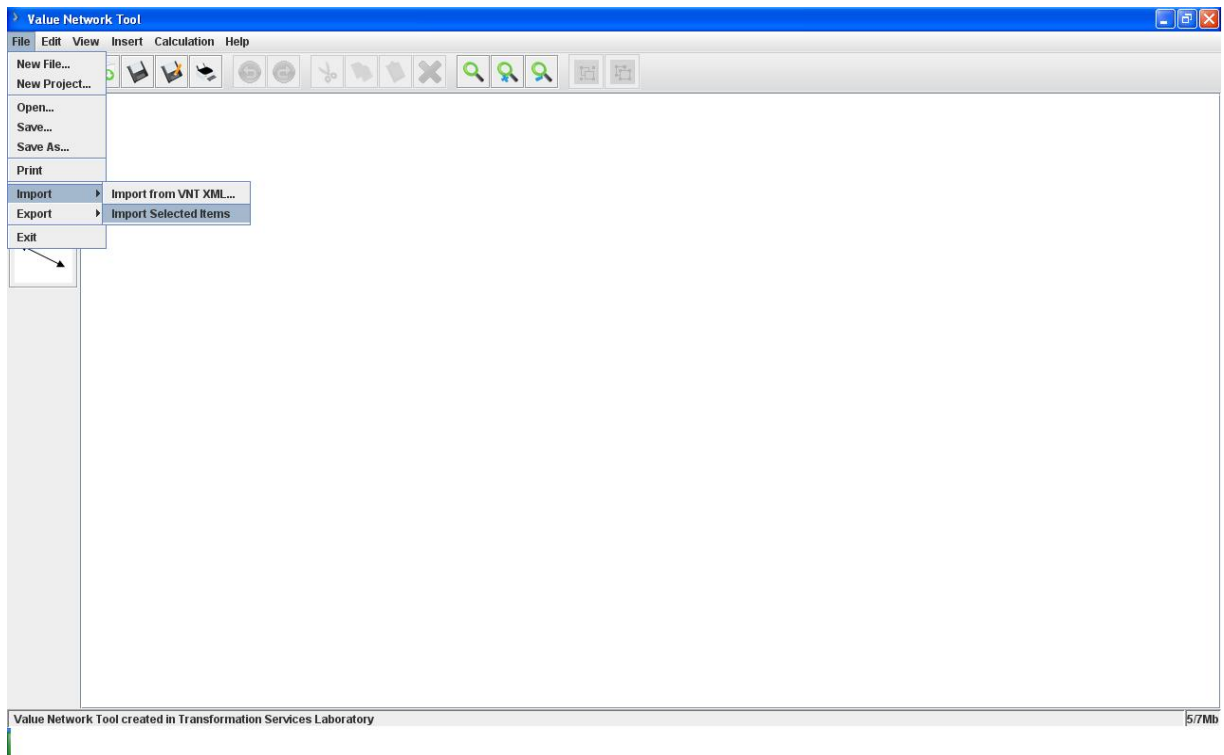
  <Participant>
    <Participant_Id>5</Participant_Id>
    <Name_of_Node>Participant 5</Name_of_Node>
    <Total_Cost>0.0</Total_Cost>
    <Total_Revenue>0.0</Total_Revenue>
    <Total_Profit>0.0</Total_Profit>
    <connections>
      <name>Cost for Participant 4</name>
      <value>0.0</value>
      <name>Revenue from Participant 4</name>
      <value>0.0</value>
    </connections>
    <Position>
      <X>270</X>
      <Y>168</Y>
    </Position>
    <Shape>0</Shape>
  </Participant>

  <Offering>
    <Offering_Id>1</Offering_Id>
    <Name_of_Offering>Offering 1</Name_of_Offering>
    <Source_of_Offering>Participant 4</Source_of_Offering>
    <Target_of_Offering>Participant 5</Target_of_Offering>
    <Cost_of_Source_Node>0.0</Cost_of_Source_Node>
    <Revenue_of_Source_Node>0.0</Revenue_of_Source_Node>
    <Cost_of_Target_Node>0.0</Cost_of_Target_Node>
    <Revenue_of_Target_Node>0.0</Revenue_of_Target_Node>
    <Shape>0</Shape>
  </Offering>
</valuenet>
```

Εικόνα-18. Selected Items XML αρχείο.

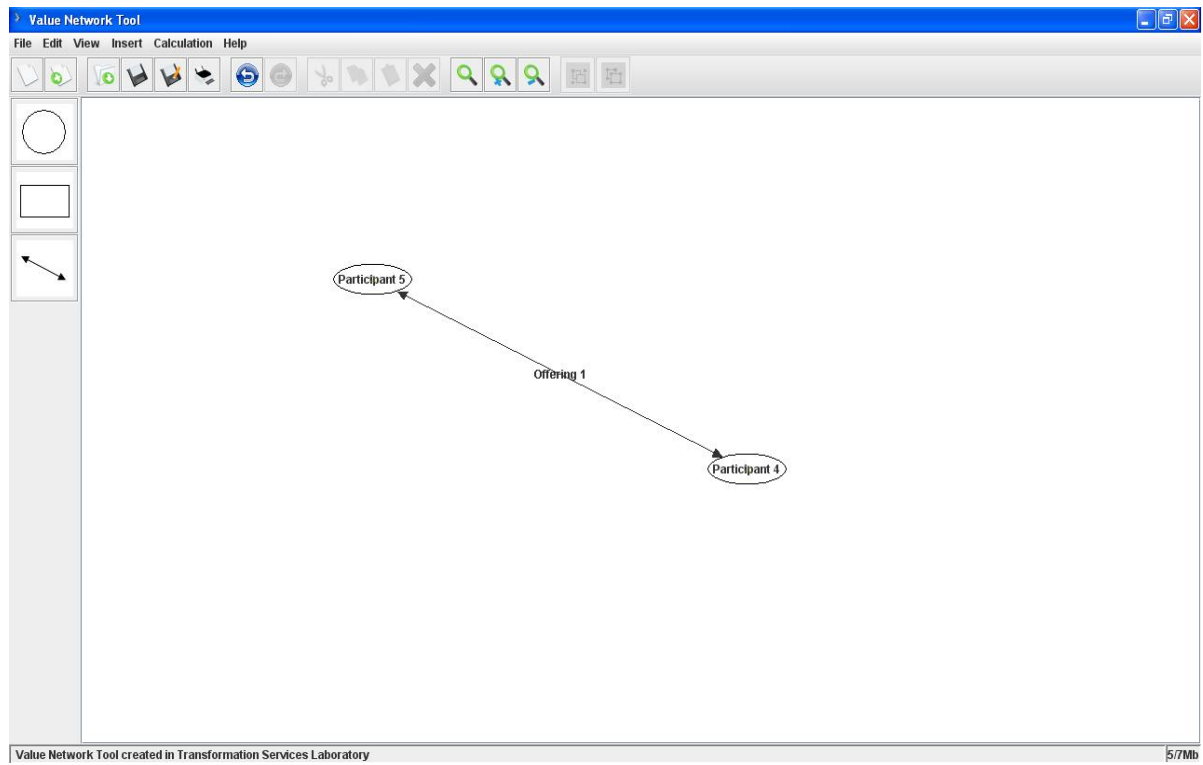
4.4. Φόρτωση Επιλεγμένου δικτύου Αξίας

Η λειτουργία της φόρτωσης του επιλεγμένου μέρους του δικτύου αξίας, γίνεται ως εξής:
Με την επιλογή File → Import → Import Selected Items (βλ. εικόνα 19).



Εικόνα-19. Import Selected Items.

Έτσι φορτώνεται αυτόματα το κομμάτι που περιγράφει το Selected XML αρχείο και εμφανίζεται στο εργαλείο (βλ. εικόνα 20).



Εικόνα-20. Αποτέλεσμα φόρτωσης επιλεγμένων.

5. Σύνοψη-Μελλοντικές Επεκτάσεις

Συνοψίζοντας, παρουσιάσαμε ένα τρόπο αναπαράστασης, μοντελοποίησης, και προσομοίωσης των δικτύων αξίας. Χρησιμοποιήσαμε τη γλώσσα προγραμματισμού Java διότι είναι ευέλικτη και επεκτάσιμη. Η επιλογή της συγκεκριμένης σχεδίασης και της λύσης που επιλέχθηκε έδωσε την δυνατότητα της εύκολης προσθήκης επιπλέον εφαρμογών και πράξεων.

Επιπλέον, η χρήση XML σχήματος επιτρέπει σε οποιονδήποτε να δημιουργήσει ένα XML αρχείο και να το φορτώσει στο εργαλείο. Επίσης η επιλογή της δημιουργίας αυτού του σχήματος είναι ένα ιδιαίτερα σημαντικό κομμάτι της δουλειάς αφού η δομή του το κάνει ευέλικτο και επεκτάσιμο αφού μπορεί να περιγράψει οποιοδήποτε δίκτυο αξίας.

Τέλος ο μετασχηματισμός του XML αρχείου σε HTML σελίδα βοηθά στην ευκολότερη κατανόηση των δεδομένων ενός δικτύου αξίας.

Μελλοντικά αναζητείται η περαιτέρω βελτίωση του εργαλείου, προσδίδοντας του μεγαλύτερη σταθερότητα και μεγαλύτερο αριθμό εφαρμογών.

Αναφορές

1. N. Caswell, S. Feldman, C. Nikolaou, J. Sairamesh, M. Bitsaki, G. D. Koutras and G. Iacovidis. "Estimating Value in Service Systems – A theory and an example". In IBM Systems Journal, Volume 47, No 1, 2008.
2. J. Spohrer, P. P. Maglio, J. Bailey, and D. Gruhl, "Steps Toward a Science of Service Systems," IEEE Computer 40, No. 1, 71–77 (2007).
3. Allee Verna, "Reconfiguring the Value Network", Journal of Business Strategy, Vol. 21, N 4, July-Aug 2000.
4. Jaap Gordijn, Hans Akkermans, "Designing and evaluating e-business models", in IEEE Intelligent Systems, Vol. 16(4):11-17, 2001..
5. Stylus Studio <http://www.stylusstudio.com/docs/>
6. The XML API <http://xerces.apache.org/xerces2-j/javadocs/api/index.html>
7. Tools of the Trade, Building Graphs with JGraph: <http://www.sampublishing.com/articles/article.asp?p=431106&seqNum=1&rl=1>.
8. JGraph User Manual: <http://www.jgraph.com/pub/jgraphmanual.pdf>.
9. JGraph Source Code Distribution: <http://www.jgraph.com/pub/jgraph-latest-lgpl-src.jar>.
10. JGraph Community Help Forum: <http://www.jgraph.com/forum>.
11. JGraph API specification: <http://www.jgraph.com/pub/api>.
12. JGraph Frequently-Asked Questions: <http://www.jgraph.com/faq.html>.
13. Zarras A. "Value Network Tool: A Tool for Modeling, Simulating and Representating Value Networks". Technical Report, University of Crete.