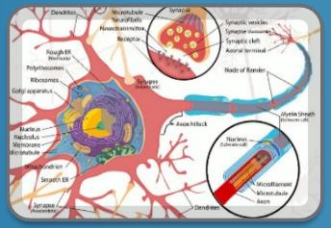
NAM-TREE

My own methodology for A.I.

Hansung Univ.

Jeongho Nam

목차

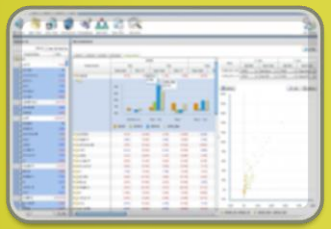


Outline, What Nam-Tree is?

- Base conception
- Composition principle
- Characteristics of A.N.N.

Manual, How to use?

- Layout
- Nam-Tree with examples
- Architecture, customize as I want



Blueprint, Plans for future

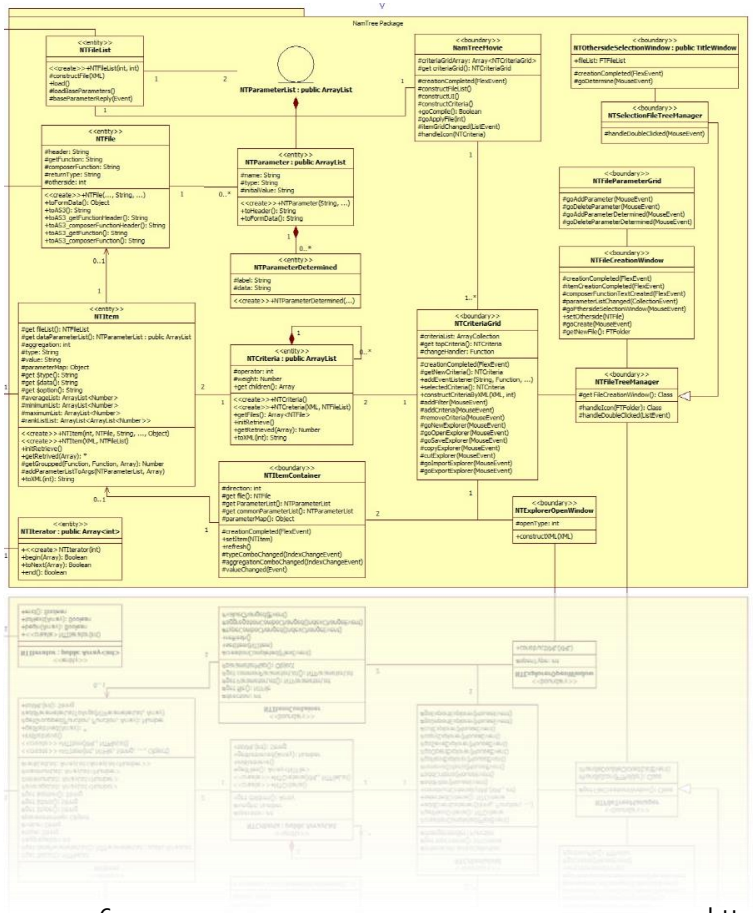
- Add targets to optimize
- Migration
- Genetic Algorithm

1. OUTLINE

What Nam-Tree is?

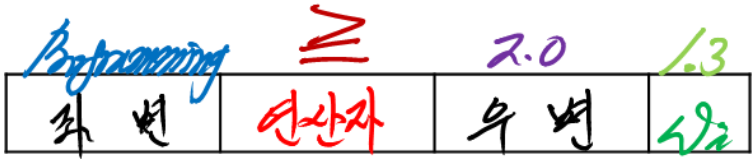
1. Base Conception
2. Composition Principle
3. Characteristics of ANN

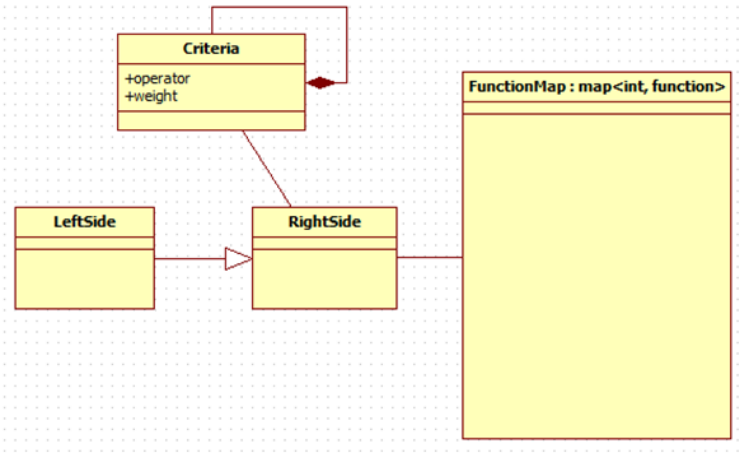
1. Base Conception – Approach



- If **logics** can be **objectified**,
  - Logics can be structured,
  - Can be coded (by serialization),
- By Genetic Algorithm (Not realized)
  - Can create infinite logics
    - By create sequences,
    - constructing objects by the sequences
  - And this is the Artificial Intelligence

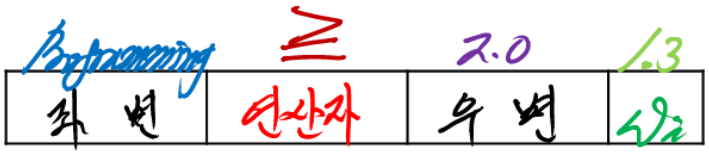
1. Base Conception – Approach

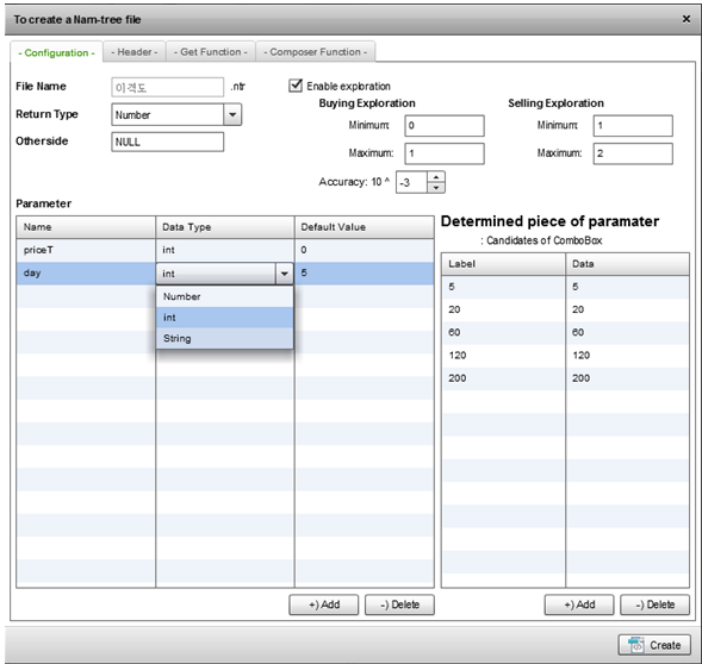




- Condition object has
  - 1:N recursive relationship
  - Left & right side, operator, weight
- Each side has a function, then
  - (function pointer)
- Its forms is tree-structured, so that,
  - Logics can be extended infinitely
    - By the 1:N recursive relationship
  - All conditions can be expressed,
- Can be objectified, so that
  - Coding (serialization) is possible
    - Create conditions by GA,
    - **then it's the A.I.**

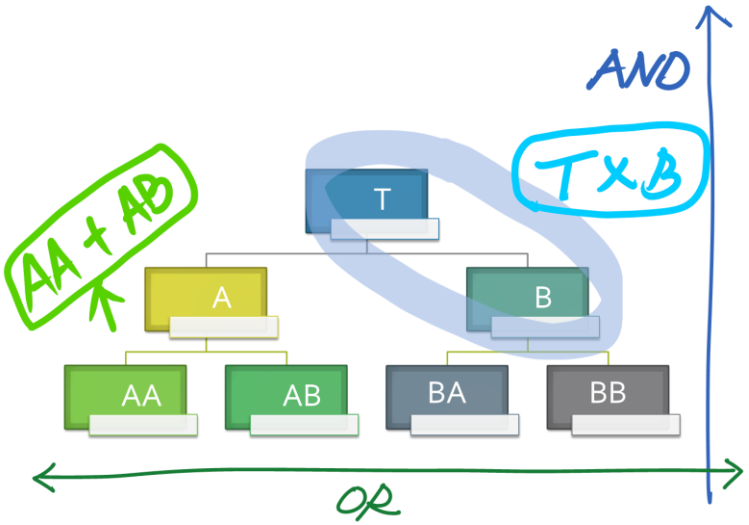
2. Comp. Principle – A condition





- **Member variables**
  - Divided by left & right side
  - Each side can has
    - Constant value or,
    - Function pointer
- **Optimization**
  - Each side's constant can be optimized
  - Parameters in function can be, too
- **Result value**
  - The condition is TRUE, then W_i,
    - TRUE (1) x weight (W_i)
  - FALSE, then return oo
    - FALSE (0)

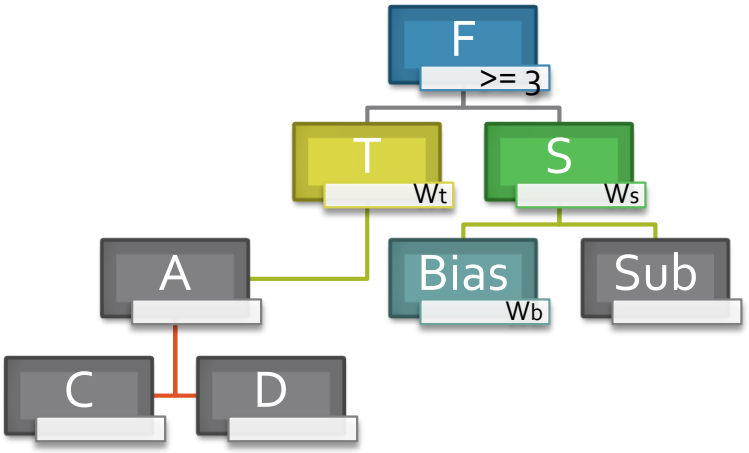
2. Comp. Principle – Hierarchical Relationship (X, +)



- **Horizontal relationship**
  - A child under same parent
    - OR
    - Plus (+)
- **Vertical relationship**
  - Relationsip between parent and child
    - AND
    - Multiply (X)
- Ex) In the left case,
  - T &&

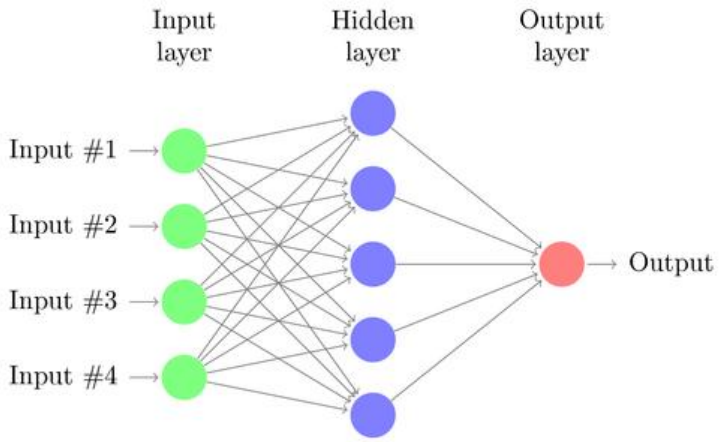
    {A && (AA || AB)} || {B && (BA || BB)}
  - T x [{A x (AA + AB)} + {B x (BA + BB)}]

2. Comp. Principle – Filter, Weight, Bias



- **Filter**
  - Filter a calculated (returned) value
  - **F** = (T + S) >= 3 ? 1 : 0
- **Weight**
  - Multiply at each condition, so that,
    - Enable to reflect importance
  - (T_{W_t} + S_{W_t} >= 3 ? 1 : 0) x W_f
- **Bias**
  - Make a child condition to be always true (ex: 1 = 1)
  - Adjust the weight, then it's the bias

3. Characteristics of ANN – Layer



- **Input layer**
  - Sample and Learning data
  - Data that will be input to Nam-Tree
- **Hidden layer** -> Nam-Tree
  - Condition tree of Nam-Tree
  - But, the condition composed not by GA automatically but by user manually,
    - Hard to be categorized in Hidden layer
- **Output layer**
  - User must define here.
  - The user-defined result value which utilized the returned value from Nam-Tree to calculate something user wants.
  - Last purpose of Nam-Tree

3. Characteristics of ANN – Target

- **Learning by Example**
  - Continuously suggests the example pairs (input, output) so that, try to construct object formed mapping
- **Generalization**
  - Neural Network completed learning prints right result although not-learned input is inserted

- **Associative memory**
  - An ability to find the most likely output pattern, which is already stored, for input which is clearly new pattern or some data is lost
- **Fault tolerance**
  - Neural Network has a lot of neurons and they are connected each other
  - But, even some connections or some neurons were collapsed, Neural Network ensures to operating normally by left others

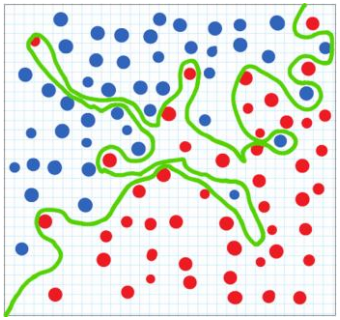
3. Characteristics of ANN – Target

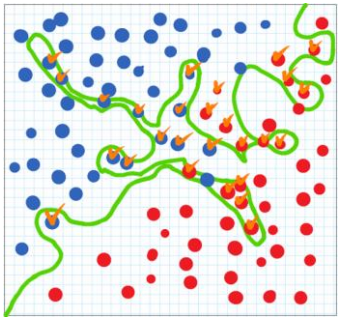
Learning by example

- **Automatic logic construction by GA**
  - Learning has to implemented not only to a condition but also to all condition set including automation and optimization
  - Condition is not composed by human,
    - But also by program
- At now,
  Coding by objectification is completed
  - By it, implementing GA's
    - Selection, mutation, crossover are
    - Left to user's work

Generalization

- Input is changed, then hard to adjust
- Problems of Local Solution
  - Extrapolation is dangerous
  - The problem all non-linear model has
    - There's no basic solution

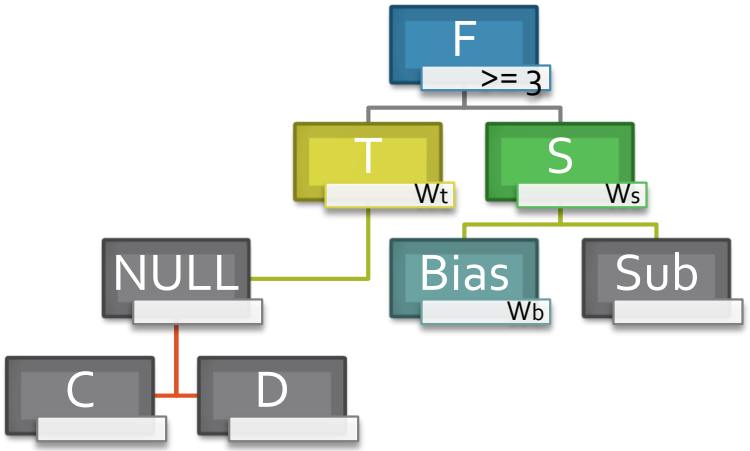




3. Characteristics of ANN – Target

- **Associative memory**
  - Deficient data like omission
  - I don't provide any solution about this
  - Handle as Fault, Do not estimate

- **Fault tolerance**
  - Phase in the concept "NULL" in DB
    - NULL in Programming -> 0
    - NULL in DB -> unknown
  - Ignore the object contains NULL

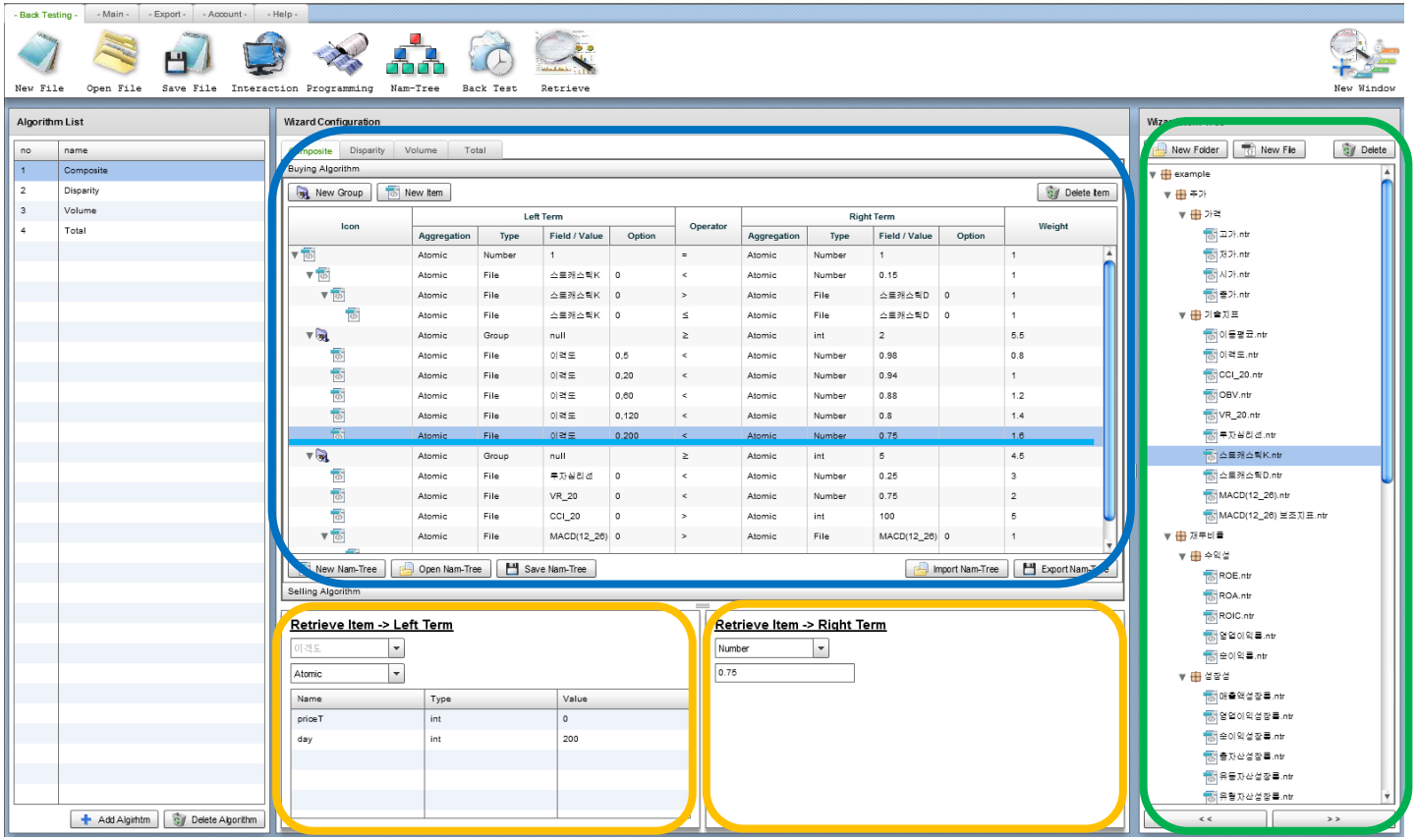


2. MANUAL

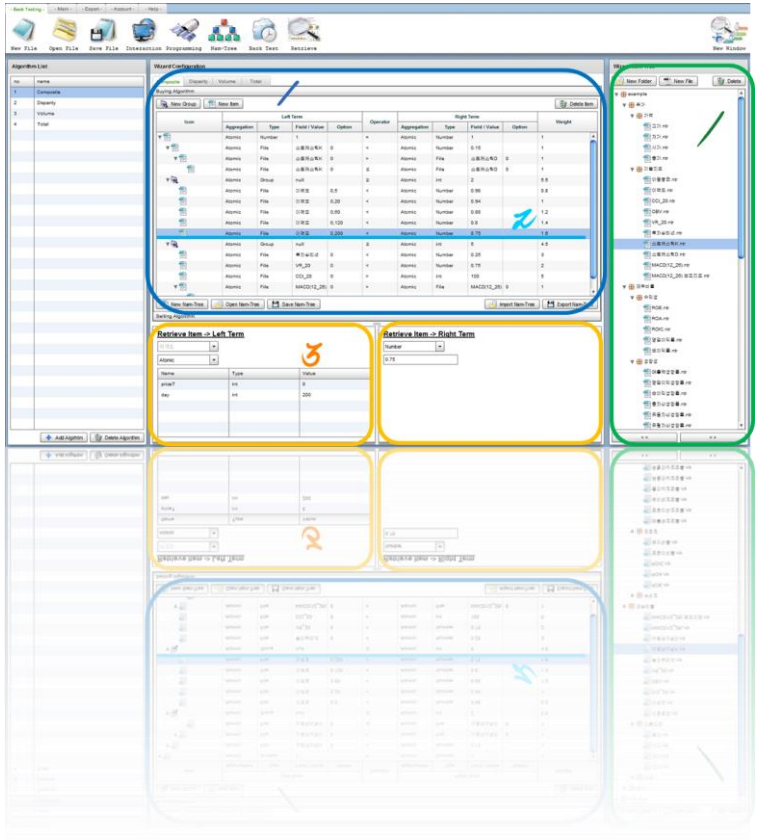
How to use?

1. Layer
2. Nam-Tree with Examples
3. Architecture, customize as I want

1. Layout


- Back Testing -
- Main -
- Export -
- Account -
- Help -
New File
Open File
Save File
Interaction
Programming
Nam-Tree
Back Test
Retrieve
New Window
Algorithm List
no
name
1
Composite
2
Disparity
3
Volume
4
Total
Add Algirithm
Delete Algorithm
Wizard Configuration
Composite
Disparity
Volume
Total
Buying Algorithm
New Group
New Item
Delete Item
Icon
Left Term
Aggregation
Type
Field / Value
Option
Operator
Right Term
Weight
New Nam-Tree
Open Nam-Tree
Save Nam-Tree
Import Nam-Tree
Export Nam-Tree
Selling Algorithm
Retrieve Item -> Left Term
Atomic
Name
Type
Value
priceT
int
0
day
int
200
Retrieve Item -> Right Term
Number
0.75
New Folder
New File
Delete
example

1. Layout



- **Nam-Tree (Condition) Grid**
  - A grid expressing hierarchical structure of conditions.
- **Condition Row (Record)**
  - A Row of Nam-Tree Grid
    - It means, a condition, itself
  - When a row is selected then,
    - The row's left & right side will be
    - Expressed in Parameter Container
- **Parameter Container**
  - Express parameters of
  - Selected row's left & right side
- **User-defined function File-Tree**

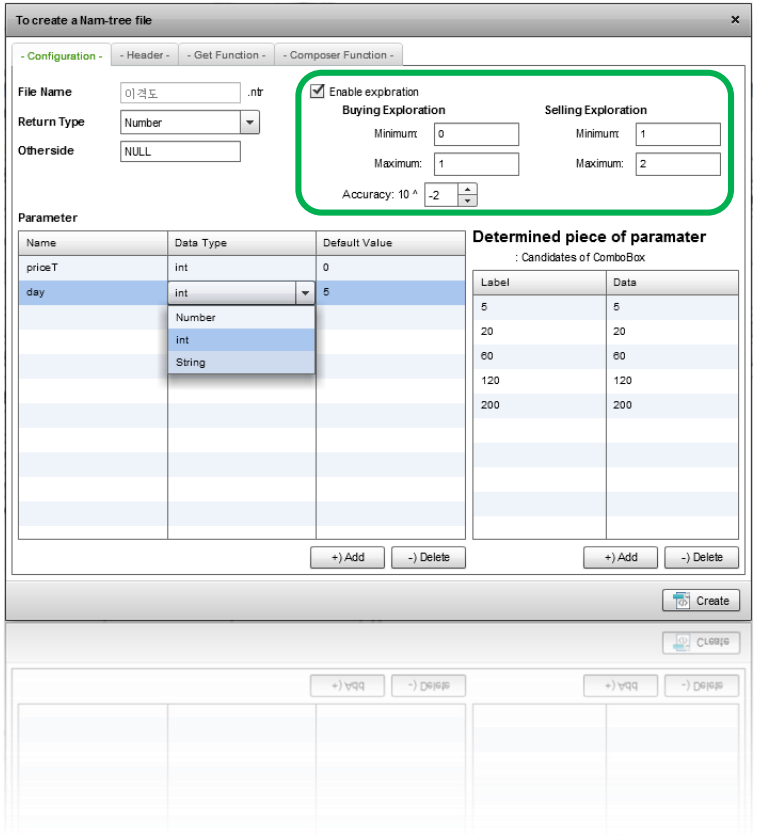
1. Layout – Condition Grid

1. Layout – Condition Grid

- **New Filter**
  - Add a new filter as a child object under the selected one
  - (Child returns) [__operator__] right side ?
    Wi : o
- **New Criteria**
  - Add a new criteria as a child object under the selected one

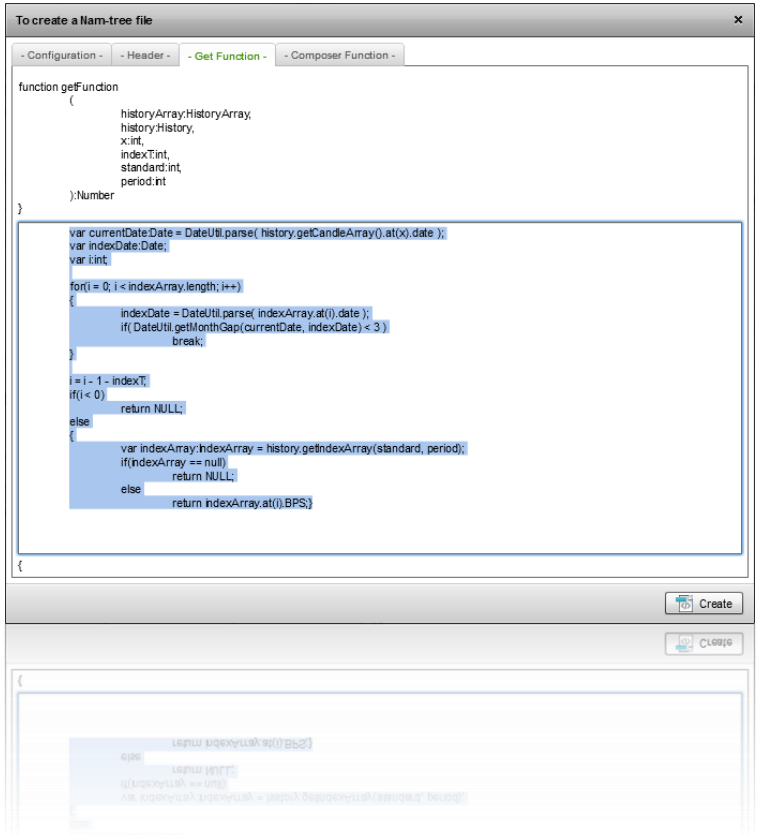
- **Left & Right Side**
  - Show the left & right side's parameters of each record
- **Operator**
  - Show operator (>, <, ==, >=, <=)
  - You can modify the operator directly on the Grid by clicking the condition (record) 's operator field
- **Weight**
  - Show weight value of each condition
  - Like the operator case, you can modify the weight directly on Grid
- **File Handler**
  - You can save and load current hierarchical conditions

1. Layout – Function Manager



- **Other-side function**
  - If the other-side has set, the two functions are inserted at once as a pair
  - ex) Insert current function to left-side in condition, then the other-side function will be inserted to the right-side at the same time
- **Exploratory Configuration**
  - Variables of optimal configuration
    - Minimum & Maximum
    - Exploratory Accuracy
- **Parameter**
  - It's the parameter, literally
- **Determined Parameter Set**
  - You can pre-set the candidates of parameter
    - Candidates are labeled in Combo-Box

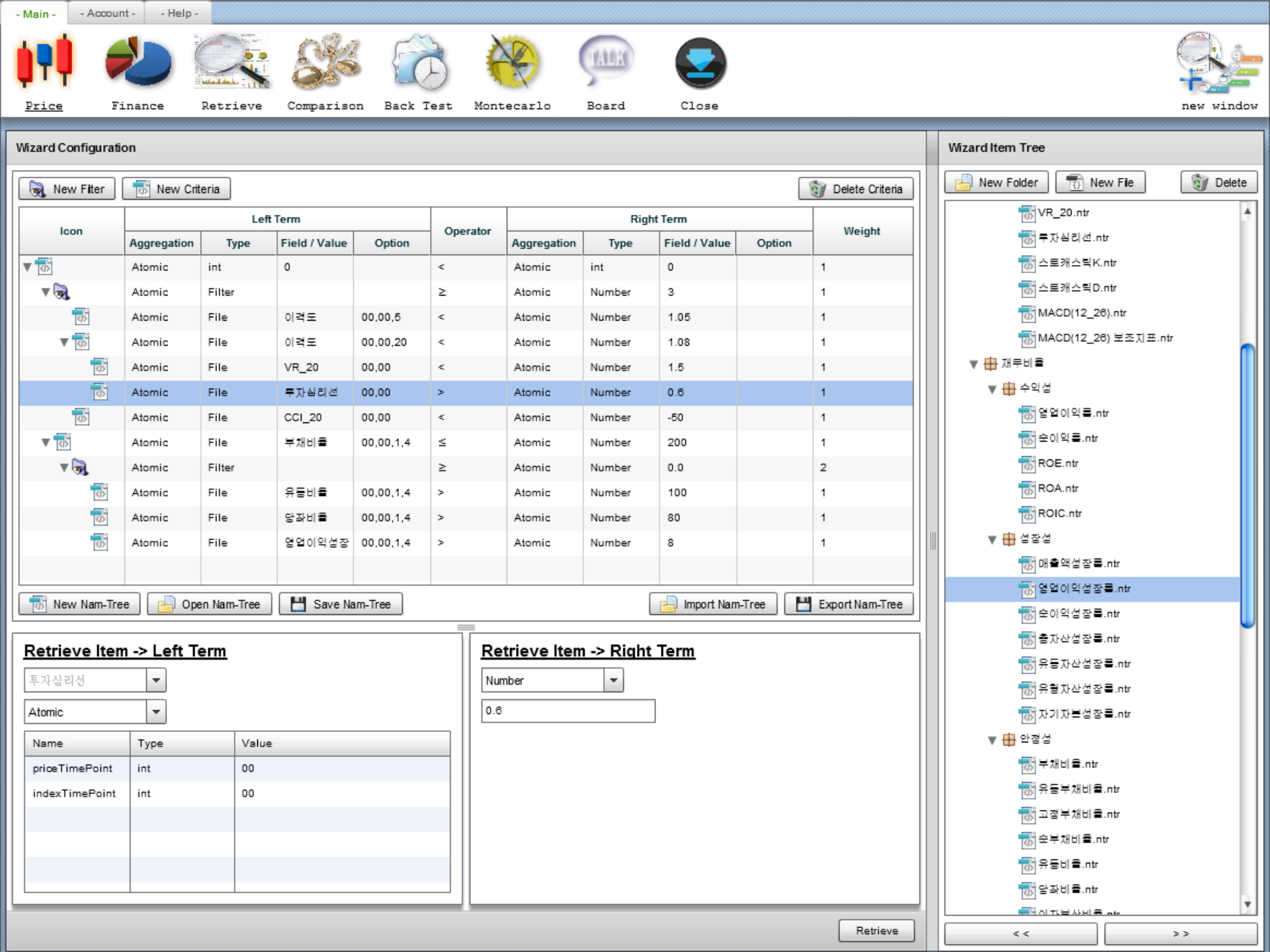
1. Layout – Function Manager



- Code field of user-defined function
  - **Header**
    - A code which will be executed directly after the compile
  - **Get Function**
    - A function code which will be executed by each side of condition.
      - NTSide in NTCriteria
    - This is the function which is linked as a function pointer in logic condition
    - ex) Call from condition's left-side
  - **Composer Function**
    - A function which is called when the "Get Function" was called at first.

2. Example – Conditional Finding

- Samchon Simulation, "Retrieve"
  - http://samchon.org/
  - Upper Menu-> Main -> Retrieve
- Hansung Timetable, "조건 검색"
  - Account for guest
    - id: guest
    - No password
  - 상단메뉴 -> Main -> 조건 검색


- Main -
- Account -
- Help -
Price
Finance
Retrieve
Comparison
Back Test
Montecarlo
Board
Close
new window
Wizard Configuration
New Filter
New Criteria
Delete Criteria
Icon
Left Term
Aggregation
Type
Field / Value
Option
Operator
Right Term
Aggregation
Type
Field / Value
Option
Weight
Atomic int 0 < Atomic int 0 1
Atomic Filter ≥ Atomic Number 3 1
Atomic File 이격도 00,00,5 < Atomic Number 1.05 1
Atomic File 이격도 00,00,20 < Atomic Number 1.08 1
Atomic File VR_20 00,00 < Atomic Number 1.5 1
Atomic File 투자심리선 00,00 > Atomic Number 0.6 1
Atomic File CCI_20 00,00 < Atomic Number -50 1
Atomic File 부채비율 00,00,1,4 ≤ Atomic Number 200 1
Atomic Filter ≥ Atomic Number 0.0 2
Atomic File 유동비율 00,00,1,4 > Atomic Number 100 1
Atomic File 당좌비율 00,00,1,4 > Atomic Number 80 1
Atomic File 영업이익성장 00,00,1,4 > Atomic Number 8 1
New Nam-Tree
Open Nam-Tree
Save Nam-Tree
Import Nam-Tree
Export Nam-Tree
Retrieve Item -> Left Term
투자심리선
Atomic
Name
Type
Value
priceTimePoint int 00
indexTimePoint int 00
Retrieve Item -> Right Term
Number
0.6
Retrieve
Wizard Item Tree
New Folder
New File
Delete
VR_20.ntr
투자심리선.ntr
스토캐스틱K.ntr
스토캐스틱D.ntr
MACD(12_26).ntr
MACD(12_26) 보조지표.ntr
재무비율
수익성
영업이익률.ntr
순이익률.ntr
ROE.ntr
ROA.ntr
ROIC.ntr
성장성
매출액성장률.ntr
영업이익성장률.ntr
순이익성장률.ntr
총자산성장률.ntr
유동자산성장률.ntr
유형자산성장률.ntr
자기자본성장률.ntr
안정성
부채비율.ntr
유동부채비율.ntr
고정부채비율.ntr
순부채비율.ntr
유동비율.ntr
당좌비율.ntr
< <
> >

2. Example – Conditional Finding

Sum of weighted value is over 3 IN

(

Disparity 5 is less than 1.05, +

Disparity 20 is less than 1.08
AND

(

VR 20 is less than 1.5
OR

Psychological Line (PL)
is over 0.6

) +

CCI 20 is less than -50

)

OR

Debt ratio is under 2.0 (200%) AND

Matches over 2 IN

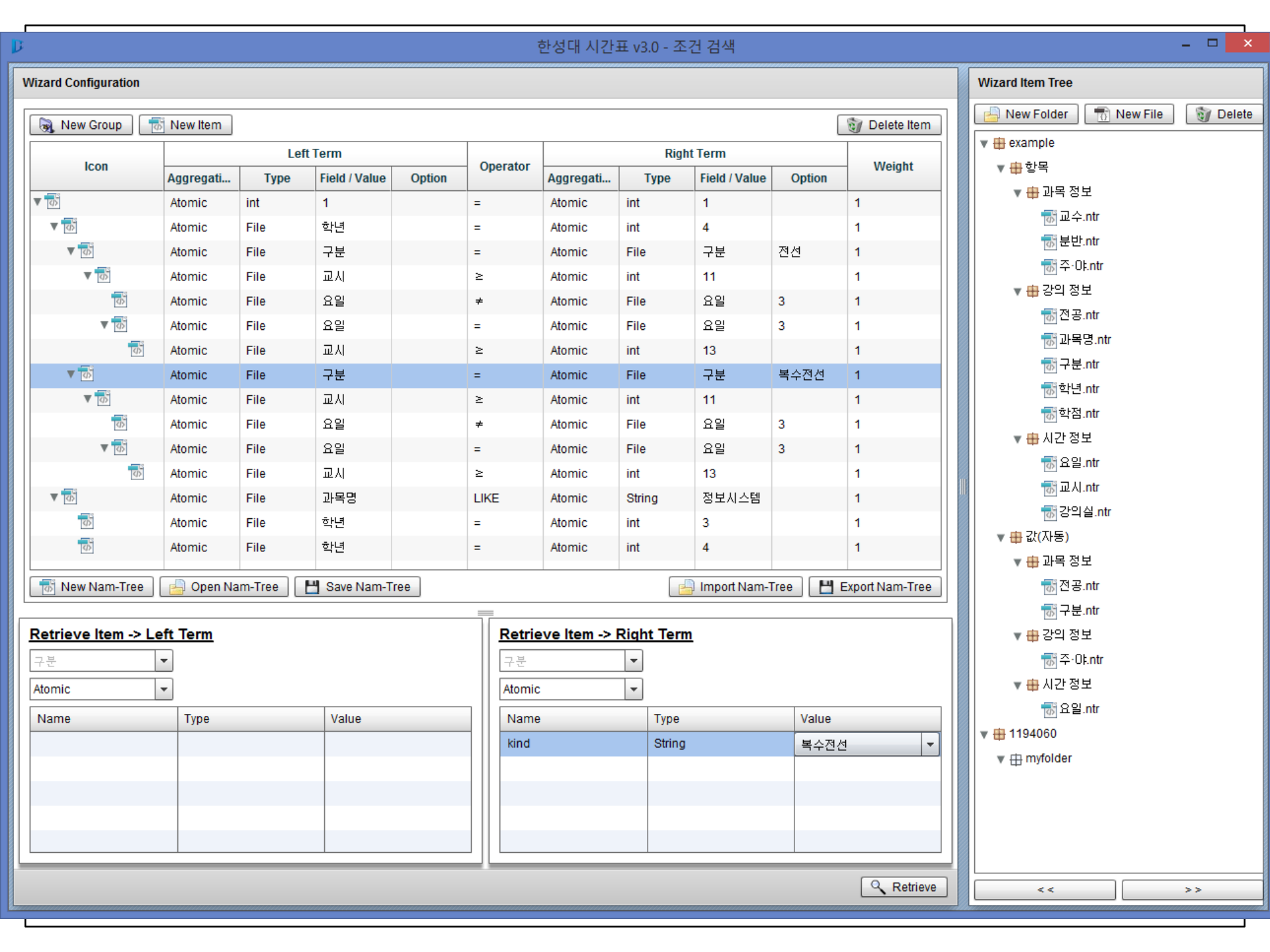
(

Liquidity ratio is over
1.0 (100%),

Quick ratio is over 0.8
(80%),

Operating profit
growth ratio is over .08
(8%)

)


한성대 시간표 v3.0 - 조건 검색
Wizard Configuration
New Group
New Item
Delete Item
Icon
Left Term
Aggregati...
Type
Field / Value
Option
Operator
Right Term
Aggregati...
Type
Field / Value
Option
Weight
Atomic int 1 = Atomic int 1 1
Atomic File 학년 = Atomic int 4 1
Atomic File 구분 = Atomic File 구분 전선 1
Atomic File 교시 ≥ Atomic int 11 1
Atomic File 요일 ≠ Atomic File 요일 3 1
Atomic File 요일 = Atomic File 요일 3 1
Atomic File 교시 ≥ Atomic int 13 1
Atomic File 구분 = Atomic File 구분 복수전선 1
Atomic File 교시 ≥ Atomic int 11 1
Atomic File 요일 ≠ Atomic File 요일 3 1
Atomic File 요일 = Atomic File 요일 3 1
Atomic File 교시 ≥ Atomic int 13 1
Atomic File 과목명 LIKE Atomic String 정보시스템 1
Atomic File 학년 = Atomic int 3 1
Atomic File 학년 = Atomic int 4 1
New Nam-Tree
Open Nam-Tree
Save Nam-Tree
Import Nam-Tree
Export Nam-Tree
Retrieve Item -> Left Term
구분
Atomic
Name
Type
Value
Retrieve Item -> Right Term
구분
Atomic
Name
Type
Value
kind
String
복수전선
Retrieve
Wizard Item Tree
New Folder
New File
Delete
example
항목
과목 정보
교수.ntr
분반.ntr
주·야.ntr
강의 정보
전공.ntr
과목명.ntr
구분.ntr
학년.ntr
학점.ntr
시간 정보
요일.ntr
교시.ntr
강의실.ntr
값(자동)
과목 정보
전공.ntr
구분.ntr
강의 정보
주·야.ntr
시간 정보
요일.ntr
1194060
myfolder
< <
> >

2. Example – Conditional Finding

Lecture of senior

Categorized in major select and its time is over 11

The day is not Wednesday or

Although Wednesday

It's ok if the time is over 13

Or, categorized in second major selection

Time's over 11

The day is not Wednesday or

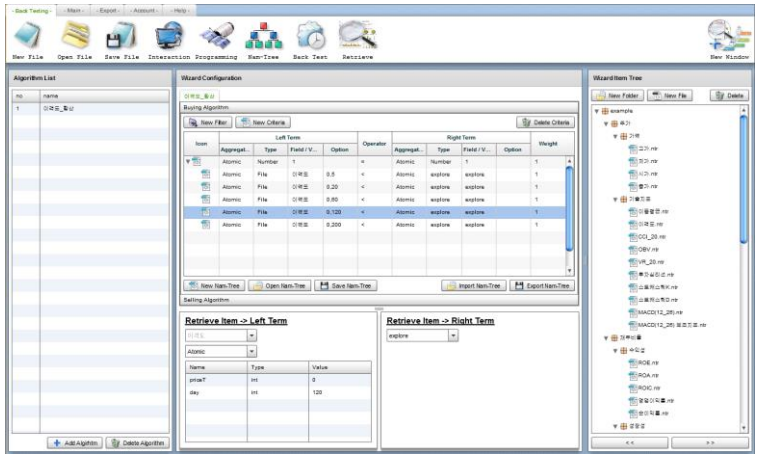
Although Wednesday

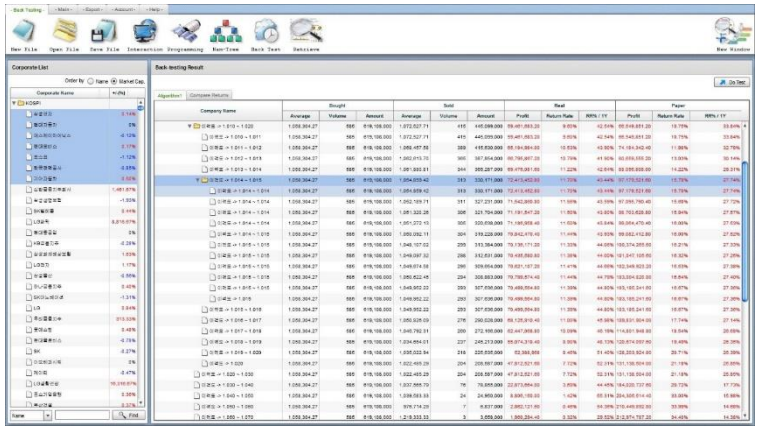
It's ok if the time is over 13

Or the subject name contains the word 'Information System'

The class for junior or senior

2. Example – Optimization

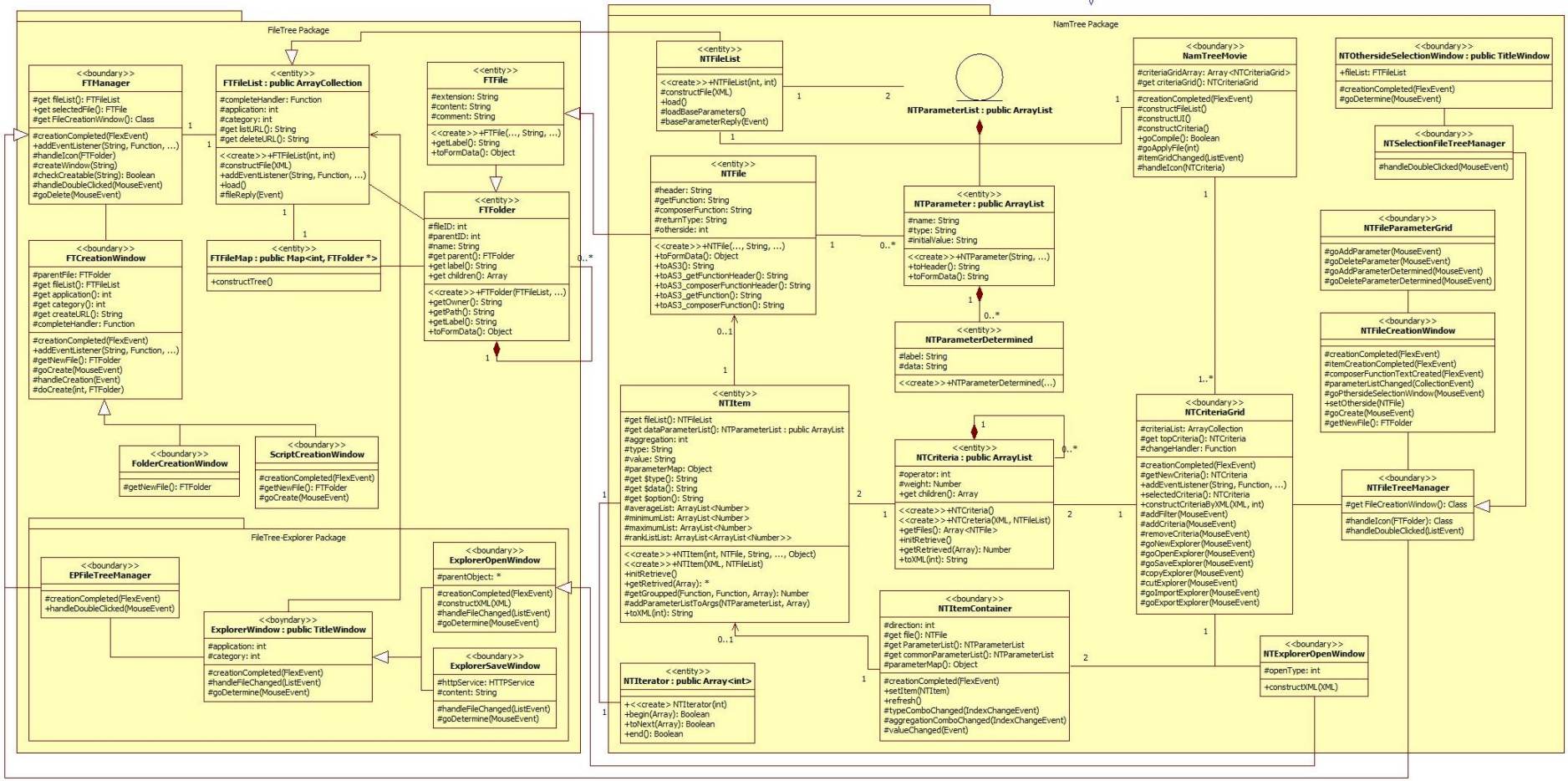


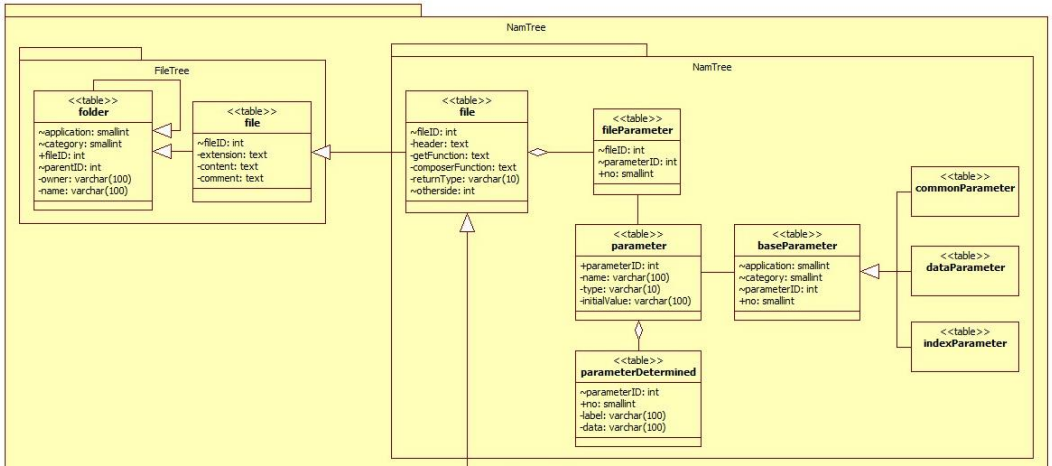


- Samchon Simulation
  (http://samchon.org)
  - Upper Menu-> Main -> Back Testing
  - Upper Menu -> Back Testing -> Nam Tree
  - Use explore

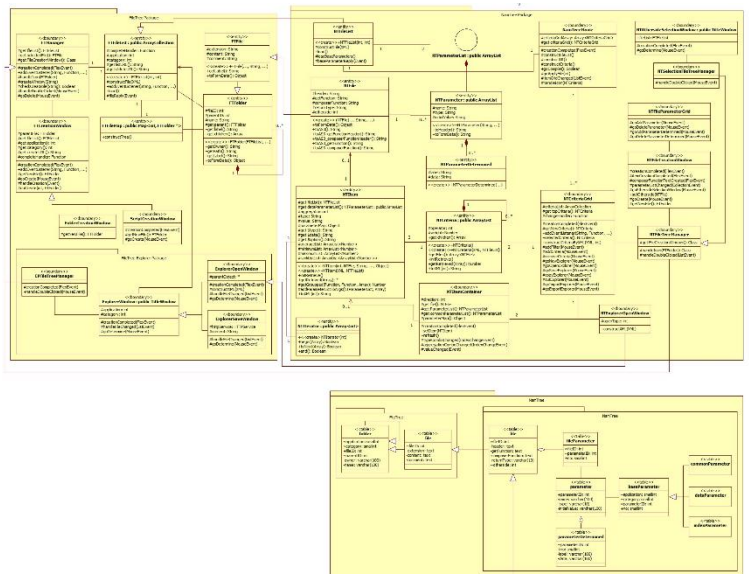
- "Explore" explores the optimal value
  - After the calculation,
    - optimal path exploration table will be constructed
  - Go back to the Nam-Tree
    - you can see that, the explore option is replaced to optimal value

- Reference the Samchon simulation manual if you want more
  - http://samchon.org/portfolio/english/portfolio.pdf


FileTree Package
<<boundary>>
FTManager
#get fileList(): FTFileList
+get selectedFile(): FTFile
+get FileCreationWindow(): Class
#creationCompleted(FlexEvent)
+addEventListener(String, Function, ...)
#handleIcon(FTFolder)
#createWindow(String)
#checkCreatable(String): Boolean
#handleDoubleClicked(MouseEvent)
#goDelete(MouseEvent)
<<entity>>
FTFileList : public ArrayCollection
#completeHandler: Function
#application: int
#category: int
#get listURL(): String
#get deleteURL(): String
<<create>>+FTFileList(int, int)
#constructFile(XML)
+addEventListener(String, Function, ...)
+load()
#fileReply(Event)
<<entity>>
FTFile
#extension: String
#content: String
#comment: String
<<create>>+FTFile(..., String, ...)
+getLabel(): String
+toFormData(): Object
<<entity>>
FTFolder
#fileID: int
#parentID: int
#name: String
#get parent(): FTFolder
+get label(): String
+get children(): Array
<<create>>+FTFolder(FTFileList, ...)
+getOwner(): String
+getPath(): String
+getLabel(): String
+toFormData(): Object
<<boundary>>
FTCreationWindow
#parentFile: FTFolder
#get fileList(): FTFileList
#get application(): int
#get category(): int
#get createURL(): String
#completeHandler: Function
#creationCompleted(FlexEvent)
+addEventListener(String, Function, ...)
#getNewFile(): FTFolder
#goCreate(MouseEvent)
#handleCreation(Event)
#doCreate(int, FTFolder)
<<entity>>
FTFileMap : public Map<int, FTFolder *>
+constructTree()
<<boundary>>
FolderCreationWindow
#getNewFile(): FTFolder
<<boundary>>
ScriptCreationWindow
#creationCompleted(FlexEvent)
#getNewFile(): FTFolder
#goCreate(MouseEvent)
FileTree-Explorer Package
<<boundary>>
EPFileTreeManager
#creationCompleted(FlexEvent)
+handleDoubleClicked(MouseEvent)
<<boyndary>>
ExplorerWindow : public TitleWindow
#application: int
#category: int
#creationCompleted(FlexEvent)
#handleFileChanged(ListEvent)
#goDetermine(MouseEvent)
<<boundary>>
ExplorerOpenWindow
#parentObject: *
#creationCompleted(FlexEvent)
#constructXML(XML)
#handleFileChanged(ListEvent)
#goDetermine(MouseEvent)
<<boundary>>
ExplorerSaveWindow
#httpService: HTTPService
#content: String
#handleFileChanged(ListEvent)
#goDetermine(MouseEvent)
NamTree Package
<<entity>>
NTFileList
<<create>>+NTFileList(int, int)
#constructFile(XML)
+load()
#loadBaseParameters()
#baseParameterReply(Event)
NTParameterList : public ArrayList
<<boundary>>
NamTreeMovie
#criteriaGridArray: Array<NTCriteriaGrid>
#get criteriaGrid(): NTCriteriaGrid
#creationCompleted(FlexEvent)
#constructFileList()
#constructUI()
#constructCriteria()
+goCompile(): Boolean
#goApplyFile(int)
#itemGridChanged(ListEvent)
#handleIcon(NTCriteria)
<<boundary>>
NTOthersideSelectionWindow : public TitleWindow
+fileList: FTFileList
#creationCompleted(FlexEvent)
#goDetermine(MouseEvent)
<<boundary>>
NTSelectionFileTreeManager
#handleDoubleClicked(MouseEvent)
<<entity>>
NTFile
#header: String
#getFunction: String
#composerFunction: String
#returnType: String
#otherside: int
<<create>>+NTFile(..., String, ...)
+toFormData(): Object
+toAS3(): String
+toAS3_getFunctionHeader(): String
+toAS3_composerFunctionHeader(): String
+toAS3_getFunction(): String
+toAS3_composerFunction(): String
<<entity>>
NTParameter : public ArrayList
#name: String
#type: String
#initialValue: String
<<create>>+NTParameter(String, ...)
+toHeader(): String
+toFormData(): String
<<entity>>
NTParameterDetermined
#label: String
#data: String
<<create>>+NTParameterDetermined(...)
<<boundary>>
NTFileParameterGrid
#goAddParameter(MouseEvent)
#goDeleteParameter(MouseEvent)
#goAddParameterDetermined(MouseEvent)
#goDeleteParameterDetermined(MouseEvent)
<<boundary>>
NTFileCreationWindow
#creationCompleted(FlexEvent)
#itemCreationCompleted(FlexEvent)
#composerFunctionTextCreated(FlexEvent)
#parameterListChanged(CollectionEvent)
#goPthersideSelectionWindow(MouseEvent)
+setOtherside(NTFile)
#goCreate(MouseEvent)
#getNewFile(): FTFolder
<<entity>>
NTItem
#get fileList(): NTFileList
#get dataParameterList(): NTParameterList : public ArrayList
#aggregation: int
#type: String
#value: String
#parameterMap: Object
#get $type(): String
#get $data(): String
#get $option(): String
#averageList: ArrayList<Number>
#minimumList: ArrayList<Number>
#maximumList: ArrayList<Number>
#rankListList: ArrayList<ArrayList<Number>>
<<create>>+NTItem(int, NTFile, String, ..., Object)
<<create>>+NTItem(XML, NTFileList)
+initRetrieve()
+getRetrived(Array): *
#getGroupped(Function, Function, Array): Number
#addParameterListToArgs(NTParameterList, Array)
+toXML(int): String
<<entity>>
NTCriteria : public ArrayList
#operator: int
#weight: Number
+get children(): Array
<<create>>+NTCriteria()
<<create>>+NTCreteria(XML, NTFileList)
+getFiles(): Array<NTFile>
+initRetrieve()
+getRetrieved(Array): Number
+toXML(int): String
<<boundary>>
NTCriteriaGrid
#criteriaList: ArrayCollection
#get topCriteria(): NTCriteria
#changeHandler: Function
#creationCompleted(FlexEvent)
#getNewCriteria(): NTCriteria
+addEventListener(String, Function, ...)
+selectedCriteria(): NTCriteria
+constructCriteriaByXML(XML, int)
#addFilter(MouseEvent)
#addCriteria(MouseEvent)
#removeCriteria(MouseEvent)
#goNewExplorer(MouseEvent)
#goOpenExplorer(MouseEvent)
#goSaveExplorer(MouseEvent)
#copyExplorer(MouseEvent)
#cutExplorer(MouseEvent)
#goImportExplorer(MouseEvent)
#goExportExplorer(MouseEvent)
<<boundary>>
NTFileTreeManager
#get FileCreationWindow(): Class
#handleIcon(FTFolder): Class
#handleDoubleClicked(ListEvent)
<<entity>>
NTIterator : public Array<int>
+<<create> NTIterator(int)
+begin(Array): Boolean
+toNext(Array): Boolean
+end(): Boolean
<<boundary>>
NTItemContainer
#direction: int
#get file(): NTFile
#get ParameterList(): NTParameterList
#get commonParameterList(): NTParameterList
#parameterMap(): Object
#creationCompleted(FlexEvent)
+setItem(NTItem)
+refresh()
#typeComboChanged(IndexChangeEvent)
#aggregationComboChanged(IndexChangeEvent)
#valueChanged(Event)
<<boundary>>
NTExplorerOpenWindow
#openType: int
+constructXML(XML)


NamTree
FileTree
<<table>>
folder
~application: smallint
~category: smallint
+fileID: int
~parentID: int
-owner: varchar(100)
-name: varchar(100)
<<table>>
file
~fileID: int
-extension: text
-content: text
-comment: text
NamTree
<<table>>
file
~fileID: int
-header: text
-getFunction: text
-composerFunction: text
-returnType: varchar(10)
~otherside: int
<<table>>
fileParameter
~fileID: int
~parameterID: int
+no: smallint
<<table>>
parameter
+parameterID: int
-name: varchar(100)
-type: varchar(10)
-initialValue: varchar(100)
<<table>>
baseParameter
~application: smallint
~category: smallint
~parameterID: int
+no: smallint
<<table>>
commonParameter
<<table>>
dataParameter
<<table>>
indexParameter
<<table>>
parameterDetermined
~parameterID: int
+no: smallint
-label: varchar(100)
-data: varchar(100)

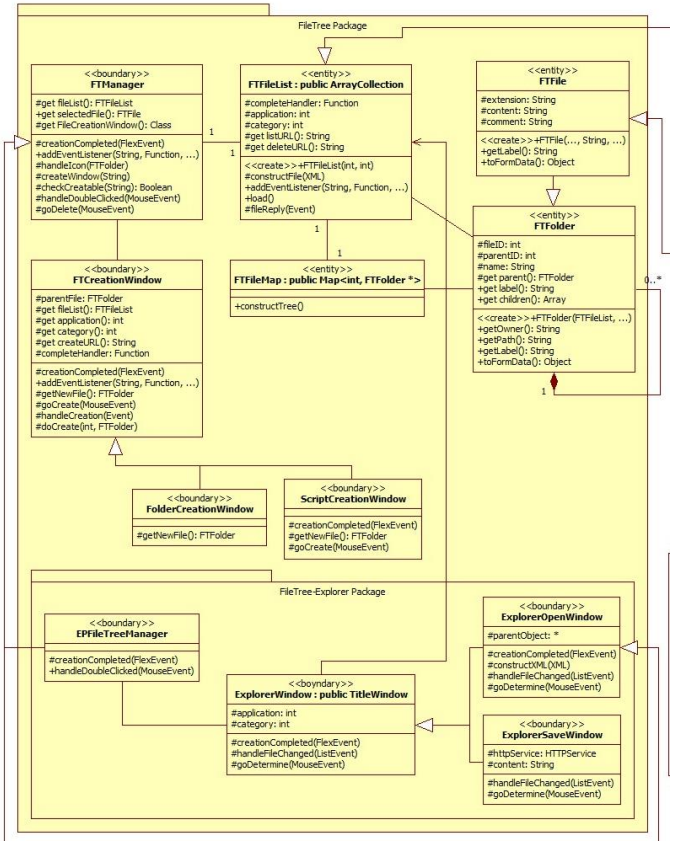
3. Architecture – Outline



- See the expanded image
  - http://samchon.org/portfolio/architecture

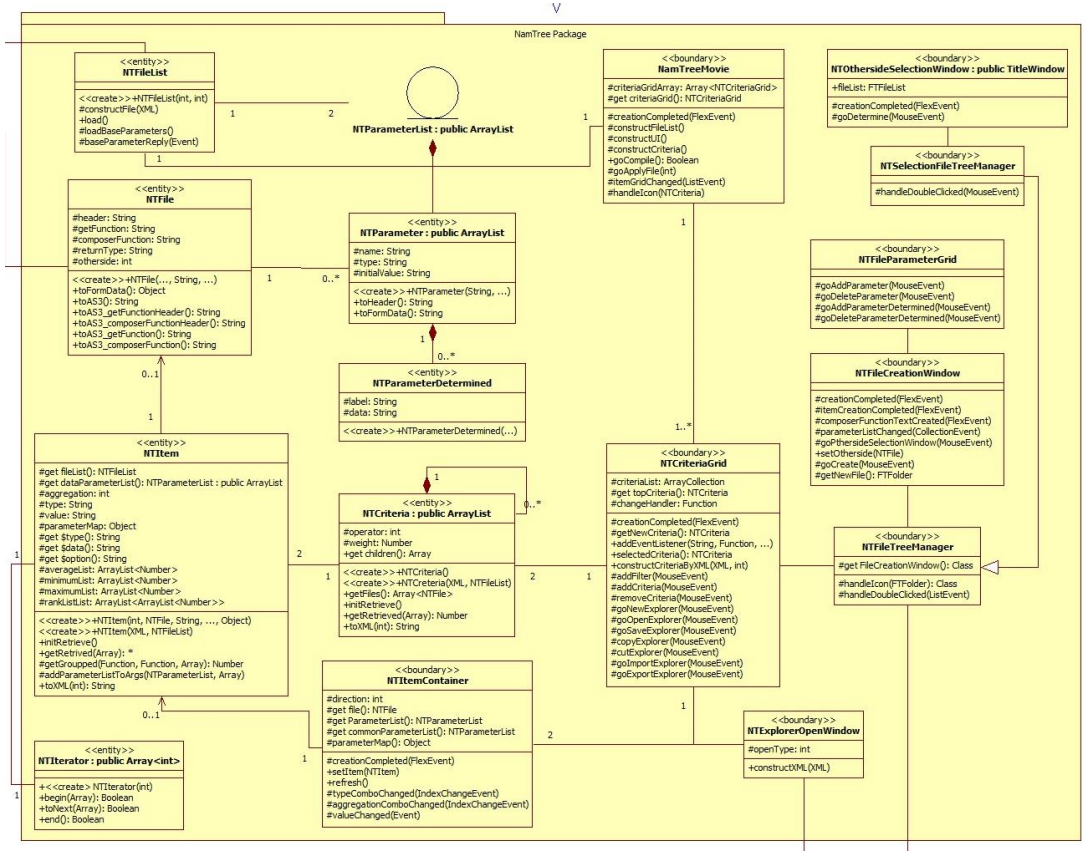
- **FileTree Package**
  - Classes for expressing file-tree structure
  - A role of superclass for function file
  - You don't have to follow this rule
    - Just my own methodology
- **NamTree Package**
  - Classes of Logical condition and its lower & dependent classes
  - The most important package
- **NamTree DB Package**
  - Just an example of storing
  - Do not handle

3. Arch. – FileTree Package



- **FTFileList** : public Vector<FTFolder>
  - Manage files and folders
  - Handle communication with DB and file-tree's composition
- **FTFileMap**
  - A member container of FTFileList
  - A map class for key (file-ID) searching
- **FTFolder**: A folder object
- **FTFile** : public FTFolder
  - A file class implemented folder
- Other boundary classes
  - These are UI classes
  - This UI-architecture is based on Flex
    - Migrate or re-design the architecture based on the language what you want to use

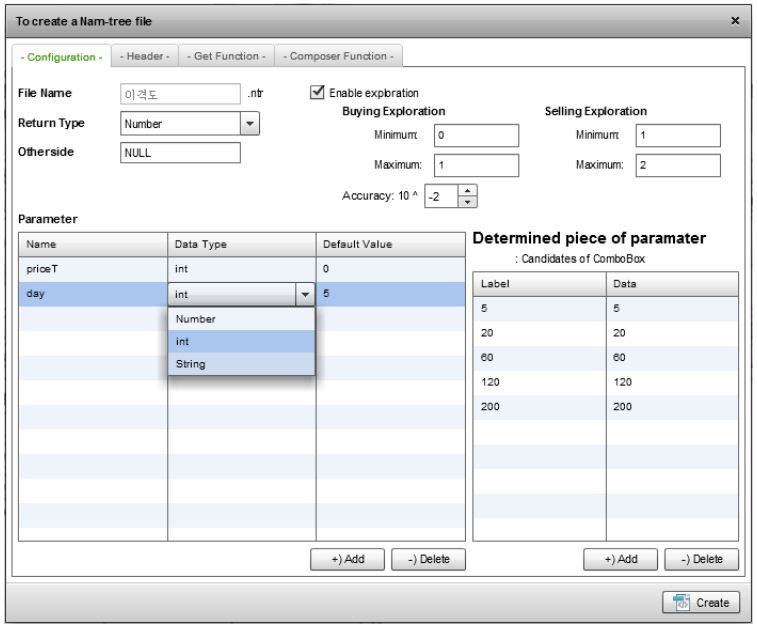
3. Arch. – NamTree Package



3. Arch. – NamTree Package

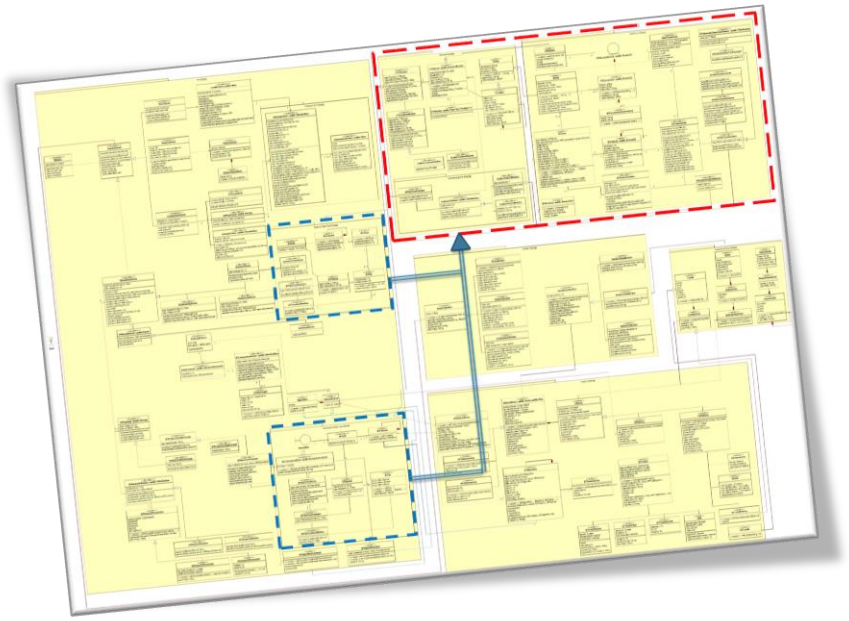
- **NTFileList** : public FTFileList
  - constrcutFile(XML) has changed
- **NTFile**: public FTFile
  - A file has function's metadata and code
    - The function can be inserted in each condition's left & right side
- **NTParameterList**
  : public Vector<NTParameter>
  - Group & Manager of NTParameter
- **NTParameter**
  : public Vector<NTParameterDetermined>
  - A parameter of user-defined function
- **NTParameterDetermined**
  - Pre-defined candidates of NTParameter
  - If this is constructed, the parent NTParameter let user to set the parameter-value by Combo-Box
- **NTCriteria**
  - It's matched on a logical condition
    - The core class of NamTree
  - Has
    - Left & Right side
    - Operator
    - Weight
- **NTSide**
  - Left & Right side of NTCriteria
  - Can has constant or function
  - Enable aggregate operation
- **NTIterator**
  - Input layer's iterator class
    - Need for iteration of aggregate operation
  - Almost same with iterator class in standard library of C++
    - ex) std::vector<T>::iterator

3. Arch. – NamTree Package



- NTCriteriaGrid
  - Hierarchical Grid for NTCriteria
- **NTItemContainer**
  - UI class expressing left & right side that is selected
  - Have to inherit not only entity or control class but also boundary class like UI class
- **NTFileCreationWindow**
  - UI class of left image
- **NTFileParameterGrid**
  - Custom, editable Datagrid to construct NTParameterDetermined

3. Arch. – Utilization



- Left image is an architecture design.
  - NTFile, NTCriteria, NTGrid, and so on
  - Inherited and utilizing them for many aspects
- Like the left case, just inherit the Nam-Tree and customize as you want
  - After the inheritance,
    - C++: virtual,
    - FLEX/AS3.0: override function
  - Customize the functions have those declarations

3. BLUEPRINT

Plans for future?

1. Add targets to optimize
2. Migration
3. Genetic Algorithm

Blueprint – To Do List

- **Add targets to optimize**
  - Parameters of user-defined function
  - Weight in each condition
- **Migration for performance**
  - C++ Migration
    - Flex/AS3.0 is mainly now
    - C++ model is implemented only to console.
      - Instead, in C++, GA model is already implemented and tested (prototype)
- **Adjusting G.A.**
  - Logical condition is objectified, so that,
    - **Coding** is possible
    - Adjust the Genetic Algorithm, creating the logics automatically, so that makes the clear **Artificial Intelligence** possible
  - Implements the selection, cross over, mutation and makes them as an Interface, so that provide the source

Q & A

Hansung Univ.
Information System Engineering
1194060 Jeongho Nam

2014.06.05