The INVEST Ontology

The INVEST ontology (IO) provides a clear structure for information that may be useful to an investment professional and articulates concepts and properties which are required as evidence for the Bayesian network**.** The ontology was developed using Protégé-OWL tool version 5.5.0 to implement the ontology in OWL, including design of the ontology and populating the ontology with instance data.

Concepts from the following ontologies were reused:

- Financial Industry Business Ontology ('FIBO') Foundations Version 1.2 [88] for the **classifier** concept and **issued equity** concept pertaining to a formal organisation.
- The SONAR financial ontology [89] for the **investment asset** concept.

1.1.1. Main Classes and Properties

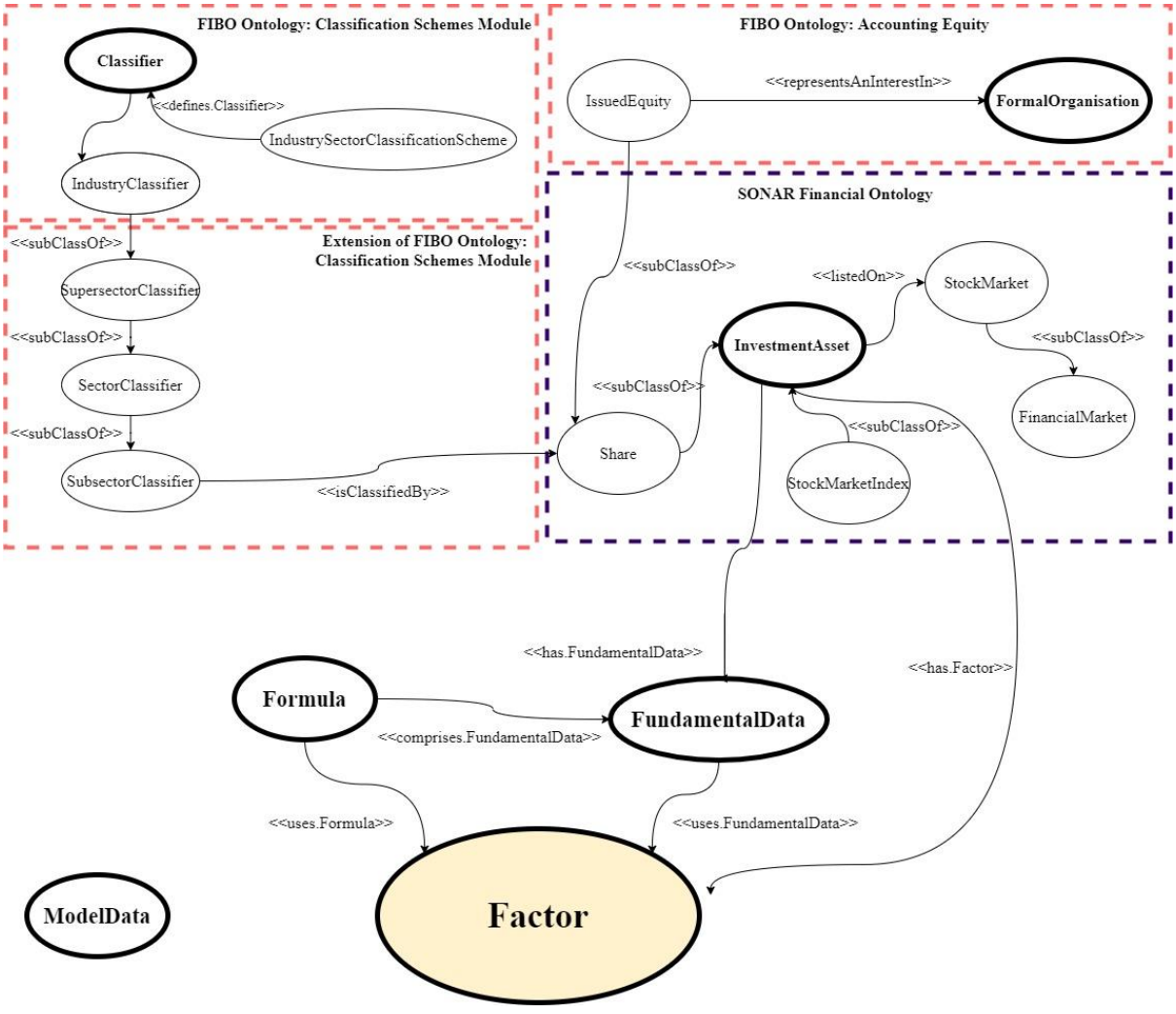


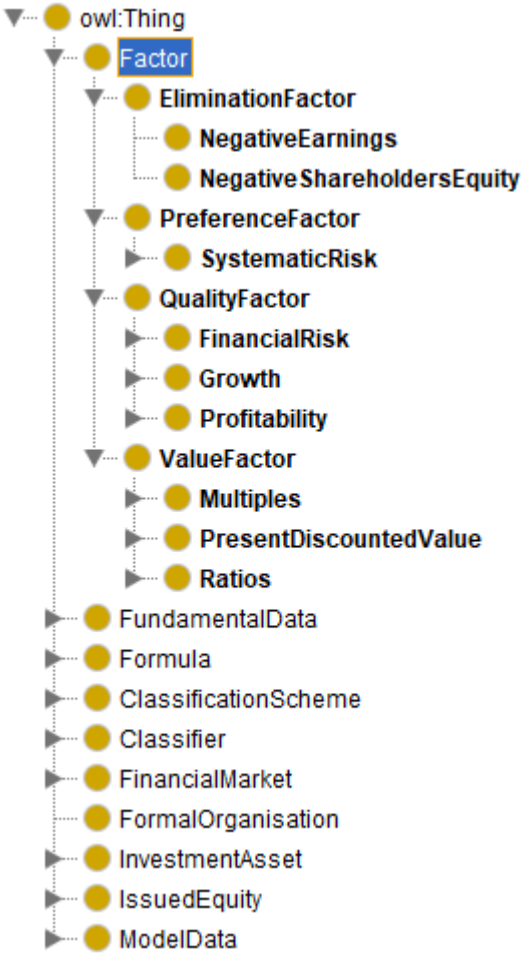
Figure 1: Overview of the key concepts, properties and relationships of the ontology

The main classes of the ontology are common to any share evaluation process; they are not confined to the value investing approach. The ontology consists of eight main classes: **Factor; Formula;**

FundamentalData; ModelData; InvestmentAsset; FormalOrganisation and **Classifier**. These determine the structure of the database for the INVEST system.

The key classes, properties and relations of the ontology are captured in Figure 7. Figure 8 below shows classes of the ontology in the Protégé ontology editor. See Appendix 1 for the complete OWL representation of the INVEST Ontology.

Figure 2: Classes of the INVEST ontology as represented in Protégé ontology editor



1.1.1.1. Factor

The **Factor** class is any ratio, figure or qualitative variable that is believed to be predictive or influential to future share performance. The factors identified and implemented are informed by research studies and the guidance of experts pertaining to the value investing approach. The **Factor** class is constructed hierarchically with sub-categories of factors represented by three sub-classes. The first level of the class hierarchy is the abstract **Factor** class and the second level includes the abstract classes: **ValueFactor**, **QualityFactor, PreferenceFactor** and **EliminationFactor**. The lower levels of the class hierarchy represent concrete observable factors. For example, **ForwardPE_CurrentvsHistory** falls under the **ValuationMultiples** sub-category and **ValueFactors** main category (see Figure 9 below). The hierarchical classes are used to represent factors as categories and sub-categories. The **evaluation objective** categorization developed in the conceptual model (see Section 4.2.2.) is implemented for the main class of the **Factor** class hierarchy. Following this, the **factor type** categorization developed in

the conceptual model (see Section 4.2.2.) is used to classify sub-categories of factors in the class hierarchy.

It is important to note that a specific factor may belong to multiple evaluation objective categories; this allows for ambiguity and the different perceptions on which evaluation objective a factor addresses. Although the categories were formulated to eliminate concept overlaps and misrepresented factors, there is still a possibility of a factor belonging to more than one category. Formalizing the model with an ontology allows for multiple association of factors with more than one class. The hierarchical representation of the **Factor** is shown in Table 14 below. The list of factor categories is not exhaustive but details factor categories specific to value investing. Literature and empirical studies would suggest there are far more factor categories that could be included. However, the intention of the model is to provide a framework for extension and to illustrate the application to the value investing approach for share evaluation.

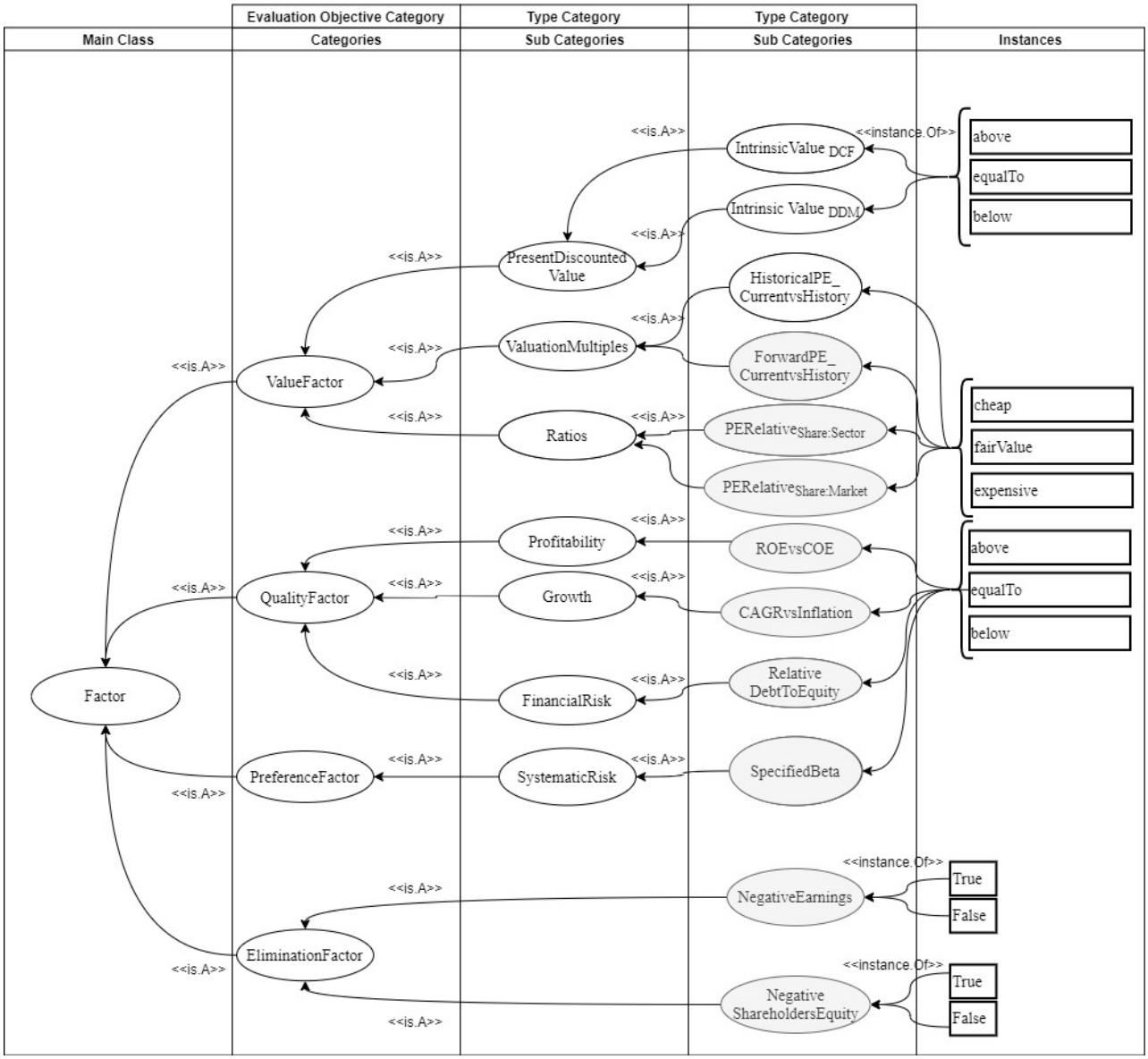
In representing the **Factor** class, a design decision was made to represent factors as the main subject of the model and not as characteristics of a share. This decision is to facilitate structuring of the factors for construction of a predictive model. This is also in line with the approaches taken by all existing categorizations (see [76]and [16]), which have been proposed by domain experts in this area.

Table 1: Sub-classes of the Factor class

Main Class	**Middle Class**	**Bottom Class**
Value Factor	Present Discounted Value	Intrinsic Value $_{DCF}$
		Intrinsic Value $_{DDM}$
	Valuation Multiples	HistoricalPE_CurrentvsHistory
		ForwardPE_CurrentvsHistory
	Ratios	PE Relative$_{Share:Market}$
		PE Relative$_{Share:Sector}$
Quality Factor	Profitability	ROEvsCOE
	Growth	CAGRvsInflation
	Financial Risk	Relative Debt to Equity
Preference Factor	Systematic Risk	SpecifiedBeta
Elimination Factor	N/A	Negative Earnings
	N/A	Negative Shareholders Equity

Figure 9 shows a conceptual diagram of the **Factor** class. It is important to note that the Conceptual Model Design (see section 4.2.2.) is encapsulated in the Factor class which incorporates the Evaluation Objective and Factor Type as sub-classes (see further detail below in 4.4.1.1. Factor). Figure 9 illustrates the class hierarchy and groupings and the types of instances which belong to the concrete classes of the Factor class. For example, the PE relative$_{Share:Market}$ has three discrete instances which indicate whether a share is “cheap”, “fairValue” or “expensive”. These discrete instances have been created through the evaluation of the current value of the PE relative$_{Share:Market}$ against a threshold; for this particular factor it would be the historical PE relative$_{Share:Market}$. Each concrete factor class will have a different set of instances depending on the threshold against which it is evaluated. The concrete classes highlighted in grey correspond to the variables used in the Bayesian networks.

Figure 3: Example of hierarchical class of the INVEST Ontology



1.1.1.2. FundamentalData

Fundamental data represents raw data that relates to a specific share and is required for share evaluation through fundamental analysis. The **FundamentalData** class has been broken down into further classes: the **FinancialStatementData** class and **MarketData** class. Fundamental data is stored in the **Database** and is updated through a data feed. Instances of the **FundamentalData** class are generally comprised of continuous data values. Fundamental data is used by the **Formula** class, and in some cases by the **Factor** class where no further processing of the data is required.

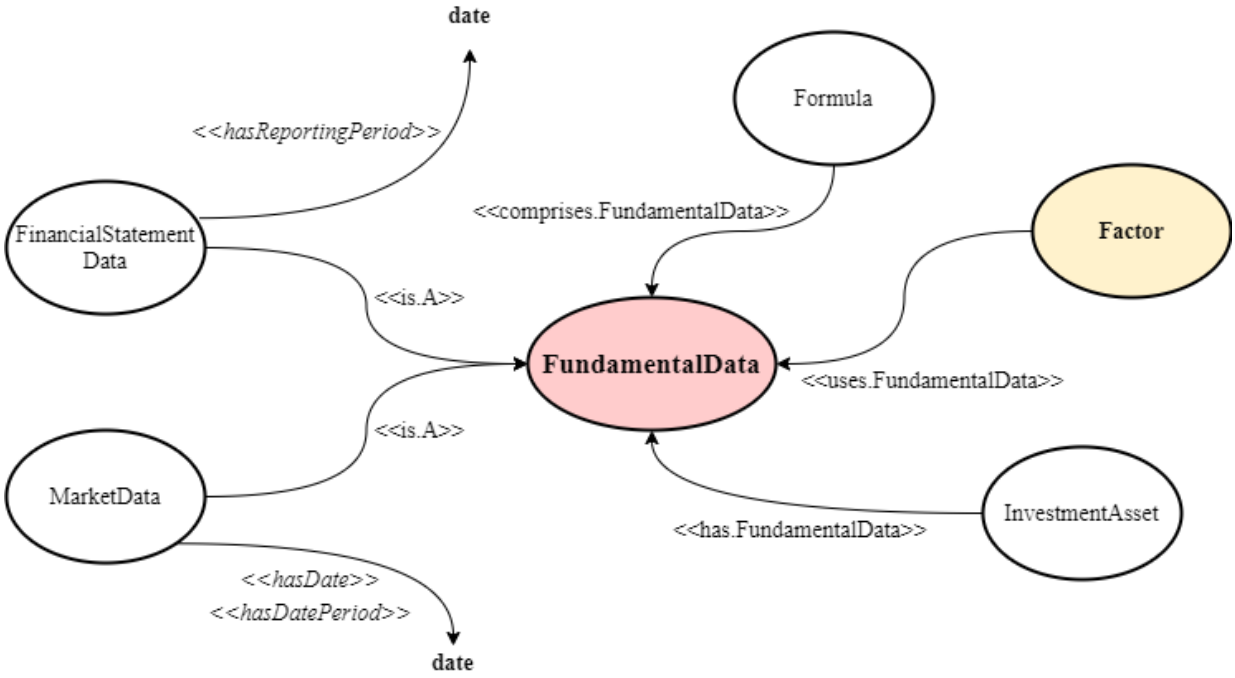
Table 2: Sub-classes of the FundamentalData class

Main Class	Middle Class	Bottom Class
Financial Statement Data	Income Statement Data	Net Income Shares in issue Earnings per share

Main Class	Middle Class	Bottom Class
	Balance Sheet Data	Total assets Total liabilities Total equity Cash and cash equivalents Total Shareholder's Equity
	Cash Flow Statement Data	Cash from operating activities Cash from financing activities
	Equity Statement Data	Retained Earnings
Market Data	Time series data	Price
		Beta
	Consensus forecasts	Forward Earnings

Data properties are associated with the **FundamentalData** class for validation of the fundamental data to be stored in the database. Where applicable, **hasDate**, **hasDatePeriod** or **hasReportingPeriod** define the date or the period within which the fundamental data is applicable. The relationship of **InvestmentAsset** class to the **FundamentalData** class is classified through the **has** relationship. Depending on the **InvestmentAsset**, there may be certain classes or instances of fundamental data which do not exist for it.

Figure 4: Extract of FundamentalData class, related properties and relationships



1.1.1.3. ModelData

ModelData represents inputs which are applicable to all shares and are used within the decision model. The **VariableParameters** class is used to represent the variable inputs and assumptions of the investor and can be manually updated to reflect the investors preference or beliefs. For example, the investor may have a certain margin of safety, expressed as a percentage, that they may want to apply when

evaluating any values in the model. This could be instituted utilizing the **MarginOfSafety** class stipulated below. The **EconomicData** class is updated as new economic figures are published.

Table 3: Sub-classes of the ModelData class

Main Class	Middle Class	Bottom Class
EconomicData	-	Inflation
	-	RiskFreeRateOfReturn
VariableParameters	Threshold	MarginOfSafety
		BetaThreshold
		MarketRateOfReturn

1.1.1.4. Formula

The fundamental data as represented by **FundamentalData** class is processed further through calculation rules stored in the **Financial Calculator** to produce calculated figures and ratios as represented by the **Formula** class. The calculation rules for these figures and ratios are specified in **Section 4.5.1.** The properties described in **FundamentalData** class are replicated for the **Formula** class.

Table 4: Sub-classes of the Formula class

Main Class	Middle Class	Bottom Class
Formula	Calculation formula	GrowthRate
		HistoricEarnings_GrowthRate
		HistoricEarnings_CAGR
		ForwardEarnings
		ForwardEarnings_CAGR
		HistoricPriceToEarnings
		ForwardPriceToEarnings
		PErelative
		Debt/Equity
		RelativeDebtToEquity
		EquityRiskPremium
		ReturnOnEquity
		CostOfEquity

An intermediate rule set is required to evaluate the calculated figures and ratios as represented by the **Formula** class against either a threshold as represented by the **Threshold** class which is a sub-class of the **ModelData** class or against another figure or ratio to produce discrete states which represent instances of the **Factor** classes. This intermediate calculation set, named the **Threshold Evaluation Component (TEC)**, is contained within the **Financial Calculator** component (see Section 4.5.2.). For example, **ForwardPriceToEarnings** may be evaluated against the average **HistoricalPriceToEarnings** for a specific share utilizing the rule set contained in **TEC**. Following this, the instances that may be observed for **ForwardPriceToEarnings** are “cheap”, “fairValue” or “expensive”. This design decision will facilitate the conversion of factors and related instances into useful primitives which can be mapped to Bayesian network variables (see Section 4.5.3.).

1.1.1.5. Classifier

A **Classifier** is a standardized classification or delineation for something, per some scheme for such delineation, within a specified context. The **IndustrySectorClassificationScheme** represents a system used for allocating classifiers to organizations by industry sector. The Financial Industry Business Ontology (FIBO) was adopted as the base ontology to describe the **Classifier** class and related **IndustrySectorClassificationScheme** class. FIBO was extended to include the **SuperSectorClassifier** class, **SectorClassifier** class and **SubsectorClassifier** class which represent standard classifications under the Global Industry Classification Standard (GICS) [90]. The GICS is the classification scheme used for share classification on the Johannesburg Stock Exchange. The **Share** class is classified by the hierarchy of **Classifier** classes.

Table 5: Sub-classes of the Classifier class and Arrangement class

FIBO Class	Sub Class	Sub Class	Sub Class	Bottom Class
Classifier	Industry Sector Classifier	Super Sector Classifier	Sector Classifier	Subsector Classifier
Arrangement	Classification Scheme	-	-	Industry Sector Classification Scheme

1.1.1.6. FormalOrganisation

The **Corporation** class is an extension of the **FormalOrganisation** class which is laid out in the base ontology, FIBO, and was included in the INVEST ontology to reflect the fact that a share represents an interest in an organisation. This is an important concept as an organisation may issue more than one type of share which forms part of issued equity.

Table 6: Sub-class of the FormalOrganisation class

FIBO Class	Sub Class
FormalOrganisation	Corporation

1.1.1.7. InvestmentAsset

An **InvestmentAsset** represents everything of value which a financial intermediary can invest in, such as stock market index funds and shares. The financial ontology as developed in SONAR was adopted as the base ontology for describing the abstract **InvestmentAsset** class and more specifically the concrete **Share** class. This class was included to incorporate pertinent information required for share identification and classification.

Table 7: Sub-class of the InvestmentAsset class

Main Class	Bottom Class
InvestmentAsset	Share

	Stock Market Index Fund

The following data properties are associated with the **InvestmentAsset** class: **hasName** and **hasCode** which capture the name and code for the specific investment asset. For certain classes such as **Share** and **StockMarketIndexFund** that may be listed further properties are defined like **listedOn** which describes the stock market the investment asset is listed on.