Glossary

- **Abstraction** is the process of extracting goals from different sources. It facilitates the organization of detail-intensive requirements information so that 'concrete phenomena' may be described in a more abstract manner. Stage [80] states that in some cases a high level of abstraction facilitates determining points of resemblance among similar situations.

- **Agents** are responsible for the completion and/or satisfaction of goals within an organization or system. For example, given an Electronic Meeting System (EMS), a meeting initiator is the agent responsible for calling, or initiating, a meeting. Given a course registration system, a student enrolling for a course is the agent responsible for registering for that course.

- **Constraints** place a condition on the achievement of a goal.

- An **Enterprise** is the business or organization for which the proposed system is intended.

- **Exceptions** are special or extraordinary circumstances that occur in the system.

- **Goals** are targets for achievement which provide a framework for the desired system. Goals are high level objectives of the business, organization, or system. They express the rationale for proposed systems and guide decisions at various levels within the enterprise. `Corporate profits maximized` is an example of a high-level enterprise goal. The two primary types of goals discussed in this thesis are achievement and maintenance goals.

  - *Achievement goals* are objectives of an enterprise or system. For example, a university course registration system may need to satisfy the goal of enrolling students in courses before the first day of class each semester. The object of the goal, identified by the stakeholders as the primary purpose of the system, is course registration. In general, achievement goals may be mapped to functional requirements.

  - *Maintenance goals* are those goals which are satisfied while their target condition remains constant or true. They tend to be operationalized as actions or constraints that prevent certain states from being reached. In general, maintenance goals map to nonfunctional requirements.

- **Goal dependency** relations exist between pairs of goals [60].

  - A *precedence* relation between goals G_1 and G_2, where goal G_1 must be completed before goal G_2 is expressed $G_1 < G_2$.

  - A *contract* relation between goals G_1 and G_2, where goal G_2 must be completed if goal G_1 occurs is expressed $G_1 \rightarrow G_2$.

- **Goal obstacles** prevent or block the achievement of a given goal [60]. Abstracting and identifying goal obstacles allows the consideration of possible methods of failure and the anticipation of exceptional cases.

- **Goal refinement** is the process of subdividing a set of goals into a logical subgrouping so that system requirements may be more easily understood, defined, and specified. This logical subgrouping consists of subgoals which must be met in order for the parent goal to be met.

- **Goal schemas** are models which specify the relationships between goals and agents in terms of events that cause a change of state. In a goal schema, goals are specified as events in terms of pre- and post-conditions.

- **Operationalization** is the process of defining a goal with enough detail so that its subgoals have an operational definition.

- A **Precondition** is a milestone or intermediate goal which must be achieved before another (dependent) goal may be achieved. Precedence relations suggest candidate preconditions.

- A **Requirement** specifies how a goal should be accomplished by a proposed system. Requirements define the capabilities that a system must provide in order to satisfy the goals of stakeholders. An example of an operation requirement for the goal `Corporate profits maximized` is: `Customer calls should be handled in less than 5 minutes` with the theory that you can improve the cost/benefit to the corporation.

  - *Functional requirements* describe the behavioral aspects of a system.

  - *Nonfunctional requirements* describe the nonbehavioral aspects of a system, capturing the properties and constraints under which a system must operate.

- **Scenarios** are behavioral descriptions of a system and its environment arising from restricted situations. They capture combinations of actions, exemplify behaviors and enable hidden needs to be uncovered. Scenarios are also useful for evaluating design alternatives and validating designs.

- A **Stakeholder** is anyone who claims an interest in a given enterprise or system. Stakeholders are those individuals who can share information regarding the proposed system, its implementation, or the problem domain.