1 Notation

The grammar for the Lando System Specification Sublanguage is written in the EBNF notation. The main elements of the notation that we utilize are:

- Terminals are represented with double or single quotes; e.g. "explanation".
- Optional bits are represented with squared brackets; e.g. ["explanation" text].
- Repetition is represented with curly braces; e.g. { identifier }
- We use a slightly enhanced notation $\{a\}^+$ to indicate *non-zero* repetitions. This is simply equivalent to: $a\{a\}$.
- For defining terminals, we use the EBNF special form to declare an extended regular expression for example: ?/\w+/?

2 Grammar

```
lando-source
                  ::=
                         { spec-element }
                                                                                        (Lando source)
   spec-element
                         informal-chart | class-dictionary
                                                                               (Specification Elements)
 informal-chart
                        system-chart | cluster-chart | class-chart
                         event-chart | scenario-chart | requirements-chart
                                                                                     (Informal Charts)
                         creation-chart
                         "dictionary" name { dictionary-entry } + "end"
class-dictionary
                                                                                    (Class Dictionary)
                         "class" name "description" text
                                                                              (Class Dictionary Entry)
dictionary-entry
   system-chart
                  ::=
                         "system_chart" name
                         ["indexing" indexing]
                         ["explanation" text]
                         { cluster-entry }
                                                                                        (System Chart)
                                                                                        (Cluster Entry)
   cluster-entry
                         "cluster" name "description" text
   cluster-chart
                         "cluster_chart" name
                  ::=
                          "indexing" indexing ]
                          ["explanation" 	ext{text}]
                          class-entry }
                         { cluster-entry }
                         "end"
                                                                                       (Cluster Chart)
                         "class" name "description" text
                                                                                         (Class Entry)
     class-entry
     class-chart
                  ::=
                         "class_chart" name
                         ["indexing" indexing]
                          ["explanation" \operatorname{text} ]
                          ["inherit" name-list
```

```
["query" string-list]
                         ["command" string-list]
                        ["constraint" string-list]
                        "end"
                                                                                         (Class Chart)
                        "event_chart" name
   event-chart
                 ::=
                        [event-direction]
                         "indexing" indexing ]
                         "explanation" text ]
                        { event-entry }
                        "end"
                                                                                         (Class Chart)
                        "event" string "involves" name-list
                                                                                         (Event Entry)
   event-entry
                        "incoming" | "outgoing"
                                                                                     (Event Direction)
event-direction
scenario-chart
                        "scenario_chart" name
                 ::=
                         ["indexing" indexing]
                         ["explanation" text]
                        { scenario-entry }
                        "end"
                                                                                      (Scenario Chart)
scenario-entry
                        "scenario" name "description" text
                                                                                      (Scenario Entry)
creation-chart
                 ::=
                        "creation_chart" name
                         ["indexing" indexing]
                         ["explanation" 	ext{text}]
                        { creation-entry }
                        "end"
                                                                                       (creation Chart)
creation-entry
                        "creator" name "creates" name-list
                                                                                       (creation Entry)
                        { identifier ':' string-list }
                                                                                           (Index List)
      indexing
                 ::=
                       identifier
                                                                                               (Name)
         name
                 ::=
                       \mathsf{name}~\{~\text{','}~\mathsf{name}~\}
     name-list
                                                                                           (Name List)
                        1"1 ? /.*/? 1"1
                                                                                               (String)
         string
     string-list
                       string { ', 'string }
                                                                                          (String List)
                       string { '\n' string }
                                                                                      (Multi-line Text)
          text
                 ::=
     identifier
                                                                                            (Identifier)
                 ::=
```