2. vseh podrobnosti v zvezi s podatkovno bazo;

3. konverzijskega postopka, kako in odkod se bo napolnila podatkovna baza;

4. podrobne sistemske logike in zlasti vključitev vseh kontrol, kot so:

   - kontrola podatkov na numeričnost, dolžino in območje vrednosti,

   - kontrola podatkov v odvisnosti od vrednosti določenih drugih podatkov,

   - kontrole ključev, opredeljene logike, datumov in druge kontrole;

5. zanesljivosti sistema, kot so pretočnost podatkov, odzivni časi,...;

Vse zgoraj naštete potrebe v bistvu sodijo v podatkovni slovar. Vsekakor pa je treba oceniti, kakšen je vpliv teh zahtev na prototip. Znano je dejstvo, da programska orodja četrte in pete generacije sicer nudijo marsikatere prednosti, a so na računalniškem prostoru in času precej potratna. V kolikor analiza in ovrednotenje zmogljivosti pokaže, da le-te ne zadoščajo, nadaljujemo z ostalimi fazami življenjskega ciklusa razvoja obdelave na običajen način in z običajnimi programskimi orodji. V strokovni literaturi je opisanih nekaj primerov, kjer so začeli uporabljati prototip kot redno obdelavo, ne da bi fizično razrešili vrsto zgornjih vprašanj; posledice so bile katastrofalne.

Logične strukture, ki so primerne za izdelavo prototipov v smislu sistemsko analitičnih orodij, so strukturirani problemi s precejšnjo množico objektov in atributov v datotečnih povezavah, toda z relativno majhnim deležem algoritmičnih procesov. Dobri kandidati so torej transakcijski sistemi operativne narave. Obdelave, namenjene odločevalcem, niso primerne. Na splošno pa velja, da paketne (batch) obdelave in celotne informacijske sisteme ne smemo razvijati s prototipnim pristopom, kar bomo razložili pozneje.

4.2. Predelava prototipa v obdelavo za redno uporabo

Predelava prototipa v obdelavo za redno uporabo je dokaj pogosta, toda po našem mnenju v bistvu ne prinaša večjih kakovostnih razlik glede na način uporabe, ki je opisan v podpoglavju 4.1. Značilno za oba pristopa je, da je