

To the Graduate Council:

I am submitting herewith a thesis written by Pamela M. Mason entitled "A Micro Computer Data Base Developed for Management of Radoactive Waste Data." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.

Maria Z. Leech

Maria Z. Leech, Major Professor

We have read this thesis
and recommend its acceptance:

V. Feag
RP Lelino

Accepted for the Council:

Cumminel

The Graduate School

**A MICRO COMPUTER DATA BASE DEVELOPED FOR
MANAGEMENT OF RADIOACTIVE WASTE DATA**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Pamela M. Mason

August 1984

ABSTRACT

The purpose of this thesis project was to design an interactive data base system on a micro computer for management of the data on radioactive waste stored at the Oak Ridge National Laboratory. The system was designed to provide managers user-friendly, menu-driven and ad hoc methods of entering and retrieving data. The system was then designed using Micro Data Base Systems III (MDBS III). The primary result was the development of a micro computer data base containing the specified data which is easily and interactively retrievable. The significance of this work was the design and implementation of an extended-network data base on a micro computer. This system replaces an antiquated method of data processing and provides the users with a state-of-the-art method of information management.

TABLE OF CONTENTS

CHAPTER	PAGE
I INTRODUCTION	1
1.1 Background	1
1.2 Purpose	4
1.3 Approach	4
2 THE APPLICATION ENVIRONMENT: THE PROBLEM IDENTIFIED	6
2.1 The Previous Data Processing System for the Radioactive Waste Stored at the Oak Ridge National Laboratory	6
2.1.1 Overview of the data storage and retrieval methods	6
2.1.2 Disadvantages of the data processing system	11
3 THE DEVELOPMENT OF A NEW SYSTEM	13
3.1 Study of Data Base Management Models	13
3.1.1 Definition and objectives of a data base	13
3.1.2 Data base management model types	16
3.1.2.1 Hierarchical data base model	16
3.1.2.2 Relational data base model	17
3.1.2.3 Network data base model	21
3.2 The Choice of a Micro Computer Based System	25
3.2.1 The advantages of a micro computer system	25
3.2.2 The choice of MDBS III for implementation	28
3.2.2.1 MDBS III compared to dBASE II	28
3.2.2.2 MDBS III features	29
3.3 Design of the New Data Base	41
3.3.1 Analysis of the data	41
3.3.1.1 Definition of the data items	41
3.3.1.2 Definition of relationships	42
3.3.2 Definition of users' requirements	42
3.3.3 Design of the data base to meet the users' requirements	47

CHAPTER	PAGE
3 continued	
3.4 Implementation of the New Data Base	57
4 RESULTS AND CONCLUSIONS.....	58
4.1 Results of the Project	58
4.2 Conclusions.....	60
4.3 Recommendations.....	61
LIST OF REFERENCES.....	63
APPENDIXES.....	66
Appendix A	67
Appendix B	72
VITA.....	75

LIST OF TABLES

TABLE		PAGE
3-1.	Comparison summary between MDBS III and dBASE II	30
4-1.	Comparison of the new data base management system to the previous data processing system	59

LIST OF FIGURES

FIGURE	PAGE
2-1. Form filled out by Chemical Technology personnel for waste disposed at ORNL	7
2-2. Example of completed form shown in Figure 2-1.....	8
2-3. Input sheet for waste information to be entered into data file	9
2-4. Example of completed form shown in Figure 2-3	10
3-1. Sample data in hierarchical form	18
3-2. Sample data in a relational table	20
3-3. Example of a relation with a two attribute key, BUILDING + STORDATE	22
3-4. Sample data in a network model	23
3-5. Example of M:N information in a network set	24
3-6. Conceptual view of two 1:M sets in a network system	26
3-7. Conceptual view of a M:N network set	27
3-8. The architecture of data base management systems.....	32
3-9. Sample DDL program.....	33
3-10. Conceptual view of the data base defined in Figure 3-9	34
3-11. Three example DML commands as they would be used in a BASIC application program	36
3-12. IDML counterparts to the DML commands shown in Figure 3-11	37
3-13. Example QRS query	39
3-14. Most needed 1:M relationship information	43

FIGURE		PAGE
3-15.	Most needed M:N relationship information	44
3-16.	Most needed 1:1 relationship information	45
3-17.	The development of the DOCUMENT record type	46
3-18.	Final data base schema	48
3-19.	Incorrect initial schema designed to represent an isotope can be stored many times	50
3-20.	Incorrect schema to represent the M:N relationship between ISOT and DOCUMENT	53
3-21.	Correct schema using intersecting data	54
3-22.	QRS command to obtain the total quantity of PU-239 stored	55
3-23.	QRS command to obtain isotopic makeup of a particular document	56

CHAPTER I

INTRODUCTION

1.1 Background

During World War II, Oak Ridge was selected as a site to fabricate weapons-grade uranium under the classified control of the Manhattan Project (ORNL/CF-82/202 1982, p.2). A necessary result of this activity was the production of materials which were radioactive or contaminated with radioactivity. Some of these materials were intentionally radioactive, such as that found in weapon components, and were used for that purpose. Other materials such as gloves or tools became contaminated with radioactivity because they were used to handle radioactive material. Also, chemical residues, which became radioactive during processing, had to be segregated from possible human contact and disposed of.

Since World War II, new, peaceful uses of radioactive materials have been discovered. Commercial involvement includes the production of electrical power in nuclear power plants. During this production, uranium fuel slowly breaks down into radioactive products to a point where the fission process is no longer efficient for power production. This spent fuel, which is removed and stored in containers, is still highly radioactive and must also be properly disposed (DOE/EIS 1980, p.1.1). Other beneficial uses of radioactive materials include the application of radioactive implants in cancer patients and diagnostic treatment, but these also result in materials that must be discarded in an appropriate manner. The term "radioactive waste" is a classification of these radioactive materials which have served their useful purpose (Jacobs 1984).

Radioactive waste must be stored in a systematic, well-defined manner. Safe storage of the waste is necessary for reasons of reducing health risks to workers and the general public. Radioactive waste may also be of economic value. These wastes may have future uses, so it is necessary that they be easily located and retrieved. The waste is also stored because of concern over the proliferation of nuclear weapons (DOE/EIS 1980, p.1.1). Because of these types of concerns, the Department of Energy (DOE) accepted the responsibility of developing technologies for the management and disposal of radioactive wastes (DOE/EIS 1980, p.1.1).

Radioactive wastes have been divided into four classifications. One classification is spent fuel which is removed from commercial nuclear power plants. This fuel may be stored, reprocessed for recycle of uranium or plutonium, or disposed. The second category is called transuranic (TRU) waste. TRU wastes are those wastes that at the end of an institutional control period are contaminated with alpha-emitting transuranium radionuclides with half-lives greater than 20 years and concentrations greater than 100 nCi/g (DOE Order 5820 1983, p.5). High-level waste, the third classification, is defined as any highly radioactive waste produced directly from the reprocessing of spent fuel (Public Law No. 97-425). Other waste forms which contain neither TRU nor high-level wastes are classed as low-level waste.

Throughout the United States, there are several facilities which have been established as storage facilities for radioactive waste. Some facilities may only store low-level wastes, while others could possibly store all four types of wastes. According to Nuclear Regulatory Commission regulations for commercial producers of waste (Title 10, CFR, Part 61) and DOE orders for

wastes generated by DOE and its prime contractors (DOE Order 5820 1983), the party which generates waste may ship it to an indicated facility for storage (ORNL/NFW-84/10 1984, p.1.1). It is the responsibility of these individual storage sites to develop methods of storage and waste management (DOE Order 5820 1983).

The Oak Ridge National Laboratory (ORNL) serves as one such storage site. Locating this site at ORNL was a logical consequence of World War II radioactive waste activities. This facility accommodates the storage of TRU and low-level wastes. The Chemical Technology Division at ORNL is responsible for both the storage of the waste and the management of the storage information. This information is of extreme importance. The public, through political and regulatory means, demands not only that the waste be stored safely, but also that records be kept which indicate how much waste of what type has been stored at a given site. Records must indicate the storage locations and how the waste was stored within a facility, who generated the waste, and what form it is in. These requirements dictate significant responsibilities for information management personnel.

Most recently, the management of this data at ORNL was performed by a second party delegated by the Chemical Technology Division. The data were maintained in a sequential, flat file on a large-scale computer system and were accessed via COBOL programs. Maintenance was performed either by executing specialized COBOL programs or by editing of the data file. This was once an accepted, "efficient" method of data processing, especially when compared to manually maintaining the information. Technology has been developed, however, which now allows managers to make inquiries and obtain answers much more

quickly than previous systems allowed. This developing technology and the desire of program managers to implement a more efficient information system were crucial elements in the initiation of the project reported here.

1.2 Purpose

The purpose of this thesis is to design and implement an interactive data base system on a micro computer for information management of radioactive waste stored at ORNL. A primary goal of the system is to provide managers user-friendly, menu-driven, and ad hoc methods of entering, updating and retrieving this data.

1.3 Approach

The approach taken involves an analysis of available data base management systems, the data, and the users' requirements. The Micro Data Base Systems III (MDBS III) (vended by Micro Database Systems, Inc.) is being used. This system meets the users' needs through an extended-network data base model.

The overall implementation of the system involves several steps. At first, data is downloaded from a large computer to a hard disk of an IBM Personal Computer (IBM-PC). This data is in a sequential, flat file. Next, a program formats the raw data into a form acceptable for input into the MDBS III data base. A second program then loads the formatted data into the data base. Once the data base is loaded, it is accessible via an interactive query language or data manipulation language, as well as through menu-driven programs. A menu-driven input program is also available to facilitate additions to the data base.

The remainder of this thesis is organized as follows. Chapter 2 gives an overview of the previous data processing system, emphasizing the disadvantages of this system. Chapter 3 begins with an overview of data base management models and the terminology and language used in data base operations, and follows with a discussion of the MDBS III software that was chosen to implement the project. The procedures used in designing the data base and its implementation are also discussed in Chapter 3. This section includes a definition of the data, the user requirements, and the design of the new system to meet these needs. A description of the data base implementation phase completes Chapter 3. Chapter 4 summarizes the results of the implementation and application of the new radioactive waste management information system and evaluates its performance. This is an evaluation by comparison of the newly designed data base system to the previous filing system with a look at time constraints, flexibility, and ease of use. The last section of this chapter discusses recommendations for further development for both the implemented system and data base systems for micro computers in general.

CHAPTER 2

THE APPLICATION ENVIRONMENT: THE PROBLEM IDENTIFIED

2.1 The Previous Data Processing System for the Radioactive Waste Stored at the Oak Ridge National Laboratory

2.1.1 Overview of the data storage and retrieval methods. ORNL is one of several sites throughout the nation which serves as a storage facility for radioactive waste. The Chemical Technology Division (CTD) is responsible for the physical storage of the waste. The information system associated with this storage was maintained by another party.

The previous process of recording this information involved several steps. The Chemical Technology Division would receive waste to be stored. As this waste was being stored, there was a form, Figure 2-1, which the CTD personnel completed. This form described the chemical makeup of the waste, the storage date, the storage facility where the waste was stored, the type of waste, and any other required information. An example of a completed form is shown in Figure 2-2. Completion of this form was the first step in the data recording process.

After this form was completed, it was sent to various personnel to be filed as a hardcopy. The CTD personnel also used it to enter the data onto a second form shown in Figure 2-3. This was the second step in the process of recording the information. Figure 2-4 is an example of the input sheet containing the data taken from Figure 2-2. This input sheet was the information that was to be keyed into the data file.

REQUESTER: EXECUTES THIS SECTION BEFORE ARRANGING MATERIAL TRANSFER

Figure 2-1. Form filled out by Chemical Technology personnel for waste disposed at ORNL.

REQUEST FOR STORAGE OR DISPOSAL OF RADIOACTIVE SOLID WASTE OR SPECIAL MATERIALS

DATE 4/19/77		AMOUNT OF WASTE (BLOS, IF ORNL) 301.9		REQUESTER'S SIGNATURE <i>Carmela Y.M. Mason</i>		EMPLOYEE NO. 158-24-9576		PHONE NO. 488-1234		OFFICE AND BLDG. NO. 1		CHARGE NO. 10932	
TOTAL VOLUME (CU. FT.) 9		COMBUSTIBLE VOL. (CU. FT.) 2		WEIGHT (LBS.) 75		ACCOUNTABILITY (IHM NUMBER)		TOTAL CUMES IN WASTE (BEST ESTIMATE)					

WASTE CLASSIFICATION (CHECK ONE)	TYPE OF WASTE (CHECK ONE)	CONTAINER(S) IDENTIFICATION (INDICATE NUMBER OF EACH)
0. ☐ TEMPORARY STORAGE 1. ☒ TRU OR 233U (> 100 µCi/Kg) - Retrievable Storage 2. ☐ URANIUM/THORIUM 3. ☐ FISSION PRODUCT 4. ☐ INDUCED ACTIVITY 5. ☐ TRITIUM 6. ☐ BETA-GAMMA TRU OR 233U (> 100 µCi/Kg) - Retrievable Storage 7. ☐ ALPHA TRU OR 233U (< 100 µCi/Kg) BETA-GAMMA TRU OR 233U (< 100 µCi/Kg) 8. ☐ OTHER _____ 9. ☐ LANDFILL	1. ☐ BIOLOGICAL (BW) 2. ☐ CONTAMINATED EQUIPT. (CE) 3. ☐ DECONTAMINATION DEBRIS (DD) 4. ☒ DRY SOLIDS (DS) 5. ☐ SOLIDIFIED SLUDGE (SS) 6. ☐ NOT CLASSIFIED (NC) BRIEF DESCRIPTION: _____ _____	1. ☒ 55 GAL SS DRUM 2. ☐ 30 GAL SS DRUM 3. ☐ 4 1/2" WALL CONCRETE CASK 4. ☐ 8 IN. WALL CONCRETE CASK 5. ☐ 12 IN. WALL CONCRETE CASK 6. ☐ 55 GAL BI DRUM 7. ☐ 30 GAL BI DRUM 8. ☐ WOOD 11. ☐ PLASTIC 9. ☐ OTHER 12. ☐ DUMPSTER 10. ☐ GI CAN 13. ☐ NONE 14. ☐ CASK NO. _____ WALL THICKNESS _____ IN. SHIELDING MATERIAL _____ METAL BOX 15. ☐ 74.5" L x 50.5" W x 38.5" H 16. ☐ 68" L x 54" W x 38.5" H 17. ☐ 88" L x 54" W x 54" H 18. ☐ 112" L x 68" W x 77" H

PRINCIPAL ISOTOPE(S): (BEST ESTIMATE)			
1. QUANTITY <u>3.00</u> ☒ GRAM(S) <u>11-233</u> ☐ CURIE(S); IDENTITY	2. QUANTITY _____ ☐ GRAM(S) _____ ☐ CURIE(S); IDENTITY		
3. QUANTITY _____ ☐ GRAM(S) _____ ☐ CURIE(S); IDENTITY	4. QUANTITY _____ ☐ GRAM(S) _____ ☐ CURIE(S); IDENTITY		

REQUESTER'S COMMENTS FOR THOSE HANDLING WASTE IN FIELD:

QUANTITY _____
WATTEN TOTAL _____

HEALTH PHYSICIST: EXECUTES THIS SECTION BEFORE MATERIAL TRANSFER

RADIATION DATA:
 BETA-GAMMA: FOR PACKAGE _____ ppm/hr @ _____ IN. OR FOR SHIELDED CASK _____ ppm/hr @ _____ IN.
 SURFACE CONT. _____ dpm/(sq. ft.); _____ dpm (a); NEUTRON READING _____ ppm/hr.
 HP SURVEYOR'S COMMENTS FOR THOSE HANDLING WASTE IN FIELD:

NP'S SIGNATURE		PHONE NO.
_____ <i>[Signature]</i>		_____

STORAGE AREA FOREMAN: COMPLETES AND SENDS COPY TO ORIGINATOR AFTER HANDLING WASTE

ACTION TAKEN: (CHECK ONE)		LOCATION:	DATE:
WASTE WAS: ☐ STORED, ☐ BURIED, ☐ COMPACTED, ☐ OTHER _____		SUSA NO. <u>5</u>	<u>04/12/77</u>
FACILITY: ☒ BUILDING ☐ WELL ☐ RAYNE	WASTE DESCRIPTION:	FACILITY NUMBER: <u>7826</u>	SUSA ATH <u>1474</u>
(Check One) ☐ TRENCH ☐ ON GROUND ☐ OTHER _____			

LOCATION WITHIN FACILITY: PFG <u>V44</u> PFT <u>02</u> , COMP. NO. _____		LAYER	FILE	RANK
PISBLE WASTE DATA: WELL FULL: ☐ YES ☐ NO		CRITICALITY COMMITTEE APPROVAL FOR:	SALE NO	GRAMS

COMMENTS FROM SUSA FOREMAN REGARDING WASTE AND/OR OPERATION:

FOREMAN'S SIGNATURE <i>John Smith</i>

**DISTRIBUTION: WHITE - STORAGE AREA FOREMAN RETAINS
 BLUE - COMPLETED AND RETURNED TO ORIGINATOR
 CANARY - RETAINED BY ORIGINATOR**

ORNL INPUT SHEET FOR DOE SOLID WASTE INFORMATION MANAGEMENT SYSTEM

DOC. NO.		DATE		SITE		BUILDING		REF. NO.	
NO.		DAY		MONTH		YEAR		NO.	
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30

PUNCH IN ALL COLUMNS

CARD CODE	ACTION	EMPLOYEE NUMBER	MISC.	INTERNAL LOCATION	PL. FROM ZONE	PL. FROM ZONE	FULL WELL	VOLUME (F ³)	WEIGHT (LBS.)	ISOTOPE 1/4		ISOTOPE 2/5		ISOTOPE 3/6	
										QUANTITY	NUCLIDE IDENTITY	QUANTITY	NUCLIDE IDENTITY	QUANTITY	NUCLIDE IDENTITY
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256
257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272
273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304
305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336
337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352
353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368
369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384
385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416
417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432
433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448
449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464
465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480
481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496
497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512
513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528
529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544
545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560
561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576
577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592
593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608
609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624
625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640
641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656
657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672
673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688
689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704
705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720
721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736
737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752
753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768
769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784
785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816
817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832
833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848
849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864
865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880
881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896
897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912
913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928
929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944
945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960
961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976
977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992
993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007	1008
1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024
1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040
1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056
1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072
1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088
1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104
1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120
1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136
1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152
1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168
1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184
1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200
1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216
1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232
1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248
1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264
1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280
1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296
1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312
1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328
1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344

[illegible]

Figure 2-4. Example of completed form shown in Figure 2-3.

The third step in the process was the entering of this data (Figure 2-4) into the data file and the assignment of a unique document number. There were a total of six eighty-character lines entered for each container of waste stored at ORNL. These six lines formed the record of each waste storage procedure. The process of keying in the data was handled by personnel at another plant, the Y-12 Operations Plant, some fifteen miles away. This group of personnel maintained the data on a large file in sequential format. This meant each additional storage required the entry of six more lines of data. To process the data, COBOL programs were developed which read the data file and produced monthly and bimonthly reports. Any other information obtained required additional COBOL programs.

2.1.2 Disadvantages of the data processing system. The major disadvantages of this system were the following:

- (1) a true data base was nonexistent;
- (2) specific ad hoc inquiries were time consuming and costly;
- (3) access and efficient use of the computer which contained the data file was difficult to attain due to a large number of users;
- (4) possibility of typographical and interpretation errors; and
- (5) Chemical Technology Division lacked complete control over the data file.

The first and foremost disadvantage was that a data base did not exist. There was only a data file which could be accessed sequentially by programs written for each specific request. The definition of a data base will be discussed in detail in Chapter 3.

The lack of an implemented data base led to the second disadvantage. Only inquiries for which programs had been written could be answered in a relatively short time frame, i.e., the monthly and bimonthly reports. This sequential data file was only accessible via programs designed for specific inquiries. Thus, each ad hoc inquiry demanded special programming time. Even the regular reports took some time to run and then to be sent over to ORNL. If there were any problems, it could possibly take up to two or three days to receive even the regular reports. Programs for ad hoc inquiries had to be specifically written, hence the user, the Chemical Technology Division, had to tolerate delays and additional cost for programmers' time.

The time requirements for developing specific programs were magnified by the large demand on the computer at the plants. Not only did the programmers have to write new programs, but they also had to compete with hundreds of users for computer time.

The physical process of filling out forms (twice), and then keypunching the data was also a disadvantage. It led to errors because the data had to be handled three times before actually entering the data file. There were three procedures (completion of the waste disposal form, the input sheet, and the actual entering of the data into the file) involved as the data was being entered.

Another problem was the distance between the data and the Chemical Technology Division. By keeping the data at Y-12 and having it maintained by another division, it really did not seem as though it "belonged" to CTD. For each inquiry made, requests had to be submitted to the Y-12 personnel. This situation did not provide CTD control over their own data. These problems eventually led to the development of the present system, and to this thesis project.

CHAPTER 3

THE DEVELOPMENT OF A NEW SYSTEM

3.1 Study of Data Base Management Models

3.1.1 Definition and objectives of a data base. A data base is built up of different entities. The smallest of these is known as the data item, also called attribute or field. Data item names refer to different types of information that will be contained in the data base. A data item contains values. These values can only be acceptable entries for a data item. This is known as the data item's domain. For example, STORDATE (a data item which names the type of information pertaining to storage dates) can only accept certain values (a range of values). An entry such as "03/01/167G" is not within the domain of the STORDATE data item. These data items or fields are grouped into records. A record type is determined by the types of data items contained in a record. For example, a record type such as WELLIND could contain the data items FULLIND and FULLAMT. Each of these data items, would have values that give information about the waste stored in wells. A record in the data file of the data processing system is made up of all the data items entered from Figure 2-4. A record is the collection of the values for such data items. A collection of these records is called a file. To say that the file from the data processing system was a data base is false. A data base is a collection of shared and integrated files with associations or relationships linking the files (Kroenke 1983, p.11).

"Integrated" means the data base contains the data for many users (Date 1977, p.4). In processing sequential data files, often there are requirements for a

data file for each user. This is a file which contains only the information for one particular application (Kroenke 1983, p.2). A data base allows all users to benefit from sharing one file.

The major objectives of a data base can be grouped into six categories: (1) ability to represent the inherent structure of the data; (2) reduction of data redundancy; (3) data integrity; (4) accomodation of changes; (5) performance and efficiency; and (6) ease of use and flexibility (Martin 1977, pp.35-42 and Cardenas 1979, pp.11-14).

Any group of data can be stored in a file and accessed sequentially for retrieval purposes. However, this method of data processing does not explicitly represent any of the relationships between data items or record types. Not only is the representation of this relationship important, but the inherent structure of the data should be represented as naturally as possible (Martin 1977, pp.35-42).

A data base which represents an inherent structure can form the relationships without a need for repeating like data values. This may lead to a reduction of the physical record size. For example, a record from the data file contains a unique document number which indicates each container of waste stored at ORNL. All the information concerning this storage is grouped into a record and connected through the document number. There are only a few types of waste (each repeated for every document number it refers to), but there are several thousand records each with a unique document number. A data base allows the user to only store the waste type once and connect it by some method to all pertaining document numbers.

Reduction of data redundancy helps to accomplish data integrity. If each type of waste is present only once in the data base, then updating it needs be

made only that one time. Compare this procedure versus possibly performing thousands of repetitive updates (which may lead to errors, either typographically or through omission). Data integrity also pertains to data as it is entered into the data base. Some data base management systems check ranges and types of data input. A data base management system could also have access codes allowing only authorized personnel to make changes to the data base.

The ability to accomodate future structural changes in a data base is a very important requirement of a data base management system. If a data base has been designed such that it can never be altered via the addition or deletion of a field or relationship, then it may quickly become outdated and unusable. The more adaptable a system is to changing information requirements, the more powerful this system will be.

The efficiency with which a system performs is certainly a factor with which all end users are concerned. Performance and efficiency are usually measured by way of a comparison of one data base management system to another or to some other method of data processing. Users need to compare the systems available for their use to realistically evaluate each system's efficiency and performance.

Finally, the ease of use of a system is an important objective. The user must be able to use the system to obtain its fullest potential. If a system is difficult to use, the user may not gain the full benefits of the data base management system.

By meeting each of these objectives, a data base management system will give the users control over their data.

3.1.2 Data base management model types. Data base management systems may be categorized into three kinds of models: (1) the hierarchical approach; (2) the relational approach; and (3) the network approach. These models differ in how they represent the data (the data base structure), and in the methods each of them use in storing and retrieving the data through a data base management system.

3.1.2.1 Hierarchical data base model. A hierarchical model of a data base refers to a tree-structured relationship among data items. This is a representation of a parent-child relationship where one value of a data item (parent) may have many values of another data item (children) associated with it. An example of a hierarchy within this application is one building with many types of wastes that can be stored in it. This is known as a one-to-many (1:M) relationship.

A hierarchical structure is easy to represent and maintain. However, even though a hierarchical structure represents 1:M relationships, a data item cannot be shared in two or more relationships. The data item must be physically present in every relationship it is needed in. To illustrate this restrictive problem, consider:

- (1) For each container of waste stored at ORNL, there is a unique document number generated that records each storage procedure;
- (2) Containers of waste are all classified as a particular nuclide (1-6). If there are thousands of containers stored, then a particular nuclide will most probably be stored more than once. Therefore, a document number is generated to represent each of these storages;

- (3) There may be several containers of waste stored at a particular location. Therefore, a document number is generated to represent each of these storages.

To summarize this example: for one nuclide classification, there may be several document numbers; for one storage location, there may be several document numbers. Figure 3-1 demonstrates these two relationships.

Figure 3-1 also illustrates the problem associated with hierarchical structures in that the data item, DOCNO, cannot be shared as a child. Therefore, the only way to avoid deletion of one of the relationships is to have two disjoint trees (where the DOCNO data item is a child of both trees) and this is usually a limiting situation (Martin 1977, p.102). One such limitation occurs when trying to perform updates to the data base. If the document number data item is a child of both trees, then any modification to a particular value of the data item would have to be performed twice. This could quickly lead to data corruption. For example, the data base programmer may forget to modify both values, or one value could be modified incorrectly.

There is also a relevant relationship in this application which cannot be represented. There are many different nuclides stored at each location, and a particular nuclide could be stored at many different locations. This type of relationship is known as a many-to-many (M:N) relationship. There is no straightforward representation of this type of relationship in a hierarchical structure. In a hierarchy, it is difficult to obtain what nuclides are stored at a location and also all the locations where a certain nuclide is stored.

3.1.2.2 Relational data base model. The basic concepts of the relational model are easy to comprehend. A relation can be represented as a

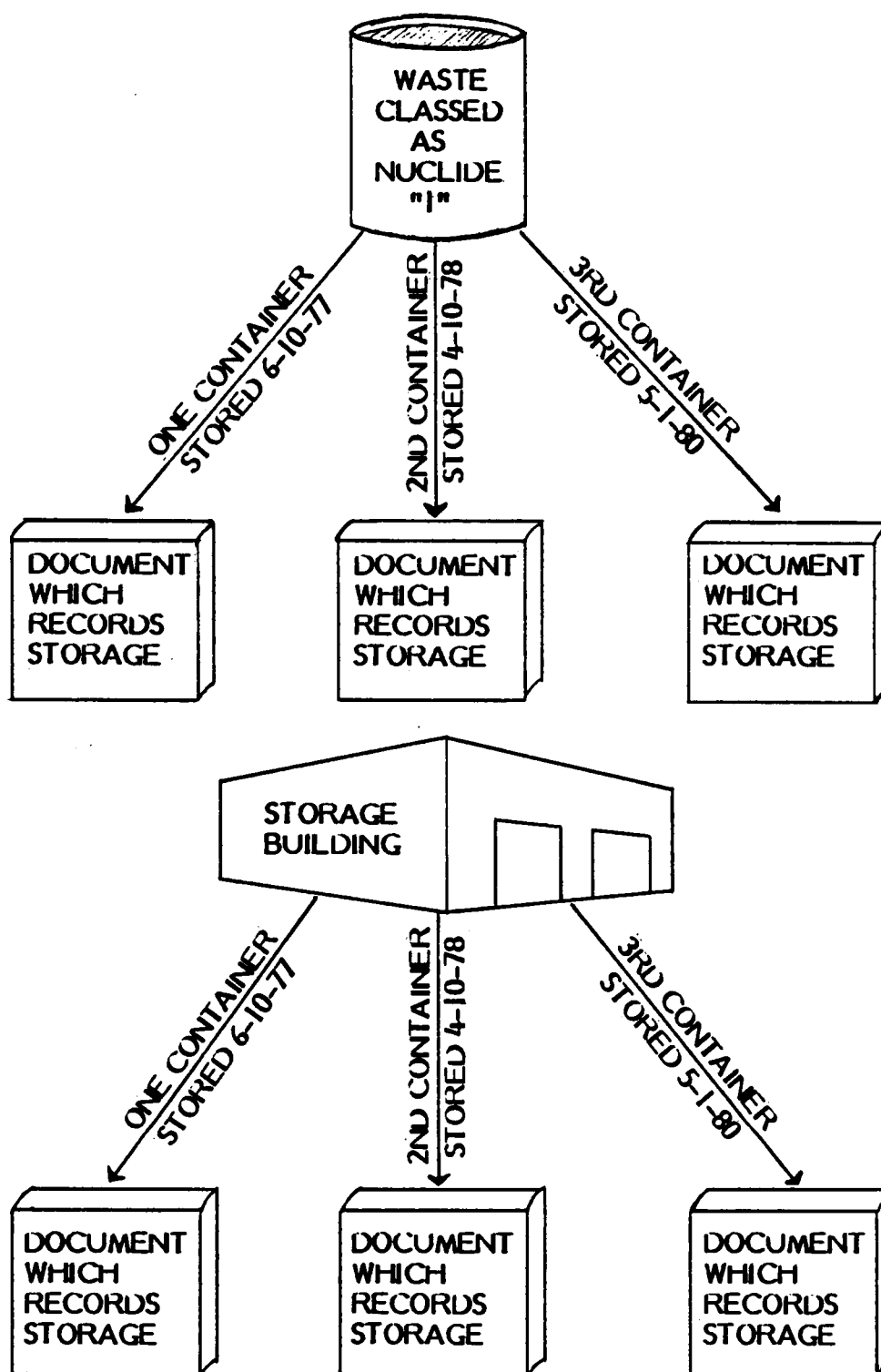


Figure 3-1. Sample data in hierarchical form.

table which is made up of data items, or attributes. Each attribute has a defined domain, and the collection of these domains are said to be the domains of the relation. Each row of the table represents one tuple of the relation. The number of tuples in a relation can be thought of as the number of record occurrences in a table (Date 1977, pp.73-74). The following criteria describe any table which is in a relational data base:

- (1) each attribute value is atomic (i.e., at every row-and-column position in the table there must be exactly one value, or null, never a set of values);
- (2) all elements in a column are taken from the domain of that attribute;
- (3) each column has a distinct name pertaining to the data item;
- (4) each row is distinct, duplicate rows are not allowed; and
- (5) viewing the rows and columns at any time in any sequence will not affect any context information or the semantics of any function using the table (Martin 1977, pp.202-206).

Figure 3-2 is an example of one relation which could be used for this application. This relation represents the type of waste and its isotopic breakdown for each container of waste stored.

In a relation, the tuples do not have to be ordered, since a relation is a set of data items and sets are not ordered. However, it is often beneficial to order the tuples of a relation as has been done in Figure 3-2. An order gives meaning to commands such as "get next" which in this example would fetch WASTE tuples in ascending order of DOCNO (Date 1977, p.74).

Another important concept in the relational model is the use of keys. Usually within a relation there is one attribute that uniquely identifies the tuples

WASTE

DOCNO	WASTETYP	ISOT1	ISOT2	ISOT3	ISOT4
37601	CE	U-233	U-235	NULL	NULL
37602	DD	PU-239	U-233	AM-241	U-235
37603	CE	U-233	NULL	NULL	NULL
37604	NC	AM-241	PU-239	U-233	U-235
37605	NC	PU-239	U-238	U-235	NULL

Figure 3-2. Sample data in a relational table.

in a relation (Date 1977, p.77). For example, in Figure 3-2, the DOCNO is the primary key because it alone distinctly identifies each tuple. For some relations, there may be another attribute, known as the secondary key, that may have this identification property. Not every relation will have a single-attribute key as in Figure 3-2. However, every relation will have some combination of attributes, when used together, that identify its tuples. An example of a two-attribute key is shown in Figure 3-3. Neither BUILDING or STORDATE alone identifies a tuple because there are duplicate values for each. Only by using BUILDING and STORDATE together can a tuple be uniquely identified.

3.1.2.3 Network data base model. The network model is similar to the hierarchical approach in that data records are related by parent-child relationships. The main difference however, is that data items may be present in any number of hierarchies (Date 1977, p.58). In the network model, the data items that represent parents are referred to as owners, and the data items that represent children are known as members. The relationship represented between an owner and its member is called a set. These sets are usually given a name that represents this relationship. Figure 3-4 illustrates how the nuclides and locations can both be linked to the same document number.

This model allows for a more straightforward retrieval of M:N information. Recall that a user may want to know all the locations where a particular nuclide is stored. This could be represented as a 1:M set. However, there may also be a need to know all the nuclides stored at a location. This too is a 1:M relationship, but to avoid having two sets where DOCNO is the member of both sets as with a hierarchical model, a network model can be used. By using Figure 3-5 as an example, one can obtain all the locations for nuclide "I" by

AMOUNT

BUILDING	STORDATE	WEIGHT
9019	03/16/1979	100
9019	05/10/1981	200
9019	06/11/1982	300
1022	03/16/1979	400
1022	06/11/1982	200
1506	05/10/1981	100
1506	06/11/1982	200
1506	09/19/1983	300
1410	06/11/1982	200

Figure 3-3. Example of a relation with a two-attribute key, BUILDING + STORDATE.

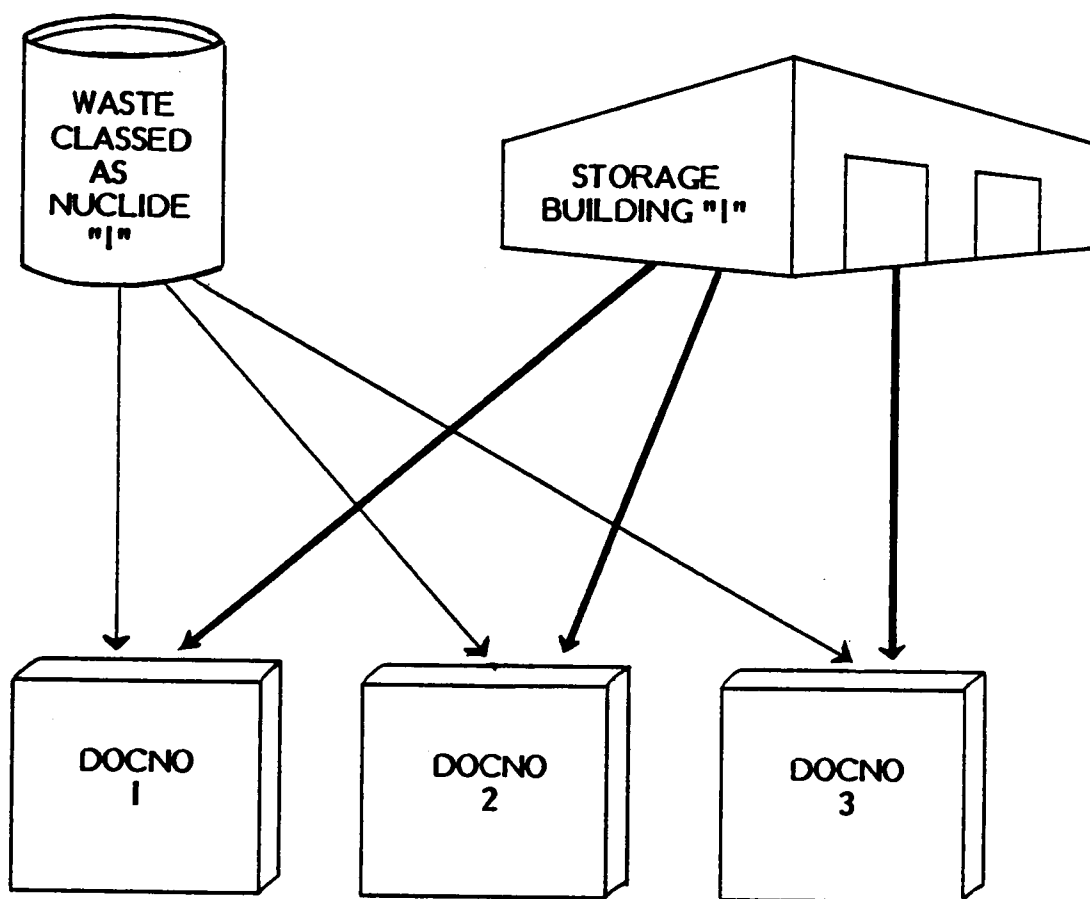


Figure 3-4. Sample data in a network model.

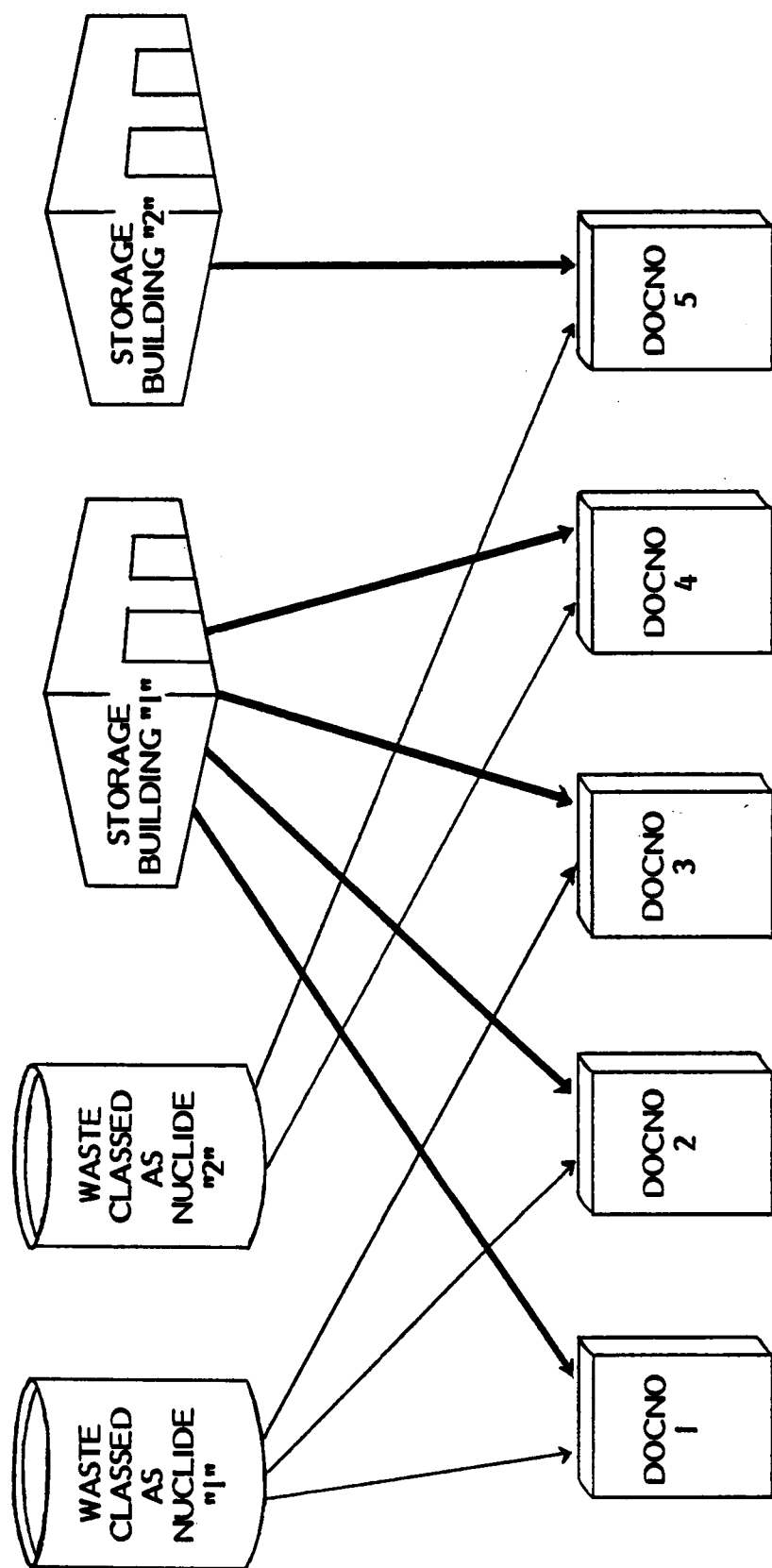


Figure 3-5. Example of M:N information in a network set.

finding each member, the document number, and then finding its owner of the location set. This would be storage location "1". For nuclide "2", this would result in location "1" and "2". To obtain the nuclides stored in storage location "1", obtain the document numbers that are its member and then their owner. The results are nuclide "1" and "2". A conceptual view of these two 1:M sets is given in Figure 3-6. Having the document number as a member of both sets is an instance of intersecting data. It is shared by both the nuclide and the location owners.

Figure 3-7 demonstrates that when the intersecting data of the 1:M sets is removed, a M:N set is directly represented. In the standard network model the 1:M sets are the only representations available. They must be used to obtain M:N set information. However, the data base management system used in this thesis directly represents M:N sets and will be discussed in detail in the Section 3.2.2.2

3.2 The Choice of a Micro Computer Based System

3.2.1 The advantages of a micro computer system. The previous data processing system needed improvement. The first alternatives to be considered were whether to keep the system on a large-scale computer and use a data base system present on this computer, or to implement a new system on a micro computer. In choosing between these alternatives, the users' system requirements had to be evaluated. Their needs included:

- (1) faster turn-around time for normal requests;
 - (2) an ability to obtain ad hoc requests without additional time and costs;
- and

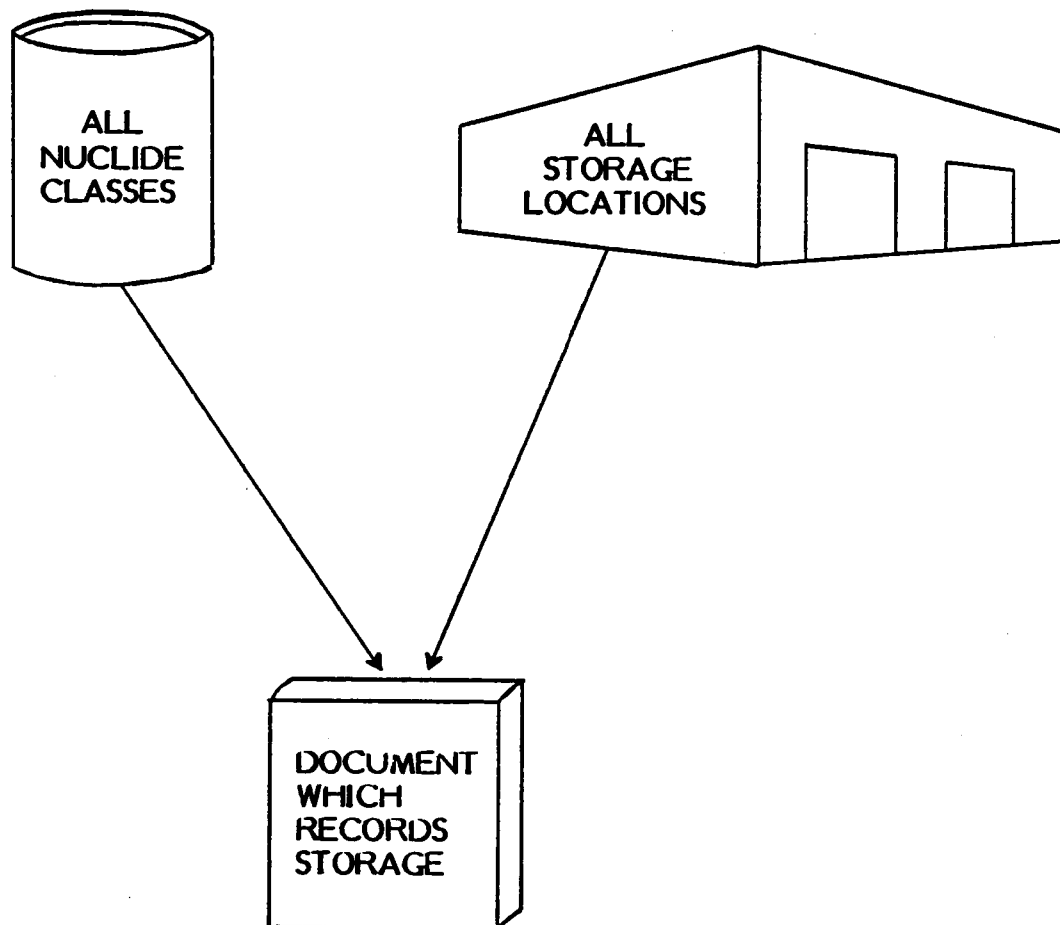


Figure 3-6. Conceptual view of two 1:M sets in a network system. The nuclide class and storage location are the owners with the document number as the member of both sets.

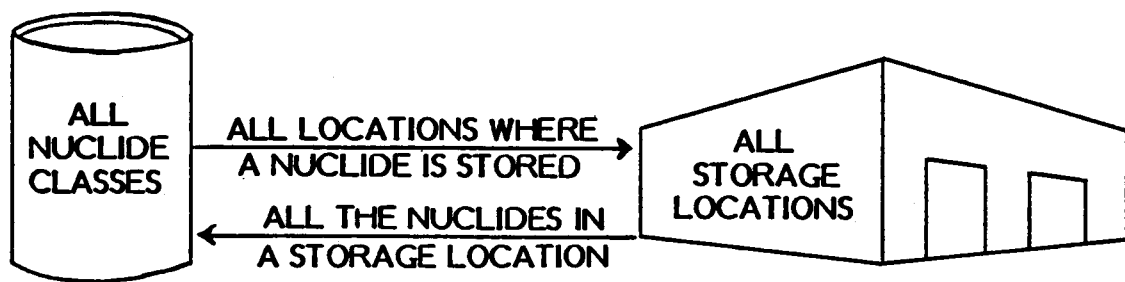


Figure 3-7. Conceptual view of a M:N network set.

- (3) independence from outside personnel in producing all the required information and updating of the information.

Based upon these requirements, the choice was made to implement a data base system on a micro computer for the following reasons:

- (1) a large-scale computer still required shared access of a computer with a large number of users;
- (2) a micro computer with 10 megabytes of hard disk storage would accomodate all of the present data plus several years of additional data;
- (3) there are existing micro computer based systems that support interactive, user-friendly querying of the data base; and
- (4) the micro computer could be used for other requirements of the Chemical Technology staff, e.g. spread sheets and word processing.

3.2.2 The choice of MDBS III for implementation. MDBS III, an extended-network data base management system, was chosen to implement the micro computer data base. The other major software consideration was dBASE II (vended by Ashton-Tate) which was currently the number-one selling micro data base management system on the market. During comparison of these two systems, the users' micro system requirements were considered which included:

- (1) security of the data through access control;
- (2) multi-user option;
- (3) user-friendly query language; and
- (4) acceptance of scientific notation entries.

3.2.2.1 MDBS III compared to dBASE II. From previous experience with dBASE II, it was determined that it did not support several of these features

required by the users. Table 3-1 is a comparison summary between MDBS III and dBASE II. The lack of these features was the deciding factor for not using dBASE II.

An advantage of MDBS III is the inherent nature of network systems to use high-level host languages in design of application programs. This topic is covered more completely in the following section, but, at this point, it suffices to say that programming with a high-level language coupled with a data base management system provides more programming flexibility than using only the data base management commands provided by the system as is the case with dBASE II. The next section details other important features that influenced the choice of MDBS III.

3.2.2.2 MDBS III features. The basic architecture of a data base management system is composed of:

- (1) the physical structure of the data base;
- (2) the logical schema of the data base; and
- (3) the application programs.

The physical structure of the data base is how the data is organized or stored on the physical media. The methods employed depend upon the type of data base management system. The user, and often the data base application programmer, need not be aware of this physical structure.

The logical schema is the associations among all the data items as conceptualized by the user. It is the logical description of the data base (Martin 1977, p.60). The examples of Figures 3-6 and 3-7 illustrated a partial schema of the data base developed for this application.

Table 3-1. Comparison summary between MDBS III and dBASE II

Features	dBASE-II ^a	MDBS III ^b
Cost	\$700	\$2000 +
Security	no	yes
Multi-user	no	yes
Validity checking	no	yes
# rec./file	65,535	unlimited
# fields/rec	32	65,535
# files open	2	unlimited
Record size	1000	65,535
Field size	254	65,535 char type 250 var. length
Fixed/variable record length	variable	either
Screen generator	programmable	option
Acceptance of scientific notation	no	yes
Access a file as input?	yes	yes

a (dBASE II Manual), b (MDBS III Manual)

The application programs are used in retrieving and modifying the data base values. By using the logical schema, application programs are developed to transverse through the relationships to obtain or modify information. These programs could be menu-driven and prompt the users for their retrieval information, or a program could be developed that prompts the user for entry values and then makes updates or new entries in the data base. Figure 3-8 illustrates these three concepts of a data base management system.

MDBS III has several software packages that enable the user and the data base application programmer to incorporate these three levels. The standard package includes:

- (1) Data Definition Language (DDL) - language used to define the data base schema; and
- (2) Data Manipulation Language (DML) - language used in writing application programs, e.g., read the data from a file and enter it into the data base, or used to access the data and produce reports.

The DDL is a language that is used to describe the logical data base schema. By using DDL commands, every data item, record type, and set is defined. The commands are grouped so as to form a DDL program; and upon execution of this program, an empty data base is intialized according to the specified relationships. Figure 3-9 is a partial sample of this application's data base definition, and Figure 3-10 is a conceptual view of this definition.

The DML is used in the development of the application programs. The DML is an interface between an application program written in a host language, such as BASIC, COBOL, or PASCAL, and the physical data base structure. The DML commands are used as function calls into the data base to execute required

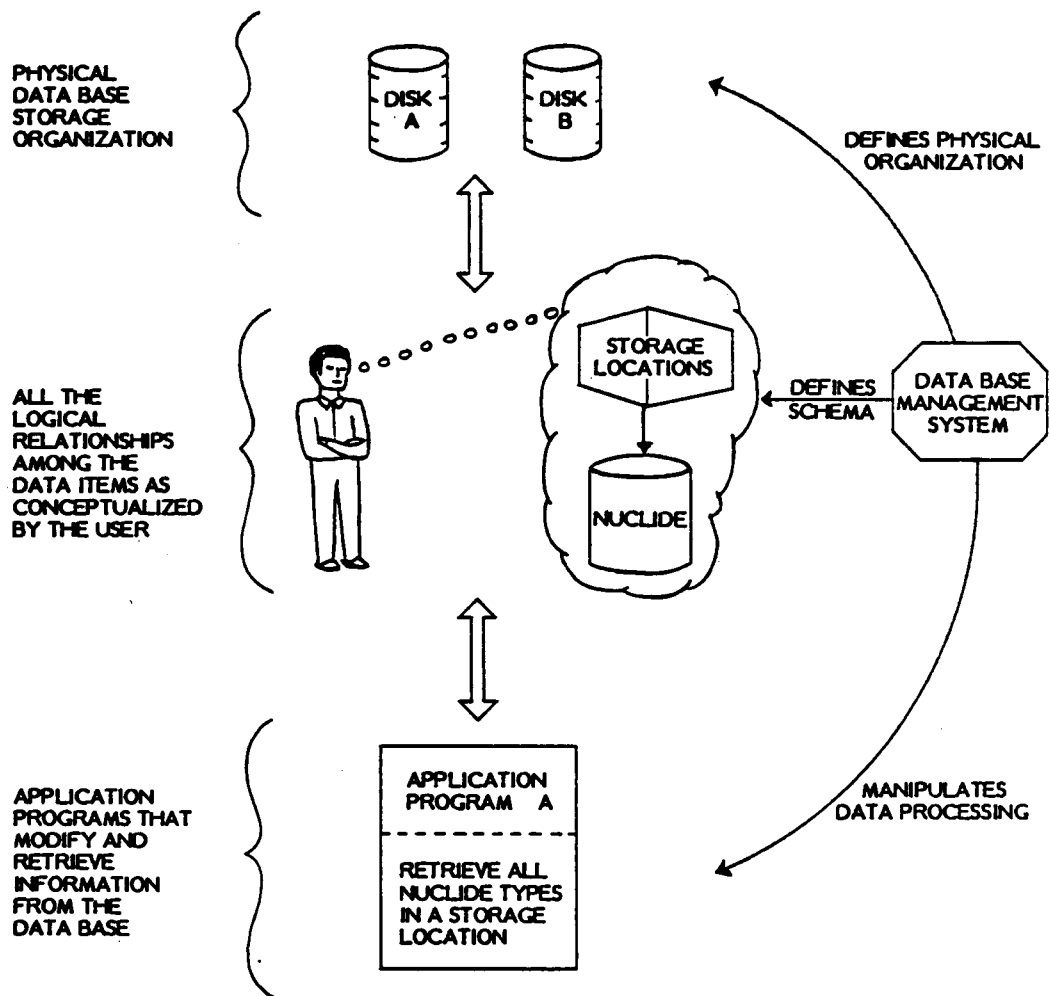


Figure 3-8. The architecture of data base management systems. (Taken from Martin 1977, p.84)

FILE "WASTE.DB"
USER NAME IS "USER" WITH "PASSWORD"

RECORD is BLDG
ITEM is BUILDING, char 6

RECORD is NUC
ITEM is NUCLIDE, char 1

RECORD is DOCUMENT
ITEM is DOCNO, integer 6

SET if SSI type is 1:M
OWNER is SYSTEM
MEMBER is BLDG

SET is FROM type is 1:M
OWNER is BLDG
MEMBER is DOCUMENT

SET is WHAT type is 1:M
OWNER is NUC
MEMBER is DOCUMENT

Figure 3-9. Sample DDL program. This program defines the data base, WASTE.DB, to contain the records BLDG, NUC, and DOCUMENT. Each record has only the one data item defined. Two 1:M sets are defined with the DOCUMENT record as intersecting data and the BLDG and NUC records as the two owners. There is also a 1:M set with the SYSTEM as the owner and BLDG as the member.

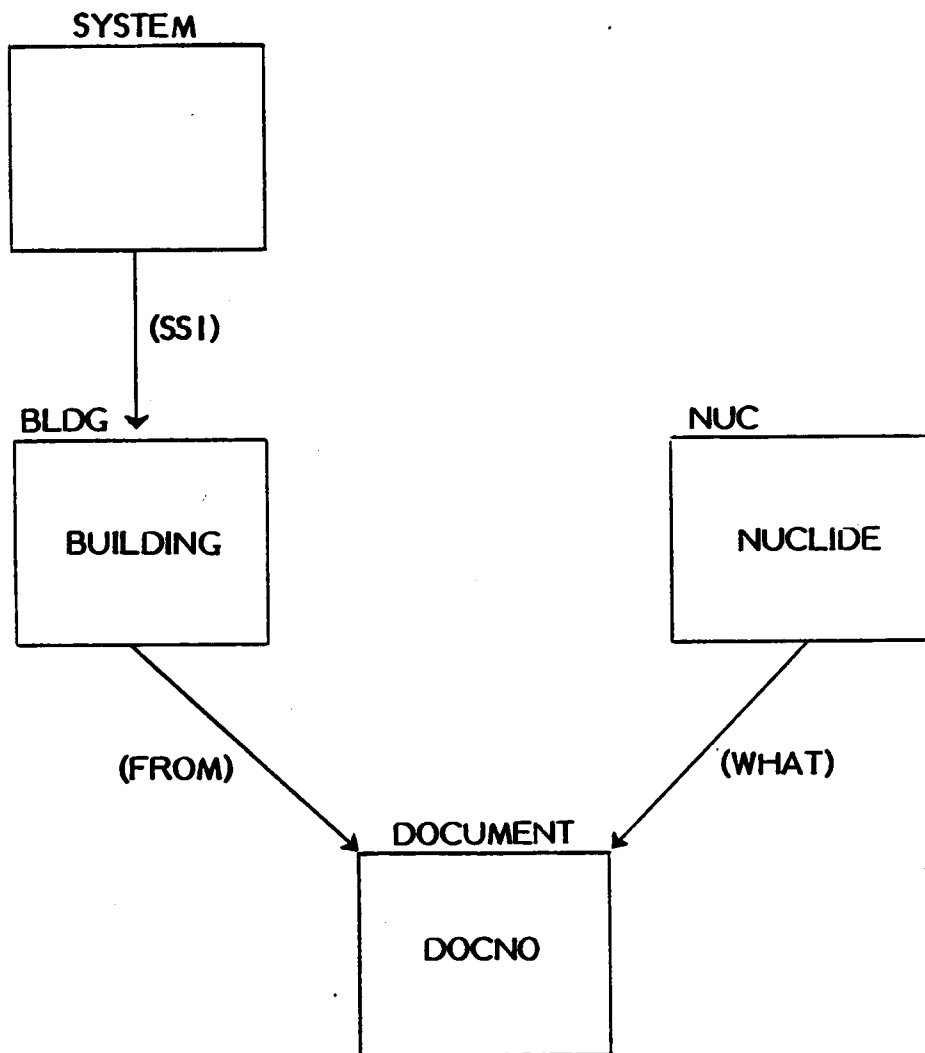


Figure 3-10. Conceptual view of the data base defined in Figure 3-9. The three 1:M sets are SSI, FROM AND WHAT.

functions. Figure 3-11 is an example of three DML commands and how they would be used in a BASIC application program. The variable "cmd\$" is assigned the value of the desired DML command (the command is in the double quotes), and then the command is invoked using the CALL DMS% statement.

These two software packages enable the programmer to define the data base and then access and modify it. Other features of MDBS III aid in the ease of programming of application programs and enable a user to do ad hoc querying of the data base. These optional MDBS III features include:

- (1) Interactive Data Manipulation Language (IDML) - the counterpart to the DML, one IDML command can be executed at a time without the requirement of writing an application program as with DML;
- (2) Query Report System (QRS) - a user-friendly, ad hoc query language that enables a user to obtain any information with the results printed in a semi-report fashion; and
- (3) Interactive Screen Definition Language (ISDL) - an interactive screen generator that facilitates the development of menu-driven application programs.

The IDML serves as an excellent debugging tool for a programmer. Each IDML command can be executed without the framework of an application program. This ability allows each statement to be checked for correct syntax and semantics before programming the DML commands into an application program. Figure 3-12 demonstrates the IDML counterparts of the DML commands shown in Figure 3-11.

QRS can be used by programmers to produce ad hoc results quickly. The commands are non-procedural, English-like statements that users can learn

cmd\$ = "FFM, set"
CALL DMS%

Find First Member

Finds first member connected to the owner of the set which becomes the current member of the set.

cmd\$ = "GETM, set, iblk"
CALL DMS%

GET data from Member

All data values in the current member of set are returned in the host language variables of iblk.

cmd\$ = "FNM, set"
CALL DMS%

Find Next Member

If the current member of set is not null when FNM is invoked: The next member record (connected to the current owner of set) following the current member of set, becomes the new current member of set.

Figure 3-11. Three example DML commands as they would be used in a BASIC application program.

FFM, set

Find First Member

Finds the first member of the set as the DML command. However, no BASIC commands are necessary.

GETM

GET data from Member

Automatically gets the data values of the current member and lists them on the screen. In the DML command, after obtaining the values in the iblk, the programmer must use BASIC commands to display the results.

FNM, set

Find Next Member

Finds the next member of the current set as the DML command does.

Figure 3-12. IDML counterparts to the DML commands shown in Figure 3-11.

easily for their own ad hoc queries. The results are produced in tabular form with the data item names used as column headings. Figure 3-13 is an example of one QRS command and its results.

ISDL can be used to develop menu-driven application programs. ISDL defines placement of the screen contents, mapping to the data base, and produces results on the screen or printer in user-defined reports. The ISDL enhances development of specific programs such as a menu-driven input program that prompts the user for the input data and directly enters the data into the data base.

Another important feature of MDBS III is that it is an extended-network data base system. This system supports the standard 1:M sets of network systems; moreover, it supports the direct representation of M:N sets as discussed in Section 3.1.2.3. This type of set was of importance to the user in the application here considered. The Chemical Technology personnel repeatedly need information on:

- (1) what nuclides are stored at a location;
- (2) what types of wastes are stored at a location;
- (3) what isotopes are stored at a location; and
- (4) what types of containers are stored at a location.

The opposite four questions are also frequently asked (at what locations is nuclide "X" stored?, and so forth). MDBS III is capable of directly representing these relationships without the use of intersecting data. Many large-scale data base management systems do not support M:N sets (Kruglinski 1983, pp.20-25).

→ LIST BUILDING DOCNO NUCLIDE THRU SSI FROM >WHAT

BUILDING	DOCNO	NUCLIDE
1	1	1
1	2	1
1	3	1
1	4	2
2	5	2

Figure 3-13. Example QRS query. This request asks for all the values of the data items BUILDING, DOCNO, and NUCLIDE. A path is specified through SSI, FROM, and to go from the member to the owner of the 1:M set WHAT. This path is needed to pass through all the records that contain values for these data items. The results are the document numbers and nuclides associated with a building.

According to Martin (1977, pp.115-116), there are two reasons for this:

- (1) it is not a straightforward method to physically represent M:N relationships; and
- (2) in practice, when M:N relationships are needed, there is often intersecting data present.

While the argument of M:N physical representation may be true for development of data base management software, it is of little consequence to an application programmer using MDBS III. This feature has been implemented with success in the software and the complexity is transparent to the user.

Secondly, Martin states that there is usually intersecting data; and if there is none initially, it is likely to be added later as the "data base usage evolves". Therefore, he states, there is no need to represent M:N sets. However, such was not the case in this application, where there was a definite requirement for information from several M:N sets. Even though there was existing intersecting data, the M:N sets could also be directly implemented and used without traversing through intersecting data. This ability provided the following enhancements to the data base designed for this application:

- (1) increased speed of access (eliminated one record that had to be accessed during a M:N information retrieval);
- (2) easier programming of application programs; and
- (3) enhanced the teaching and understanding of users in performing their own ad hoc queries.

By using a micro computer coupled with MDBS III, a flexible, interactive data base could be developed for the user to provide:

- (1) fast, user-friendly access of the data;

- (2) a resident, more independent data base; and
- (3) one-time entry of the data.

3.3 Design of the New Data Base

This section discusses the general approach to the design of a data base and focuses on this application. The steps involved in the design are:

- (1) an analysis of the existing data:
 - (a) to determine the definitions of all data items; and
 - (b) to determine the definitions of all relationships among data items;
- (2) to determine the users' requirements; and
- (3) design of a data base to meet these requirements.

3.3.1 Analysis of the data

3.3.1.1 Definition of the data items. There are two steps in defining data items. These are:

- (1) to define the meaning; and
- (2) to define the domain of each data item.

This first step is crucial before any data base can be appropriately developed (Kruglinski 1983, pp.1-10). This involves the complete understanding of every data item's concept. For example, this application has the data item BUILDING. It is evident that the values accepted as entry represent buildings, but the data item name could represent several concepts. Such as:

- (1) buildings where containers of waste are stored;
- (2) buildings where waste is received prior to storage; or
- (3) buildings where particular personnel is housed.

By this example, it is clear that a data item concept must be defined before a domain is defined and before any relationships to other data items can be determined.

The second step is the process of determining the data item domains. This is establishing all the legal entry values (or a range of values) for each data item. This step is essential for defining the data base field types, writing application programs, and maintaining data integrity (validity checking upon entry).

The data item descriptions for this application are summarized in Appendix A. The first column contains the field name of each item in the data base, followed by a description of that item concept. Appendix B further describes the data by listing each data item type and domain. The length column in Appendix B indicates the maximum number of bytes allowed for each field entry in the original file.

3.3.1.2 Definition of relationships. The second step in the analysis of data is to define the relationships among the data items. During this application, this step was accomplished through interviews with the users. The users knew how each data item related to another data item. Figures 3-14 - 3-17 are illustrations of the most relevant relationships for this application.

3.3.2 Definition of users' requirements. An evaluation of the users' most requested inquiries gave the following results:

- (1) what nuclides are stored at a location;
- (2) what isotopes and their quantities are stored at a location;
- (3) what nuclides are stored in a certain type of container;
- (4) what containers are stored at a location;

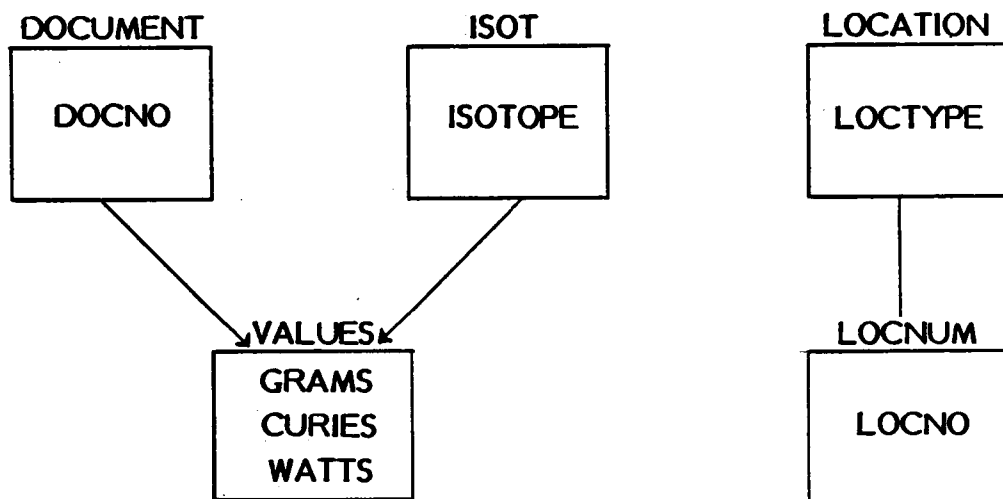
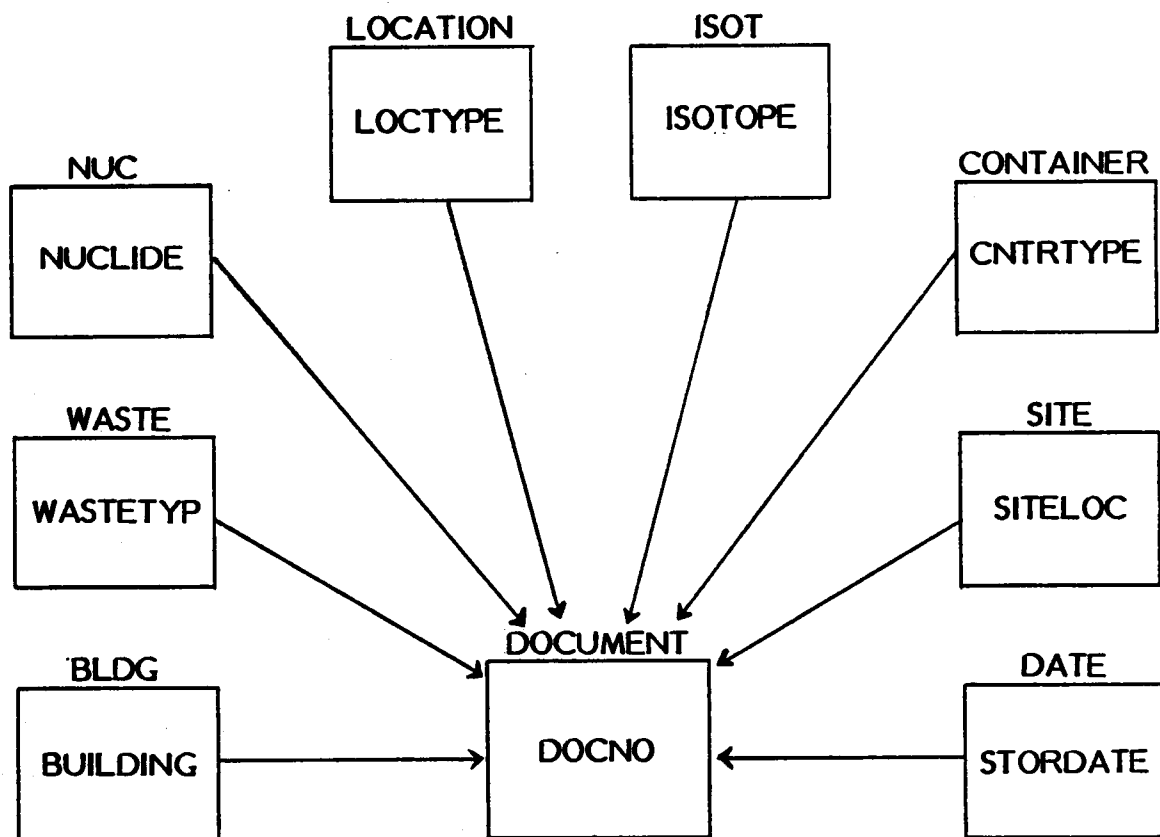


Figure 3-14. Most needed 1:M relationship information.

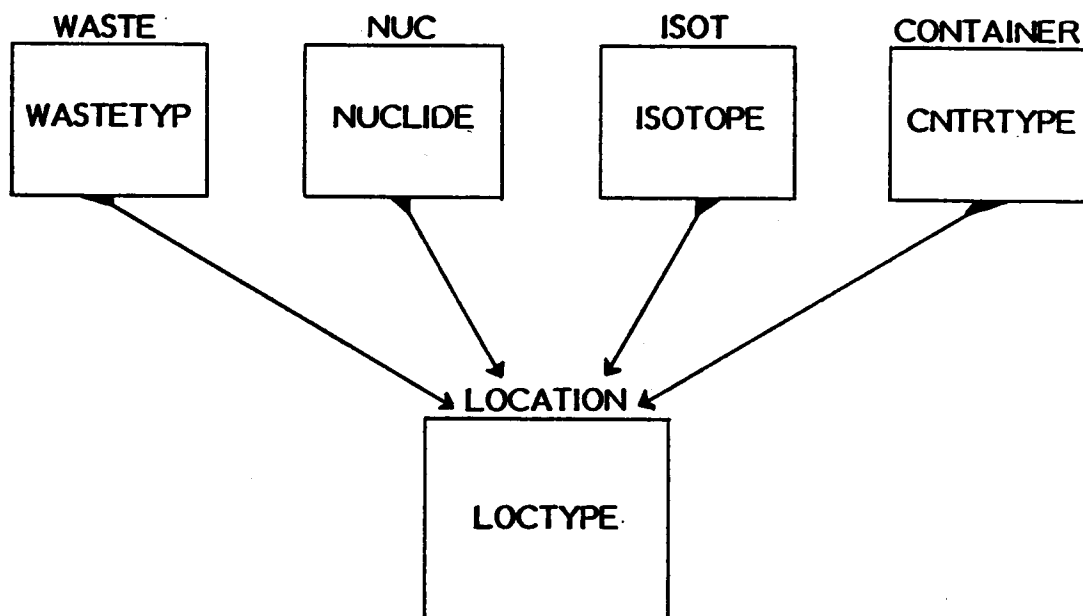


Figure 3-15. Most needed M:N relationship information.

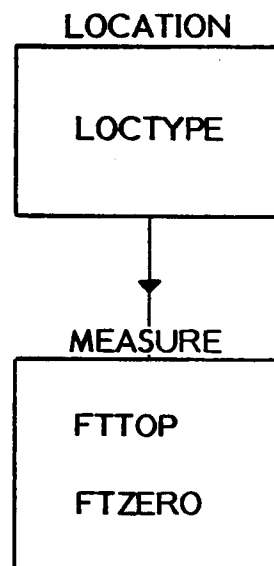
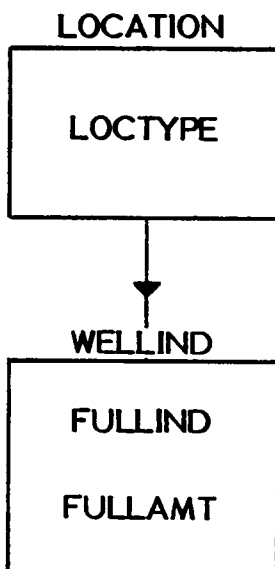
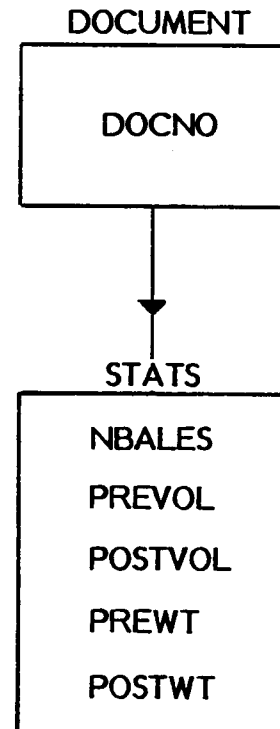
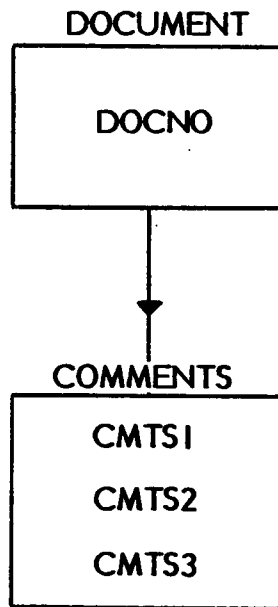


Figure 3-16. Most needed 1:1 relationship information. A 1:1 relationship is for each occurrence of the owner, there is 0 or 1 occurrence of the member.

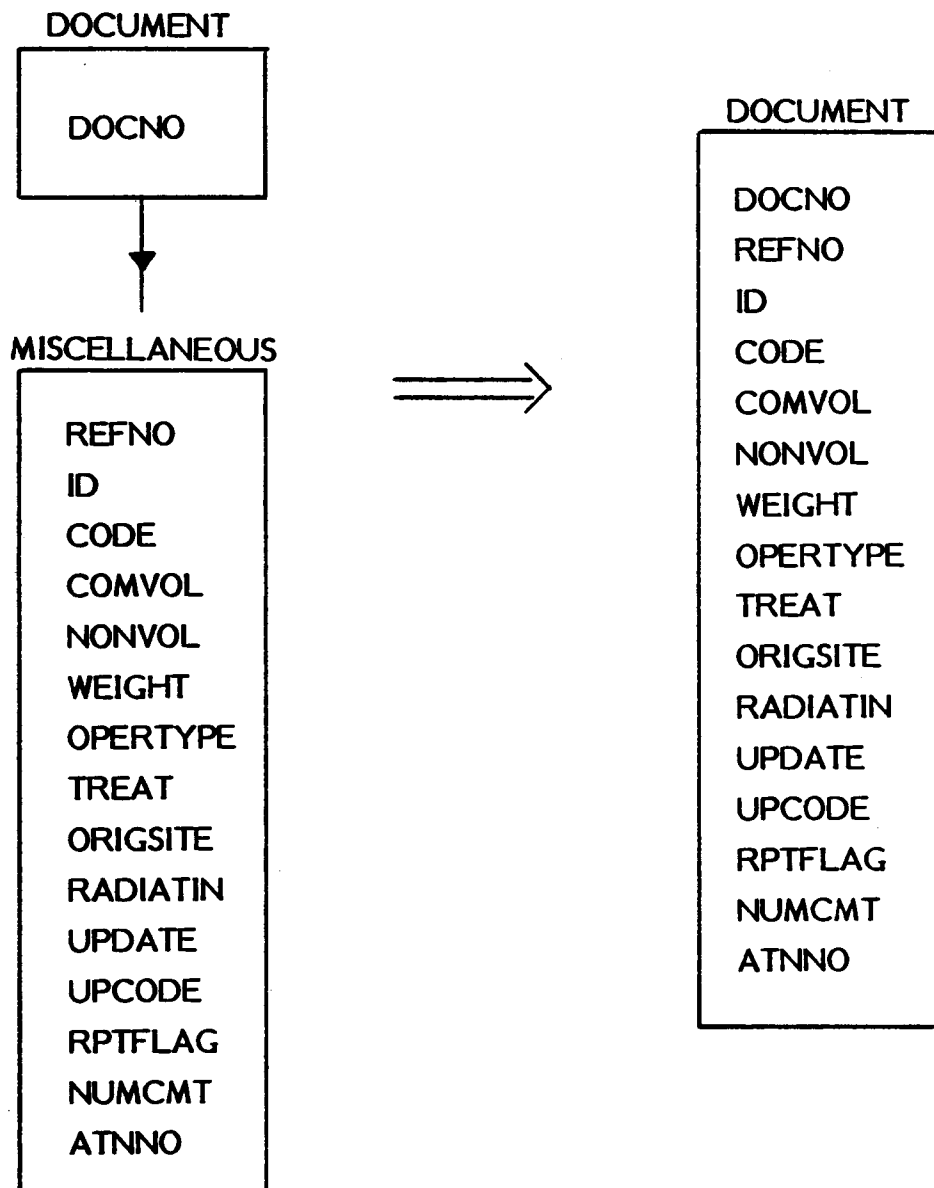


Figure 3-17. The development of the DOCUMENT record type. There was a 1:1 relationship between DOCNO and these data items of the record MISCELLANEOUS. Since these data items had little individual importance, they were grouped into the record type DOCUMENT with DOCNO.

- (5) what is the quantity of a particular isotope stored for a particular date; and
- (6) what types of wastes are stored at a location.

These requirements were given full attention during the development of the data base. Its design was intended to provide the user with fast, flexible access in obtaining these results.

3.3.3 Design of the data base to meet the users' requirements. To attain a flexible and efficient design for a data base, the procedures discussed in Sections 3.3.1 and 3.3.2 must be completed. By first defining the data items and their relationships and then incorporating the users' needs, a useful, efficient data base can be developed.

Figure 3-18 depicts the final data base schema. The major relationships are the 1:M sets from building, waste type, nuclide, location type, isotope, location site, and storage date to the document number. Each of the records are also members of system sets. A system set allows for quick access to a record since it is the first step into the data base. As detailed in Section 3.3.2, these data items were the most important data items required by the users; therefore, the system sets provide direct access to these data items. There are four M:N relationships directly implemented between the location type from: (1) type of waste; (2) the nuclide; (3) the isotope; and (4) the container. This allows the users to obtain inquiries directly about the types of waste, nuclides, isotopes, and containers stored at a particular location without processing through the DOCUMENT record. The M:N sets also easily give the reciprocal information on the location types where a particular waste type, nuclide, container, and isotope

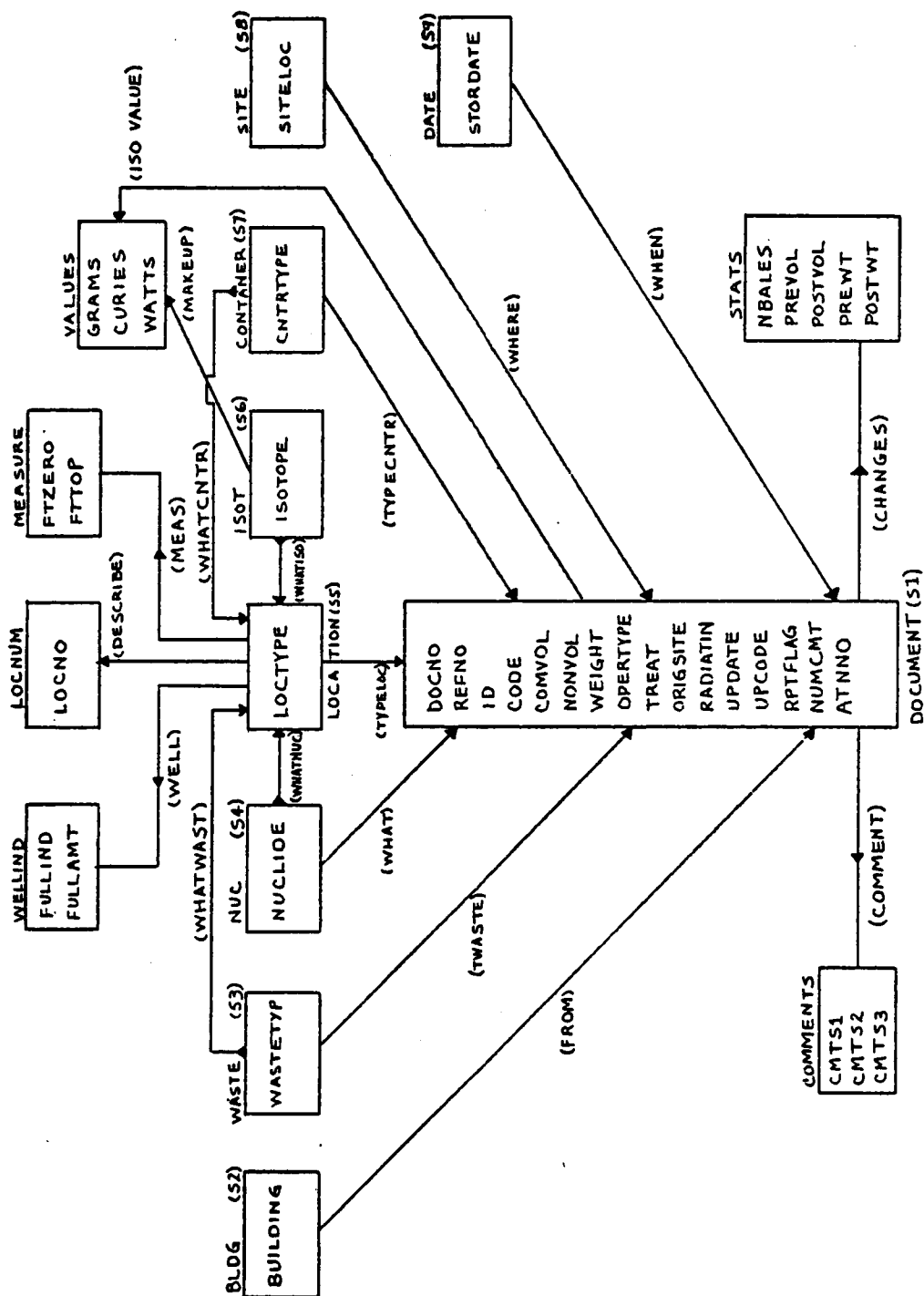


Figure 3-18. Final data base schema.

is stored. When access for a particular document number is needed, there is fast access via the system set SI on the DOCUMENT record. The data definition of Figure 3-9 introduced this concept: the system owned set. This is a set where any record can be a member and the owner is the system. These sets function as "gateways" to the data base. The data base is entered (accessed) through the set owners (the system).

A problem encountered during initial testing of the data base led to a refinement process. The problem and its solution are detailed below.

As shown in Figure 2-3, there can be four isotopes entered to detail the makeup of the waste being stored. Each isotope has an associated quantity and unit. The units can be either grams or curies. The initial phase of the data base loaded the maximum of four isotopes, their quantities, and their units for each container of waste stored. In original thinking, the schema was to represent the 1:M set where a particular isotope could be part of several containers of waste stored (this would mean many documents would record the storage of an isotope). This concept is true, but the original design did not allow an efficient method for obtaining any information that users required after the initial schema was laid out.

The first design established the record type ISOT to contain the data items ISOTOPE, QUANTITY, and UNIT. These three data items were also array items and could have as many as four values each. ISOT was the owner of a 1:M set where the DOCUMENT record was the member. This schema is depicted in Figure 3-19. The problem with this design is that the relationship actually represented a 1:1 relationship between ISOT and DOCUMENT. The definition of a 1:M relationship is that for each occurrence of the owner there may be many

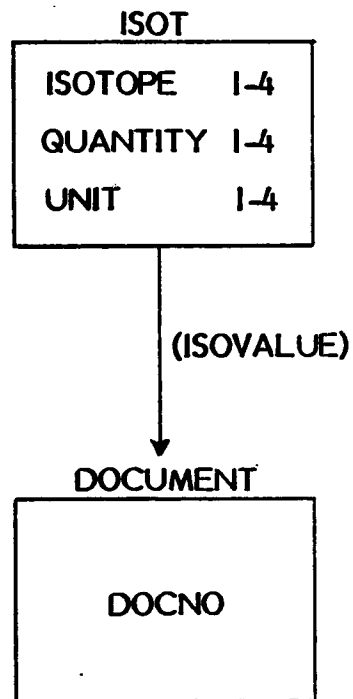


Figure 3-19. Incorrect initial schema designed to represent an isotope can be stored many times.

occurrences of the member. This means every occurrence of an owner record is unique from another occurrence of an owner record for a set. Therefore, all twelve data item values (4 values * 3 data items) had to be exactly the same for each occurrence of its' associated member. Under this arrangement, many containers of waste were assumed to have identical isotopic makeup which conflicts with reality.

The second problem discovered with this design was that it was useless for providing quantitative results. This was evident in two situations:

- (1) to obtain all the information for a particular isotope, all the owner records (which in essence was the same number of records as the number of documents) had to be searched to find all the occurrences of that isotope; and
- (2) because of conversion restrictions, the values in the ISOT record could not be added conveniently.

The first situation stemmed from the flaw in the design of the I:M set. Any variance in the quantity or unit caused another owner record to be established. This required the processing of all the ISOT records to obtain the quantities for one isotope.

The second reason for the difficulty in obtaining quantitative information resulted from the fact that the quantity values were entered in either grams or curies. Grams and curies cannot be added together. For example, two grams of Uranium 235 and two curies of Uranium 235 do not equal four grams or curies of Uranium 235. There is a specific conversion factor that converts gram quantities to curie quantities, and vice versa, and only then can the two

quantities be added together. To complicate this situation, each isotope has its own conversion factor.

In refinement of this initial phase of development, the conversion factors were implemented into the data base loading program. There are now routines that convert gram units to curies and curies to grams depending upon which unit is in the original data. Then after the curies have been calculated (or entered), there is a conversion to watts from curies. These values are important for producing reports where there is a need to know the amount of waste in grams, curies, and/or watts.

The new design implements two 1:M sets where the record types ISOT (with data item ISOTOPE) and DOCUMENT are the owners and the record type VALUES (with data items GRAMS, CURIES, and WATTS) is the member of both sets. This is an instance where intersecting data plays an important role.

Recall that DOCNO can have many (4) values entered; so, one could say there are many isotopes that make up DOCNO, and ISOTOPE could be stored many times. This reasoning would indicate that ISOTOPE has many different quantities stored. This could lead to the design in Figure 3-20 which contains a grave error. The information is available as to what isotopes make up a document and in what documents an isotope is stored, but there is no connection to DOCUMENT to represent the correct member of MAKEUP. Therefore, the design of Figure 3-21 must be used, where the VALUES record is kept as intersecting data. With this schema, obtaining the information of an isotope's total quantity stored and the isotopic makeup of a document is easily retrieved, as seen in the two QRS commands illustrated in Figures 3-22 and 3-23.

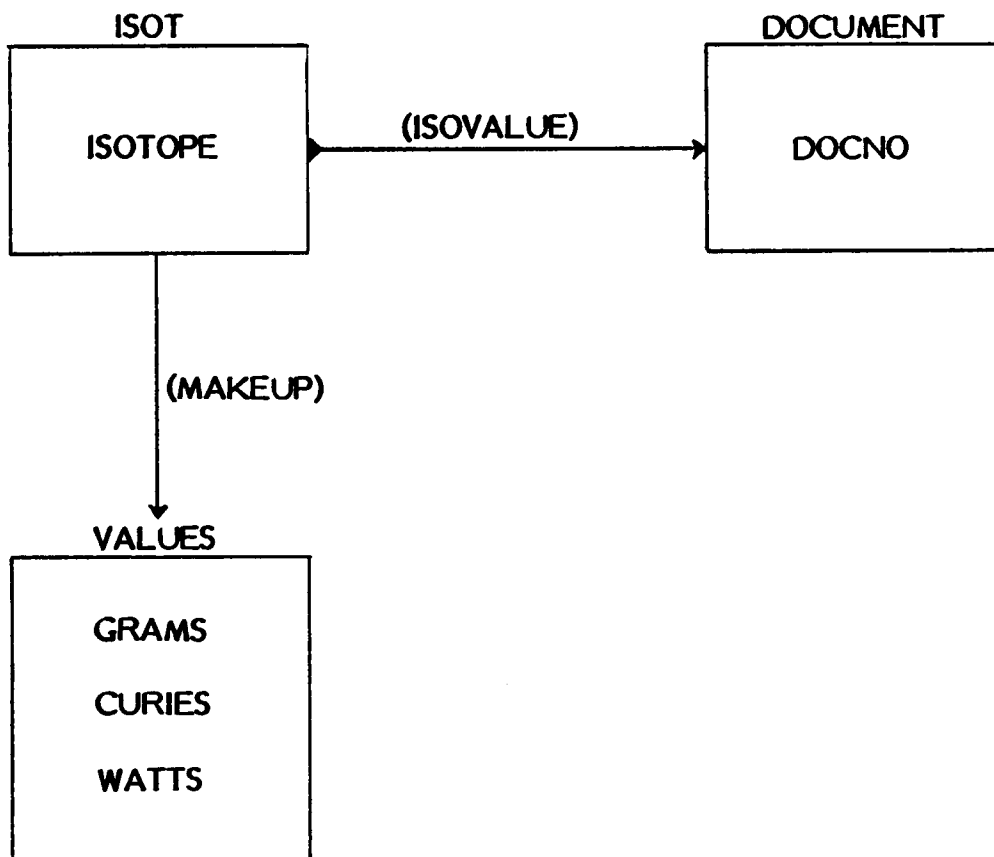


Figure 3-20. Incorrect schema to represent the M:N relationship between ISOT and DOCUMENT.

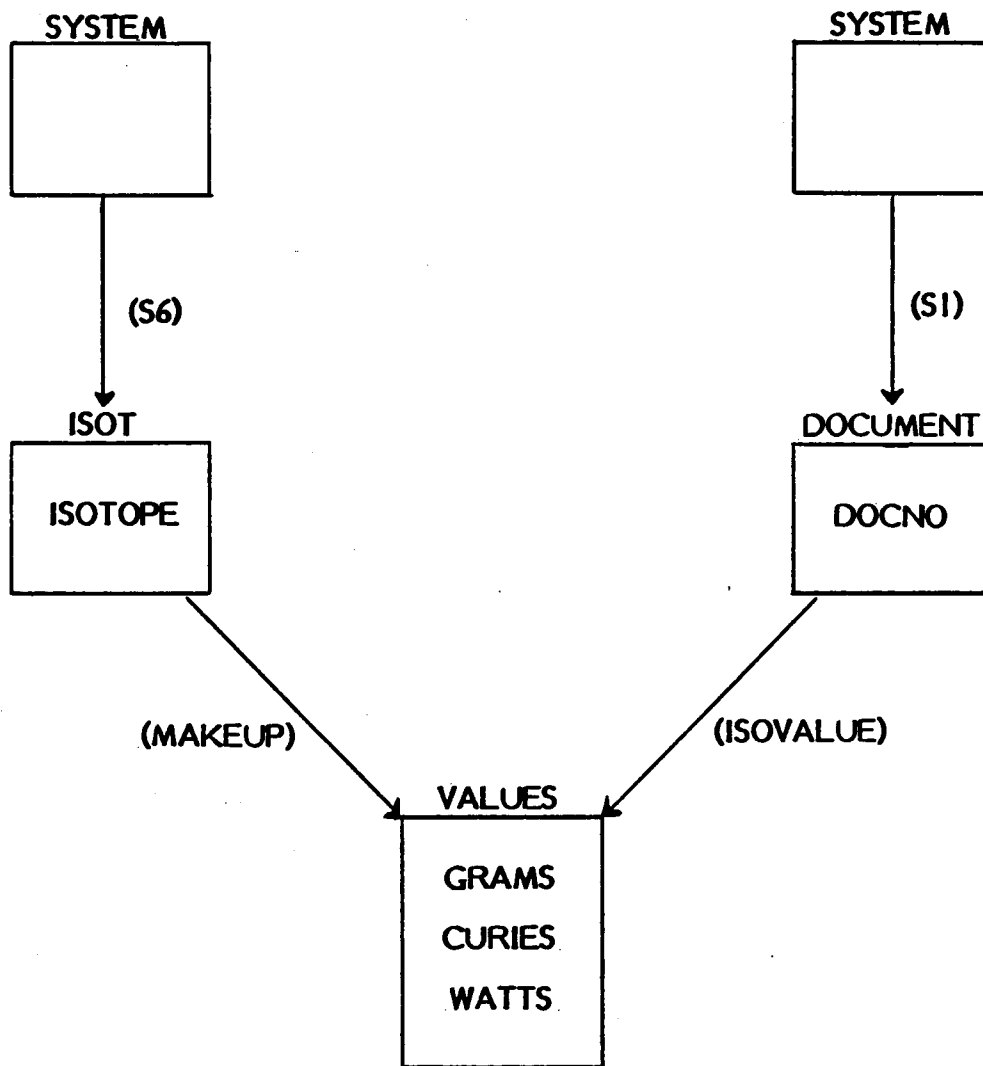


Figure 3-21. Correct schema using intersecting data.

→ LIST ISOTOPE GRAMS CURIES WATTS FOR ISOTOPE = "PU-239 "
THRU S6 MAKEUP

ISOTOPE	GRAMS	CURIES	WATTS
PU-239	0.490000	0.030723	0.000983
PU-239	0.015949	0.001000	0.000032
PU-239	0.009000	0.000564	0.000018
PU-239	0.359000	0.022509	0.000720
PU-239	1.000000	0.062700	0.002006

ISOTOPE	GRAMS	CURIES	WATTS
---------	-------	--------	-------

no of observations: 5

1.87395	0.117496	0.003759	sum
1.00000	0.062700	0.002006	max
0.00900	0.000564	0.000018	min
0.37480	0.023499	0.000752	ave

Figure 3-22. QRS command to obtain the total quantity of PU-239 stored.

→ LIST ISOTOPE GRAMS CURIES WATTS FOR DOCNO = "37602" THRU
SI ISOVALUE >MAKEUP

ISOTOPE	GRAMS	CURIES	WATTS
PU-239	0.490000	0.037230	0.000983
U-233	3.000000	0.284400	0.000824
AM-241	2.000000	6.599999	0.224400
U-235	57.20000	0.000126	0.000003

ISOTOPE	GRAMS	CURIES	WATTS
---------	-------	--------	-------

no of observations: 4

62.69000	6.92176	0.225468	sum
57.20000	6.59999	0.244000	max
0.490000	0.000126	0.000003	min
15.67250	1.730440	0.056367	ave

Figure 3-23. QRS command to obtain isotopic makeup of a particular document.

3.4 Implementation of the New Data Base

The implementation phase was threefold:

- (1) download the data file from a large-scale central computer to the micro computer;
- (2) convert data file to acceptable MDBS III format; and
- (3) develop DML application program to load the data into the data base schema.

The downloading process was a transfer of the data file from the mainframe to the micro computer by means of a communication software package. After completion of this step, the data was converted into a format that a DML application program could read. The last step was the development and execution of this DML program which read the data file and loaded the data item values into the data base.

CHAPTER 4

RESULTS AND CONCLUSIONS

4.1 Results of the Project

The principal result of this project was the implementation of a data base management system for management of the radioactive waste data stored at ORNL. This effort provided the user, the Chemical Technology Division (CTD), with a:

- (1) flexible,
- (2) easy to use, and
- (3) efficient method

of information management. These results are evident as illustrated by Table 4-1. Through comparison of the previous data processing system to the newly designed data base management system, it can be concluded that the new system is superior.

The table illustrates the flexibility of the system by indicating the use of QRS and menu-driven application programs to obtain results. QRS makes it possible for a user to query the data base for any combination of data items. Menu-driven application programs provide the user with a variety of options for requests.

The QRS software also substantiates that the system is easily used. Simple QRS statements and the ability to directly implement any conceptualized relationship enables the user to quickly learn the QRS facility.

This system is providing the users with a much more efficient method of information management. The time constraints of the previous system in

Table 4-1. Comparison of the new data base management system to the previous data processing system.

SUBJECT	CONSTRAINTS		OPERATION PERFORMED BY	
	PREVIOUS SYSTEM	NEW SYSTEM	PREVIOUS SYSTEM	NEW SYSTEM
Time required to obtain a regular report	1-2 days	1 hour	Y-12 Personnel after request is made	Application program executed by CTD at any time desired
Time required to obtain ad hoc results	Possibly 2 weeks	Minutes with QRS	Y-12 Personnel after explanation of request is communicated	CTD Personnel for any inquiry
Entry of data	1-2 days minimum 3 forms to complete 3 people involved	10 minutes 1 hardcopy form 1 person	CTD personnel Keypunch personnel Y-12 personnel	Menu-driven application program executed by CTD at their demand

obtaining ad hoc requests and the method of going through several procedures for data entry were both inefficient. Along with the efficiency of the micro based system, this system also provides the users with their own resident system. This improves efficiency and gives the users more control over their data.

4.2 Conclusions

Two major conclusions to be drawn from this project are:

- (1) micro computers can be effective information tools; and
- (2) the needs of the users are an important factor in data base development.

Micro computers can be applied to scientific data base problems with success. Data base management software available today enables programmers to develop systems for microcomputers as sophisticated as those running on large computers (Kruglinski 1983, p.2). Through the use of micro computers, managers can be provided with capabilities they have never had before as evidenced by the flexibility of QRS and menu-driven programs.

Another way in which micro data base management systems provide effective systems is that data base management software can significantly reduce the costs of software development. This can be done several ways:

- (1) the data base management system maintains an internal data directory (in the DDL), by keeping descriptions of data and relationships among the data items;
- (2) the data base management system lets the programmer access data values in a method appropriate to the request, rather than forcing the programmer to write a program that fits the data; and

- (3) a data base approach provides flexibility and economy in altering the initial system to meet new needs (Gagle and Koehler 1981, p. 122).

The development of a data base is predicated by the needs of the users (Kroenke 1983, p.15). These needs must be researched and evaluated before a data base can be an effective information tool.

4.3 Recommendations

A general recommendation to be made is that the effectiveness and capabilities of the micro computer and available data base management software should not be underestimated. Managers need to explore these capabilities for further scientific and business applications.

Future recommendations for this application include:

- (1) establishing an integrated, nation-wide data base of radioactive waste information; and
- (2) performing decay analyses on the waste data stored at ORNL.

There is increased interest for the development of a nation-wide, integrated data base of radioactive waste data. This system could be modified, if needed, to integrate an entire national data base. With the advent of hard disk drives of greater than 60 megabytes, this integrated system could be accomplished. Another advantage of the micro computer system enables this system to be transported easily to other sites in the nation for their own on-site implementation.

The second recommendation for this application involves the development of decay analyses on the data. This system now reflects management of radioactive levels (the CURIES data item) as recorded on the storage date, but it

currently does not have the capability to calculate and report the actual radioactive waste level at any other point in time. This results from the fact that radioactive isotopes decay over time, and the amount recorded at the time of storage is not present now. The ability to calculate isotopic decay for each isotope from its storage date would enhance this data base as a management tool. This idea is supported by Patterson (1984, p.59) as he states that managers are now able to provide results for any request, so the emphasis should now focus on the importance of the request. Users need to fully utilize the data bases that are provided. Managers should examine the value of the information they require their staffs to obtain, i.e., are the data necessary, are they being used, do they mean anything? We now have a tool which should be developed to meet the real needs of the users.

LIST OF REFERENCES

LIST OF REFERENCES

- Ashton-Tate. dBASE II. Culver City, CA: Ashton-Tate, Inc., 1984.
- Cardenas, Alfonso. Data Base Management Systems. Boston, MA: Allyn and Bacon, Inc., 1979.
- Date, C. J. An Introduction to Database Systems. Reading, MA: Addison-Wesley Publishing Co., 1977.
- DOE/EIS-004F. Final Environmental Impact Statement--Management of Commercially Generated Radioactive Waste. Washington, D.C.: U.S. Department of Energy, 1980.
- DOE Order 5820. Radioactive Waste Management. Washington, D.C.: U.S. Department of Energy, 1983.
- Gagle, Michael and Gary J. Koehler. "Data-Base Management Systems: Powerful Newcomers to Microcomputers." BYTE. November, 1981, pp.97-122.
- Jacobs, Donald. Personal interview with Donald Jacobs, 25 April, 1984.
- Kroenke, David. Database Processing: Fundamentals, Design, Implementation. Chicago, IL: Science Research Associates, Inc., 1983.
- Kruglinski, David. Data Base Management Systems--A Guide to Microcomputer Software. Berkeley, CA: Osborne/McGraw-Hill, 1983.
- Martin, James. Computer Data-Base Organization. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1977.
- Micro Database Systems, Inc. MDBS III. Lafayette, IN: Micro Database Systems, Inc., 1984.
- ORNL/CF-82/202. History of Disposal of Radioactive Wastes into the Ground at Oak Ridge National Laboratory. Oak Ridge, TN: Oak Ridge National Laboratory, 1982.
- ORNL/NFW-84/10. Procedures and Technology for Shallow Land Burial of Low-Level Radioactive Waste. Oak Ridge, TN: Oak Ridge National Laboratory, 1982.
- Patterson, William. "Corporations in Crises: Coming to Grips with the Information Age". Industry Week. 5 March 1984, pp.57-60.

Public Law No. 97-425. Nuclear Waste Policy Act. Act of 7 January 1983.

Title 10, CFR, Part 61. "Licensing Requirements for Land Disposal of Radioactive Waste". Code of Federal Regulations.

APPENDIXES

APPENDIX A

THE DATA ITEMS DESCRIPTIONS

<u>FIELD</u>	<u>DEFINITION</u>
ATNNO	Unique number assigned by CTD to waste (nuclide 1 - 6).
BUILDING	Building where the data may be stored.
CMTS1	General comments on the waste made by CTD. First of maximum 3 cards (CMTS2 & CMTS3 are also present).
CNTRTYPE	The type of container the waste is stored in. 1 - 55 gallon SS drum 2 - 30 gallon SS drum 3 - 4.5" wall concrete cask 4 - 6" concrete cask 5 - 12" concrete cask 6 - 55 gallon FE drum 7 - 30 gallon FE drum 8 - Wood or metal box 9 - Other A - GI can B - Plastic C - Dumpster D - None E - Cask (Shielded carrier) F - 74.5"L x 50.5"W x 38.5"H Metal Box G - 68"L x 54"W x 38.5"H Metal Box H - 88"L x 54"W x 54"H Metal Box I - 112"L x 68"W x 77"H Metal Box
CODE	A III indicates the record has had no updates made. A 222 means there have been updates made.
COMVOL	Combustable volume of the waste, in cubic feet.
DOCNO	Unique number given each record.
FTTOP	If waste is stored in a trench, LOCTYPE = "T", indicates the location of the waste in the trench, feet from the top of the trench.
FTZERO	If waste is stored in a trench, LOCTYPE = "T", indicates the location of the waste in the trench, feet from the zero end of the trench.
FULLAMT	If FULLIND = "X", this indicates amount in this full well.
FULLIND	If waste is stored in a well, this indicates if it is full or not. "X" means full, a blank means not full.

<u>FIELD</u>	<u>DESCRIPTION</u>
ID	Badge or social security number of the shipper who authorizes the disposal of the waste.
ISOTOPE	Up to 4 isotopes can be input for a record. The isotopic breakdown of the waste, input in decreasing amount. Principal isotope makeup is first.
LOCNO	Further describes the location.
LOCTYPE	Type of burial location. L - Landfill B - Building T - Trench W - Well R - Ravine A - Above ground
NBALES	If the waste is compacted, TREAT = "C", this indicates the number of bales after compaction. Otherwise, blank.
NONVOL	Noncombustable volume of the waste, in cubic feet.
NUCLIDE	Radioactivity waste classifications. 0 - Temporary storage 1 - TRU or U-233 (100mCi/Kg) - Retrievable Storage 2 - Uranium/Thorium 3 - Fission product 4 - Induced activity 5 - Tritium 6 - Beta - Gamma TRU or U-233 (100mCi/Kg) - Retrievable Storage 7 - Alpha TRU or U-233 (100mCi/Kg) Beta - Gamma TRU or U-233 (100mCi/Kg) 8 - Other 9 - Landfill
NUMCMT	The number of comment cards for this record, 0-3.
OPRTYPE	Operation type. B - Buried R - Retrievable

<u>FIELD</u>	<u>DESCRIPTION</u>
ORIGSITE	Organization or site from which the waste was received. 1 - ORNL 2 - Y-12 3 - ORGDP 4 - 176 - Other seldom used sites 177 - 500 - Off area. Numbers are assigned if there is a new customer.
POSTWT	If TREAT = "C", indicates weight of waste after compaction. Else, is blank.
POSTVOL	If TREAT = "C", indicates volume of waste after compaction. Else, is blank.
PREWT	If TREAT = "C", indicates weight of waste before compaction. Else, is blank.
PREVOL	If TREAT = "C", indicates volume of waste before compaction. Else, is blank.
QTY	Quantity of each isotope found in the waste. Input with ISOTOPE and UNIT.
REFNO	Number assigned to each record before DOCNO has been assigned.
RPTFLAG	Not sure of use.
SITELOC	Burial ground used. Burial ground 1-4 not used since 1940's. Burial ground 5 & 6 used now. Burial ground 7 to open soon.
STORDATE	Date storage of waste was performed in the field.
TREAT	Type of treatment process performed on the waste. C - Compacted N - Normal Incineration O - Other Blank - Nothing
UNIT	Either G for grams or C for curies. This indicates the unit used in quantity of the isotope.

<u>FIELD</u>	<u>DESCRIPTION</u>
UPCODE	Type of action that has been taken on the record. A - New action. (Record has had no updates) C - Updates have been made. D - Mark the record for deletion.
UPDATE	Date the record was updated. Used only if UPCODE = "C" and CODE = "222".
WASTETYP	Type of waste. BW - Biological Waste CE - Contaminated Waste DD - Decontamination Debris DS - Dry Solids SS - Solidified Sludge NC - Not Classified
WEIGHT	Weight of the waste in pounds.

APPENDIX B

THE DATA ITEMS DEFINITIONS

<u>FIELD</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>DOMAINS</u>
ATNNO	Character	5	Numeric
BUILDING	Character	6	Alpha-numeric, with "-"
CMTS1 CMTS2,CMTS3	Character	80	Alpha-numeric
CNTRTYPE	Character	1	Alpha-numeric
CODE	Integer	3	Numeric
COMVOL	Integer	3	Numeric
DOCNO	Integer	6	Numeric
FTTOP	Character	3	Numeric
FTZERO	Character	3	Numeric
FULLAMT	Integer	3	Numeric
FULLIND	Character	1	Alphabetic
ID	Character	9	Alpha-numeric
ISOTOPE	Character	7	Alpha-numeric, with "-"
LOCNO	Character	5	Alpha-numeric
LOCTYPE	Character	1	Alphabetic
NBALES	Real	4	Numeric with decimal
NONVOL	Integer	3	Numeric
NUCLIDE	Character	1	Alphanumeric
NUMCMT	Integer	1	Numeric
OPRTYPE	Character	1	Alphabetic
ORIGSITE	Integer	4	Numeric
POSTWT	Integer	7	Numeric

<u>FIELD</u>	<u>TYPE</u>	<u>LENGTH</u>	<u>DOMAINS</u>
POSTVOL	Integer	7	Numeric
PREWT	Integer	7	Numeric
PREVOL	Integer	7	Numeric
QTY	Real	9	Numeric, scientific notation
REFNO	Integer	2	Numeric
RPTFLAG	Integer	1	Numeric
SITELOC	Integer	1	Numeric
STORDATE	Character	8	Converted to a date, i.e., mm/dd/yyyy.
TREAT	Character	1	Alphabetic
UNIT	Character	1	Alphabetic
UPCODE	Character	1	Alphabetic
UPDATE	Character	8	Converted to a date, i.e., mm/dd/yyyy.
WASTETYP	Character	2	Alphabetic
WEIGHT	Integer	5	Numeric

VITA

In the spring of 1980, Ms. Pamela M. Mason received a Bachelor of Science degree in Home Economics from The University of Tennessee, Knoxville. In the winter of 1981, she began study toward a Master of Science in Computer Science to be received in August 1984.

During her three-years of study, Ms. Mason worked as a Programmer with the Computer Sciences Division at Oak Ridge National Laboratory. Her work there included the design of computer code to be used with human exposure analysis systems for health risk assessments and the design and implementation of the project reported in this thesis. Ms. Mason has also held the position of graduate teaching assistant in the Computer Science Department. Currently, she is employed by the MAXIMA Corporation located in Oak Ridge, Tennessee as a Systems/Programmer Analyst.

Ms. Mason is a member of the Phi Kappa Phi, Beta Gamma Phi, and Omicron Nu Societies, the Society for Women Engineers, and the Association for Computing Machinery.