

Analysis and Conversion Tools for Euro Currency Migration

RAINER GIMNICH

IBM Global Services, EMU Transition Services, D-70548 Stuttgart, Germany
gimnich@de.ibm.com

SUMMARY

The introduction of the single European currency (the euro), at the beginning of 1999, has presented a number of challenges, business opportunities and threats to a huge number of European organizations, companies and public administration offices. Euro migration is typically driven from business strategy and business process reengineering. However, the IT issues in implementing the companies' euro strategy have turned out to be the key factor of the quality and success of the transition.

Tools have been advocated as major accelerators in view of the complexity, resource constraints, and shortage of experts in many euro transition projects. Here, we concentrate on the tools aspects from a software reengineering point of view. We look into potential conversion strategies and methodology support. We derive tool requirements and consider the analysis and conversion tasks to be supported in more detail.

1. INTRODUCTION OF THE EURO

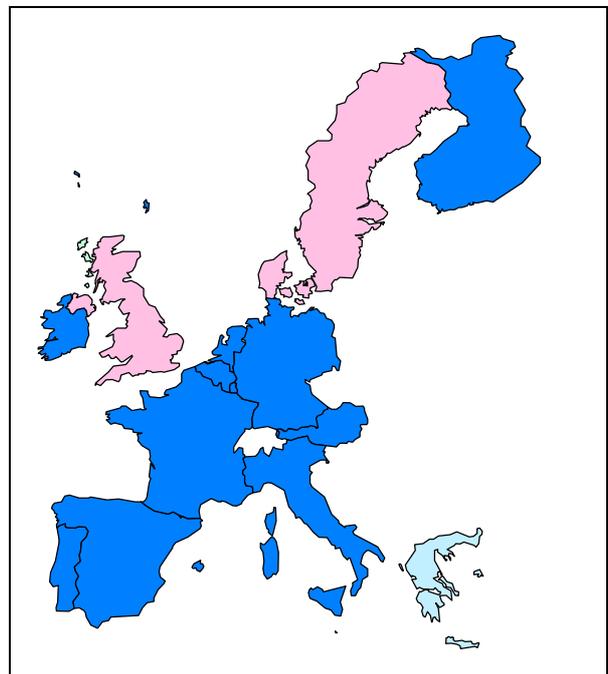
At the European Union (EU) summit in Brussels in May 1998, the presidents of the EU member states have decided on the 'first wave' of countries to build the European Monetary Union (EMU). Now, 'Euroland' extends over 11 countries: Austria, Belgium, Finland, France, Germany, Ireland, Italy, Luxemburg, the Netherlands, Portugal, and Spain (see [Figure 1](#)). From the remaining 4 EU countries, 3 had previously decided to consider joining EMU at some later point in time (Denmark, Sweden, and the U.K.) and one country has not yet met the qualifying criteria (Greece).

We must keep in mind that the number of EU member countries is likely to increase. Several more countries, especially from Eastern Europe, have applied for membership. Consequently, the range of EMU countries is likely to be extended in the future.

The migration timetable to EMU had been defined by the EU years before, outset already at the Maastricht summit in 1992. The migration cornerstones are depicted in [Figure 2](#).

The European Monetary Institute (EMI) was founded in 1994 and has, on 1 July 1998, turned into the European Central Bank (ECB), which has been responsible for the European monetary policy since 1 January 1999.

1 January 1999 has been the 'birth date' of the euro. Fixed exchange rates between the euro and the EMU



[Figure 1](#): EMU member countries

national currencies had been defined and published the day before. There are 11 EMU countries and 10 different euro rates. This is because Belgium and Luxemburg have had a monetary union since 1922: the Belgian franc and the Luxemburg franc are equally valued and interchangeable.

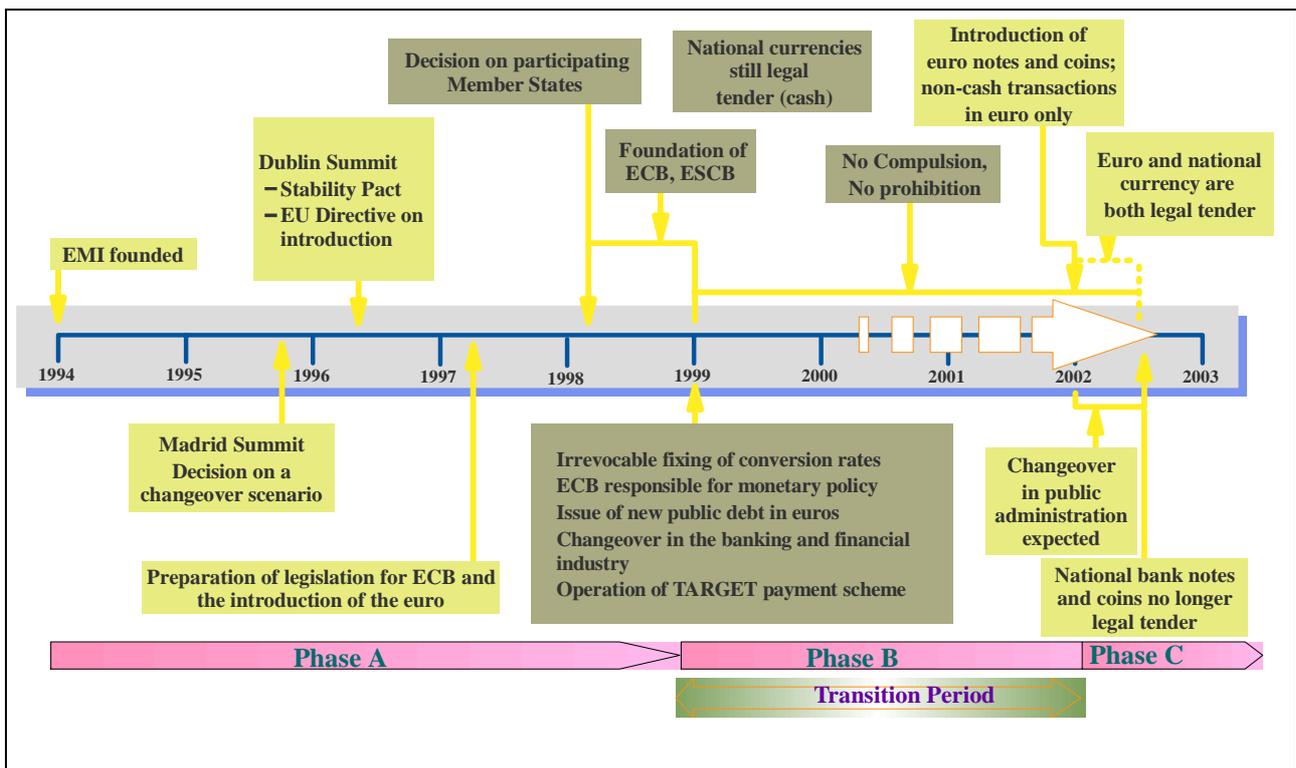


Figure 2: EMU timetable

Since 1 January 1999, the euro is used at the stock exchange, so investment banking applications had to be migrated (or extended) in the first place. For all other businesses, the euro and the national currencies (NCs) are used in a 'no compulsion, no prohibition' manner for up to 3 years. During this timeframe, each organization must complete its migration to the new currency. Legally, all NCs of the EMU member states have become subdenominations of the euro. Financial transactions in these countries may be performed in euro or in NC, where strict rules are defined for conversion, rounding, and accuracy.

In the transitional period, practically all organizations will have to deal with both the euro and their NC to some extent. This leads to enormous efforts in the migration, in addition to the move to the new currency. Therefore, a number of migration scenarios are usually developed for an organization in order to select the most suitable one once the business strategy is sufficiently defined.

According to plan, on 1 January 2002 at latest, the euro bank notes and coins will be introduced. The production of 'physical euros' for more almost 300 million people is an enormous logistic task and will be performed from mid 1998 to mid 2001. Some weeks or months after 1 January 2002, at latest on 1 July 2002, the national bank notes and coins will be withdrawn from circulation and cease to be legal tender. All systems will have to work smoothly in a single currency again which is now the same currency as in 10 neighbor countries.

From this framework scenario, we see that the information technology (IT) in the EMU member countries will certainly be most afflicted by the move to the euro [1, 2, 5]. However, all countries dealing with 'Euroland' will have to take care of internal changes due to the fact that their customers or suppliers will pay or charge in either euro or their NC for a 3-year transition period and only in euro after it.

Many of the problems arising in dealing with (at least) 2 currencies in the transition period can be solved by using powerful converter mechanisms. These must not be straightforward, specialized multipliers, but rather highly adaptable 'engines' which are capable of dealing with the EU regulations on accuracy and rounding. We try to illustrate this by the example in Figure 3.

In this example, we show conversions in 'round trips' from the euro currency to the German currency (Deutsche Mark, DM) and back to euro in the upper table, and from DM to euro and back to DM in the lower table. In the first case, we convert from a relatively coarse-grained currency (euro) to a more fine-grained currency (DM), which has a 'precision factor' (conversion rate) of almost 2. Thus, the recomputation of the original value is always possible, without any imprecision due to rounding. This desirable situation holds for the individual amounts as well as for the sum of amounts.

In the second case, we convert DM into the more coarse-grained euro. For two of the three individual

Conversion rate example: 1 EUR = 1.95583 DM

Product	Premium in EURO	Conversion in DM	Reconversion in EURO
Premium - Liability	60 EUR	117,35 DM	60 EUR
Premium - Accident	50 EUR	97,79 DM	50 EUR
Premium - Transportation	10 EUR	19,56 DM	10 EUR
		SUM = 234,70 DM	SUM = 120 EUR
Total Premium	120 EUR	234,70 DM	120 EUR



Product	Premium in DM	Conversion in EURO	Reconversion in DM
Premium - Liability	60 DM	30,68 EUR	60,00 DM
Premium - Accident	50 DM	25,56 EUR	49,99 DM
Premium - Transportation	10 DM	5,11 EUR	9,99 DM
		SUM = 61,35 EUR	SUM = 119,98 DM
Total Premium	120 DM	61,36 EUR	120,01 DM



Figure 3: Example of conversion and reconversion

amounts we are unable to recover the original DM amounts. The sum of the converted amounts is 199.98 DM, so we lose 0.02 DM. Moreover, if we convert the DM sum into euro and then back again, the result is 120.01 DM. There we gain 0.01 DM.

We see from this simple example that the sum of converted values may differ from the converted sum of values. And there are situations where it is simply impossible to regain the original value by reconversion: the 'round trips' fail. Therefore, converters must work with a higher internal precision than multipliers and they must be able to retain the gains or losses of conversions, preferably in both currencies entailed.

2. IT ISSUES IN EURO MIGRATION

For clarity, we distinguish between euro 'inherent' properties from the currency definition and the impact of introducing the new currency into software systems.

2.1 Euro inherent properties

According to the EU Council Regulation No. 1103/97 [3] from June 1997, the following rules are defined and mandatory in all member states:

- 'The Conversion rates shall be adopted as one euro expressed in terms of each of the national currencies of the participating Member States. They shall be adopted with **six significant figures**.

- The conversion rates shall **not be rounded** or truncated when making conversions.
- The conversion rates shall be used for conversions either way between the euro unit and the national currency units. **Inverse rates** derived from the conversion rates **shall not be used**.
- Monetary amounts to be converted from one national currency unit into another shall **first be converted into** a monetary amount expressed in the **euro** unit, which amount may be rounded to **not less than three decimals** and shall then be converted into the other national currency unit. No alternative method of calculation may be used unless it produces the same results.'

And further in this regulation:

'Monetary amounts to be paid or accounted for when rounding takes place after a conversion into the euro unit ... shall be **rounded up or down to the nearest cent**. Monetary amounts which are to be paid or accounted for which are converted into a national currency unit shall be rounded up or down to the **nearest sub-unit** or in the absence of a sub-unit to the nearest unit, or **according to the national law or practice** to a multiple or fraction of the sub-unit or unit of the national currency unit. If the application of the conversion rate gives a result which is exactly half-way, the sum shall be rounded up.'

From these specifications we see that the euro arithmetic rules are clearly defined and unique. Even in applica-

tions relying on multi-currency systems, a correct euro behavior is not immediately guaranteed, but needs to be checked and added if necessary (and feasible).

Most applications in insurance, government, telecom and other industries, are 'currency-blind': the currency is not even specified. It is clear from the 'context' that the amounts handled in the applications are in national currency.

In implementing the euro migration, all amount fields must be checked. If they are not marked by means of currency indicators or naming conventions, a significant effort should be allocated to the analysis.

Besides, in most solutions, a currency code needs to be introduced to avoid ambiguities in the interpretation of numbers. Here, the word 'euro' may be used, or its ISO code 'EUR', or the euro currency symbol.

2.2. Euro collateral IT effects

The following considerations hold for mono-currency (and even 'currency-blind') applications before or during euro migration:

Amount field detection and completeness: How to find all relevant amount fields? Where to start searching? How to achieve completeness? What will happen if one relevant amount field remains undetected and/or untreated in the migration?

Amount field expansion: While in Year 2000 projects field expansion could be avoided in most situations, by using a century-window and re-interpreting the 2-digit year fields, now the expansion effort cannot generally be avoided. This holds unless every single field has at least 2 decimals and a 'sufficient' number of digits before the decimal point (or comma). Practice shows that we have to be careful:

- Some currencies, such as the Belgian and Luxemburg franc, the Italian lira, the Portuguese escudo and the Spanish peseta, have *no decimals*. Here, a general transformation of the kind **DEC(x,0) --> DEC(x,2)** may be necessary. This entails not only the programs or the include files / copybooks, if declarations are handled there. It also entails the databases, files, screens, maps, transaction systems and all other 'subsystems' where amounts occur.
- Even in '2-decimals currencies', there are applications which, for simplicity, handle only as many decimals as required. For instance, in a German car insurance application, many computations rely on *one decimal*. To use more decimals, even if only internally in the program, would have caused a programming overhead. Now,

if these variables are to hold euro values in the near future, a transformation of the kind

DEC(x,1) --> DEC(s,2) with $x \leq s$ may be necessary, assuming that *s* represents some company-wide standard length, to avoid any further trouble.

- There are currencies, presently in EMU only the Irish punt, which have a granularity more coarse than the euro. Though we have 2 decimals here, the number length as a whole may be insufficient to hold more the fine-grained values from the euro. Here, transformations of the kind **DEC(x,y) --> DEC(x+1,y)** may be required. An evaluation of present and future maximum values will be helpful for decision-making.

Literals: In many cases, amounts are hard-coded in the application programs. For instance, statements of the kind **IF amt_1 < 1000 THEN ...** appear quite often. Here, the threshold value is simply used as a constant in the program: no symbolic constant, no variable declaration, no external amount table read.

There may also be string-type literals (or substrings) denoting the currency ('DEM', 'USD', ...) or its symbol ('\$' , ...). These often appear in declarations pertaining to the user interface. They must be detected and potentially updated as well.

Dual-currency handling on the user interface: Since there is a transitional period, some customers may wish to know the equivalent values of their contracts in euro. Or they want to check new euro tariffs against their former NC contracts. Both currencies must be used for at least a subset of amount fields in the user interface, no matter if the user interface is a printed letter to a customer, or a panel of an inhouse application, or a telephone call to a helpdesk, or an Internet form.

In any case, it must be decided whether a permanent dual display is to be implemented (with all space-related consequences) or a toggling facility to switch back and forth from one currency value to the other (with potential space problems as well). Both of these solutions can be supported by dedicated tools.

Historical data: There is a practical tendency to convert historical data upfront (or by need) to the euro as well, in contrast to convert euro values back to a soon-dying national currency. However, given the size of historical databases in huge organizations and the effort required for data migration, the technical impact must be clearly evaluated before business decisions are taken.

This list of collateral effects of the euro introduction is by no means complete. It is meant to give a flavor of the migration planning environment that needs to be considered.

2.3 Migration strategies

The most obvious strategy to adopt is to perform a **big bang** conversion at one point in time, i.e. one long weekend sometime between 1999 and 2001. Before the big bang, every amount is stored in NC and there occurs some forward conversion to euro. After the big bang, everything is stored in euro, with the potential need for some backward conversion to NC.

The big bang is often the least expensive approach, as it requires no bridges between the 'old' and the 'new' world. However, the IT risk is high, and in larger IT environments, a big bang approach is simply not feasible: even a week's shutdown would not suffice to migrate the data and make the migrated programs, user interfaces, subsystems etc. productive and to test the *whole* 'migrated world'.

To render the migration feasible, we can perform a **phased migration**, i.e. a sequence of 'small bangs', giving the application status after each phase (or subphase) into production after sufficient testing. However attractive and less risky, this approach means partitioning the whole application environment in a 'most suitable' way and building bridges between the migrated packages and the not-yet-migrated world. After all is migrated the bridges need to be removed.

Equally important is the fact that phased migrations may be necessary to apply to each 'migration target' (e.g. euro correspondency capability, amount field expansion, euro tariffs introduction, final data migration) with completely different phase structures. Besides, the planning effort and project management effort is enormous. However, this type of migration is the most preferred one, probably because there are no viable alternatives.

Dual currency handling on the whole is a different possibility. One might double the existing application system and adapt and use the copy for euro purposes only. This approach requires a lot of resources: machine time and space to handle the 'double load', but predominantly development, maintenance and management time. So this is an expensive and seldom used way of handling the transition period and later dropping the NC version of each application. However, it entails low risk.

There are a number of additional euro migration strategies, often used in combination with each other or with one of the three kinds depicted above.

For instance, one can decide to introduce standard software packages, e.g. ERP products, to replace (some of) the existing application functionality and obtain euro readiness alongside. To be able to handle euro requests in the meantime, a euro correspondence capability could be implemented with limited effort on top of the existing applications. Here, the company would look 'euro

ready' to the outside, but use the time to implement some pre-fabricated IT functionality. This implementation must be feasible long before the end of 2001, to allow enough time for testing and 'take-over'.

In the following section, we will show how these IT related issues are embedded in a more general methodology, starting from business requirements and continuously interlinking business requirements and IT requirements to achieve the desired transition results.

3. THE IBM EUROPATH METHODOLOGY

In order to provide a generally applicable, end-to-end methodology, IBM has developed a dedicated path, called **EuroPath**, within our project management and support environment WSDDM (Worldwide Solution Design and Delivery Methods). This path is accessible to all IBM services experts and project managers on a global basis. Its top-level structure is shown in [Figure 4](#).

IBM EuroPath conceives the Business Strategy Impact Analysis and the Process and IT Impact Analysis phases to be completed before the Business Transition and the IT transition are performed in parallel. The latter is structured into Detailed IT Planning, IT Solution Generation, and IT Solution Validation. The Business Transition and IT transition branches are reconciliated in a Euro Solution Deployment phase. This top-level phase structure is detailed into task structures and activities with their associated deliverables. Also, on the lower levels, more interdependencies between the tasks become evident.

The methodology knowledge within EuroPath has been achieved from several complex euro projects, and it is continuously enhanced by new relevant project experience. This methodology provides the basis for the functional requirements within the IBM Euro Tools Portfolio.

4. THE IBM EURO TOOLS PORTFOLIO

In view of the necessary activities, structured in EuroPath and applied to a specific organization to be transitioned, the need for effective tool support is apparent. These are the major benefits expected from 'euro tools' to be applied predominantly in the IT related phases, i.e. in the Process and IT Impact Analysis phase and in the 3 IT transition phases:

- **Productivity:** The scarce human resources and expert skills are used for higher-level tasks of applying the tools in order to quickly achieve project results. For instance, it makes no sense to employ programming resources to write converter modules or scanners for code or data analysis.

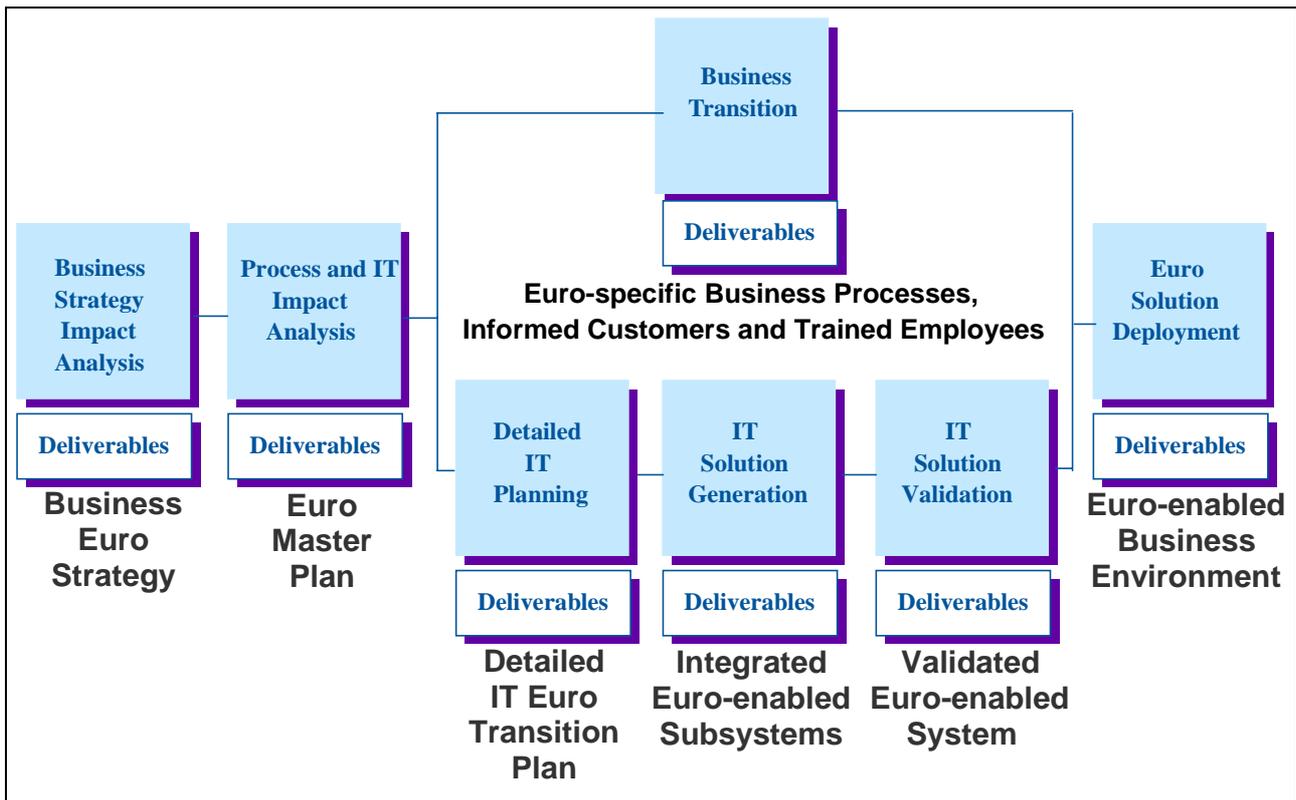


Figure 4: IBM EuroPath

By using tools, the scheduled end-times of migration projects are more reliable. In general, we have to face a rapidly increasing shortage of skilled personnel for euro projects in the years 1999-2002.

- **Costs:** In view of the perceived sizes of our customers' (and our own) euro projects, and the relatively low tool prices, the savings potential is enormous. This applies to analysis effort, coding effort, testing effort, and even maintenance effort. However, we have to bear in mind that no tool offers any complete automatism, human skills are still required but they may be dedicated to non-automatable work.
- **Quality:** As the IT analysis/migration is supported/generated by Euro tools, their results are more reliable than manual work or ad-hoc programming, and the need for white-box testing is reduced.
- **Conformity:** By using tools, the code changes are implemented in a uniform way. Therefore, the maintenance effort of the application is reduced, and further changes to the application can more easily be applied. The documentation effort is reduced alongside.
- **Control:** The usage of tools has the positive side-effect of introducing implementation guidelines. For instance, the use of a suitable converter product

guarantees the same coding strategy for conversions on whatever platform/language is used. Besides, the conversion parameters can be held under central control for the company-wide changes.

- **Timing:** According to our project experience, the use of tools allows for more appropriate effort estimation and an improved scheduling of the conversion tasks. We have seen euro projects which would not have been completed before 2002 without a dedicated, customized combination of tools.

This latter observation leads to the point of evaluating tools, building up a collection of some 20 or 30 powerful tools for different purposes and environments in order to select an optimal subset of these tools for the client's purposes and to effectively customize and use the toolset.

Already in 1997, we have set the **evaluation criteria** for the euro tools to be admitted to our portfolio. The main criteria are functionality and environment.

Functionality denotes the areas of support for IT euro project activities required in practice and documented within EuroPath. For instance, in the Process and IT Impact Analysis phase, the major IT activities to be supported by tools are: inventory analysis, amount field search, cross-referencing, partitioning/packaging and effort estimation. Later, in the IT Solution Generation phase, the activities include: program changes in field

types, field values and processing logic, file/database changes, process changes (JCL), user interface changes, converter implantation, test planning, etc.

Environment means the technical basis on which the tools are to operate, i.e. the platform (development platform, deployment platform), programming languages, DBMS/file systems, transaction management system, etc. The development platform (operating system where the tool runs) may differ from the deployment platform (target system which the tool has to serve). For instance, a tool may run on a Unix machine and analyse or change code being transferred from/to an MVS host. For another instance, a tool may include a component running on a PC operating system to define parameter files, which are then transferred to a number of other platforms to be used by other tool components installed there.

Functionality and environment are clearly the most important evaluation criteria, but there are others as well: status (product, internal tool, tool included in services, ...), maturity (used in practice for several years, still in development,), pricing model, interfaces/extensibility, customer references, support capabilities, and many more. All of these factors help us to precisely define the characteristics of each tool and position it in a multi-dimensional portfolio.

The resulting IBM Euro Tools Portfolio consists of IBM and partner tools, and it is used as background knowledge in customer calls. One of our services is a workshop (Tools Feasibility Study) to identify the optimal tool environment for an existing or planned euro project. Then the introduction, installation, customization and usage of the tools are usually performed within service contracts. These may include the employment of IBM Conversion Centers. This does not preclude the choice to use the tools installations already present at the Conversion Center, or to choose an in-house installation of the tools and have the Conversion Center perform all or part of the migration work.

From this approach it is obvious that the tools in our portfolio must meet the highest quality standards also in terms of capacity, performance and stability, running analyses and conversions of several hundred programs at a time.

The portfolio itself needs to be continuously enriched and updated. There is a subset of 'kernel tools' which is intended to be solid. These tools are further developed and used in service projects, they can easily be integrated, and they will remain in the portfolio. As new euro tools will come up, these will be evaluated against the given criteria and then confirmed or rejected for the portfolio. Tools, other than kernel tools, which are already in the portfolio may turn out to fail in certain situations, for various reasons, and may consequently be dropped from the portfolio.

In the following section, we will look into the functionalities of some of the analysis and conversion tools in the portfolio. We do not intend to present the complete portfolio nor the complete descriptions of some of the tools. Rather, the functionality overviews shall give some insight into the nature of euro tools and their usage.

4. ANALYSIS TOOLS FOR EURO MIGRATION

Analysis tools are required in various 'granularities': High-level analyses will be requested on large application environments with tens of thousands of programs, databases, files, JCL procedures, etc., yet these analyses may have to be performed down to the variables level. 'Medium-size' analyses will comprise partitions (packages) of several hundred elements. 'Fine-grained' analyses will be performed on program level, looking into each variable definition and its usage, along with data flow and control flow properties, down to the functionality of 'program slicers' [10].

Phase 2 of EuroPath (Process and IT Impact Analysis) has a substructure of activities to be performed, which is depicted in [Figure 5](#). There we see that on the IT part of that phase, the major work is on IT inventory analysis, pattern matching, IT impact analysis, partitioning, and estimating. Tool support for this phase is strongly required unless a suitable and up-to-date repository is available within the organization. Many repositories we have seen did not contain information in sufficient detail in order to assess the company's euro IT impact.

As such wide-ranging analysis tools are mainly requested in mainframe environments, the tool itself must be capable of handling the vast amounts of parse units and analysis results to be encountered. This is why such tools are often required to have MVS as their development *and* deployment platform. The source components to be captured must include:

- programs (in COBOL, PL/I, Assembler, ...),
- copies / includes,
- load modules,
- jobs,
- on-line transactions,
- data files,
- screens,
- reports.

As far as the inventory-taking functions are concerned, similar approaches have already been required for Year 2000 ('Y2K') projects [7, 8]. Generally, the analysis approaches which have successfully been used in the Y2K context before, can now be re-used. These include effort estimation approaches such as those based on work-breakdown structures [4].

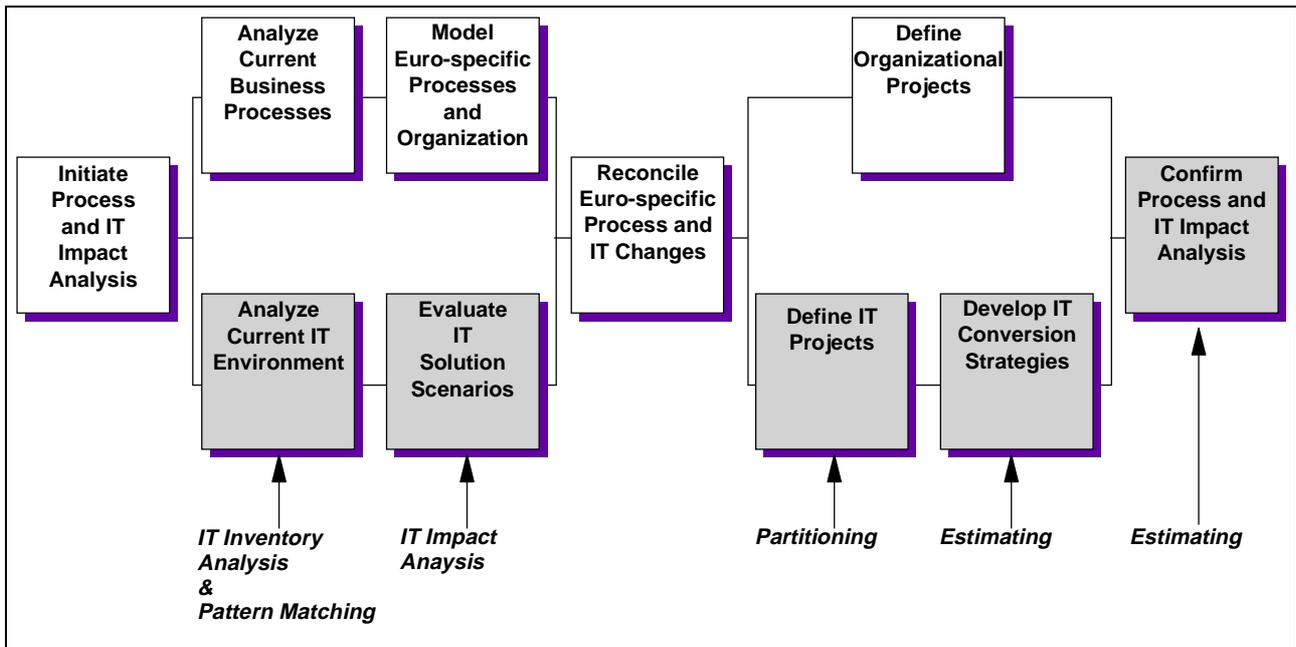


Figure 5: Process and IT Impact Analysis activities and IT analysis functions

It is important during inventorying, that the **inter-dependencies** between the relevant analyzed items are recorded, e.g. caller/called relations, variable/source file relations, source file/job relations, source file/transaction relations, job/dataset relations, etc. Also during inventory analysis, missing source code units have to be detected.

The analysis processes following inventory analysis, e.g. pattern matching, will have to be altered for euro project purposes. These now have to apply to amount fields (rather than date or year fields) and are technically denoted as **CSVs (currency sensitive variables)**. Unfortunately, CSVs are less structured and less standardized than DSVs (date sensitive variables). With CSVs, variations in naming as well as in type or picture declarations are enormous: numbers of various lengths, with or without decimals, packed or not, strings of various lengths, etc.

Figure 6 shows examples of amount declarations in PL/I and COBOL. Here, name patterns such as ***BETRAG*** or ***SUM*** would hit, but often, type/picture and even comment information is necessary to obtain a satisfying set of CSVs to start with. The predominant type pattern we have found consists of up to 18 digits (the COBOL limit) with or without suppression of leading zeros ('Z...') and including up to 6 decimals. Among these declarations are the many variables, e.g. holding insurance sums, which are declared with no decimals at all, e.g. **PIC 9(7)**. And there are the many variables which hold 'encrypted' values such as amounts in thousands of the local currency or special values indicating an exceptional situation rather than the value itself, e.g. **0.01** to indicate that a legal standard value applies which is stored somewhere else.

In order to perform the activities of inventory taking, pattern matching (for a starter-set of CSVs), and further analysis, e.g. for partitioning or effort estimation, we have chosen the tool **IBM Redevelopment Assistant (RA)** and enhanced it for euro purposes into **RA-Euro**. Its basic architecture is depicted in Figure 7.

The source code to be analyzed may entail all components in scope of the customer's environment in a number of (commonly used) programming languages. The source code is parsed against a given information model, and the results are stored in a DB2 repository. Now, general application understanding functions on the source inventory may be performed. So far, the approach is not euro-specific.

Currency inspection is indicated in the lower part of Figure 7. Currency patterns are specified as regular expressions, which are matched against the repository contents on variable declarations and uses. In iterative steps, a 'seed file' (base file of definitely detected CSVs) is derived. This may be used for further analysis, e.g. currency impact analysis. If a detailed data flow analysis is requested within the currency impact analysis, we usually export the seed file to IBM EuroCheck (see below). Further, partitioning analysis may be performed to suggest technically optimal packages (these must then be consolidated with semantic requirements). Also, effort estimation may be initiated, and - if the seed file is deemed sufficient - it may be handed over to conversion. For instance, it may be exported to IBM OSCAR-Euro (see next section).

The tool **IBM EuroCheck** is used for CSV identification and propagation. Its main feature is the iterative and interactive finding of impacted CSVs from

```

DCL 1 BDLISTR
...
5 BETRAG PIC 'ZZZZZ9V.99',
...;

DCL LASY0501 FIXED (15,2) /* SALDO */

DCL 1 LAST,
...
2 LAST0401 CHAR(12), /* NETTOBETRAG */
...
2 LAST1102 CHAR(6), /* BANKSPESEN */
...;

DCL PRUEFBETRAG DEC FIXED (15,2);
DCL SAMMELBANKSP DEC FIXED (15,2);
DCL LISTE3_SEISUM DEC FIXED (15,2);

DCL BANKSPESEN PIC 'Z9V.99';

01 ZHGGESB1 PIC 9(7)V9.
01 K2BEIT PIC 9(12)V9.
01 BSP1SUM PIC 9(5).
01 EZGPRS PIC 9(8)V999.

```

Predominant Type Pattern: Z(z)9(x)V9(y) with z+x+y <= 18 and y <= 6

Figure 6: Amount declaration examples

a given seed file, with the intention to identify *all* CSVs pertaining to the given set of programs. Thus, EuroCheck can be used in conjunction with RA-Euro (in large environments) or it may be used stand-alone. EuroCheck is an interesting example of a tool which was developed in the euro context and later adapted for Y2K purposes, and not vice-versa.

The architecture of EuroCheck is shown in Figure 8. In a pre-processor stage, the source code is transformed

into a language-independent form. Procedural code and variable declarations are separated and the captured information is stored in the respective parts of the EuroCheck repository.

The Variables part of the repository may be used for CSV identification, in conjunction with currency patterns. The result will be a first seed file, which is used and enhanced in further analysis steps. The seed file may as well be 'imported' from an RA-Euro analysis

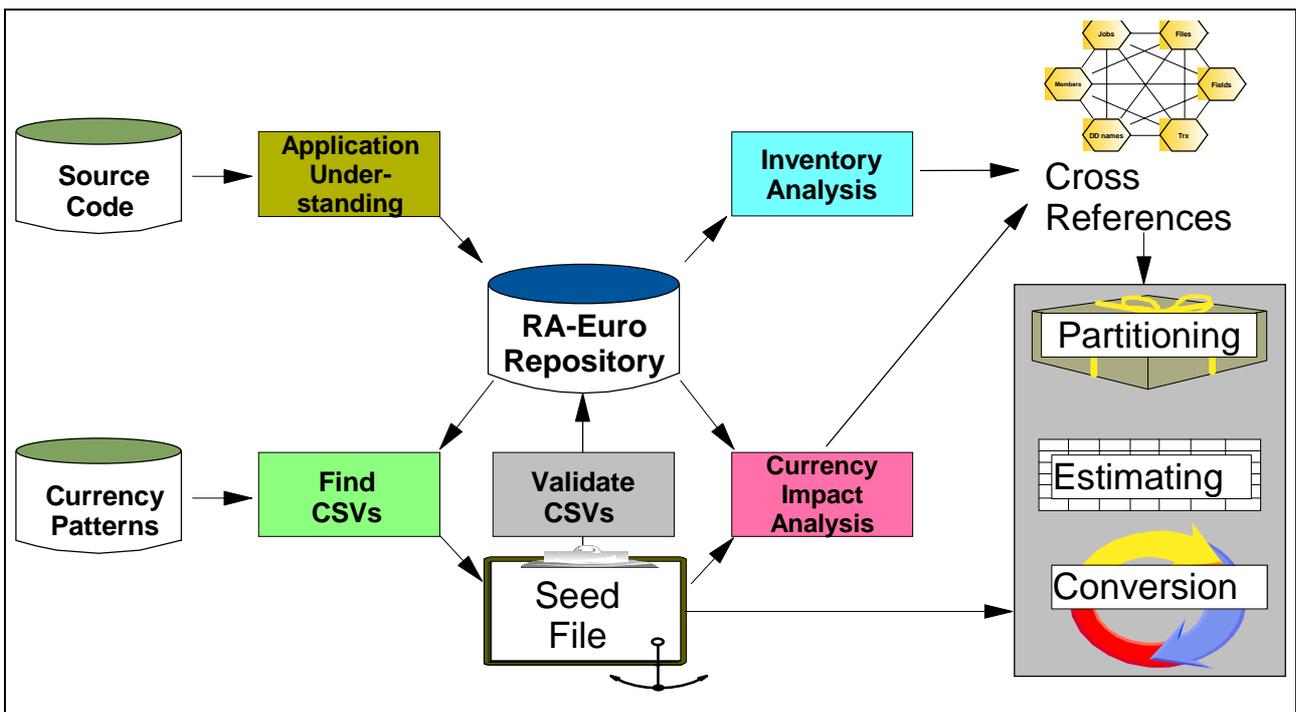


Figure 7: IBM Redevelopment Assistant - Euro (RA-Euro)

or from analysis results of other tools, or it may have been built manually.

An important seed file extension which can optionally be used is the concept of maximum target values (MTVs). An MTV is the maximum value a CSV can have after euro transition. Such information is often useful in checking whether impacted fields must be expanded or not. This feature has been used extensively in our projects in Ireland.

The most important analysis stage in EuroCheck is the propagation. Here, each CSV already in the seed file is checked for dataflow relationships with other variables or literals. After each iteration step, the user may accept or reject the findings, or even manually insert or delete other CSVs.

At the end of EuroCheck processing various reports can be produced: a Metrics Report, a Statement Analysis Report, and a Currency Sensitive Variables Report are available. In addition, the CSV result file of EuroCheck can be exported to other tools, such as IBM OSCAR-Euro (see below).

5. CONVERSION TOOLS

The Tool **IBM OSCAR-Euro** enables the user to perform and control the euro migration process program per program. OSCAR means ‘**On-line SCA**

Repository’ while SCA stands for ‘**Source Change Assistant**’.

Both OSCAR and SCA were developed in the Y2K context. SCA provides the code change functions, e.g. for century-windowing in the Y2K case and for amount conversion (via converter calls) in the euro case. OSCAR is a smart editor to initiate such changes and to control the change process and keep track of the current change status. OSCAR runs in an MVS/ISPF environment.

OSCAR-Euro works with a list of CSVs whose declarations and uses in the program statements are potentially changed. This list called a **Control Information Table (CIT)**. It is inserted into the source code of each program as a comment block. The CIT can be customized to each customer’s needs. A small example is shown in [Figure 9](#). The first few lines contain module-specific information which is created and updated automatically. In the following, we find information on 4 CSVs currently known in this module: 2 have been detected with RA-Euro and 2 with OSCAR-Euro or SCA-Euro afterwards. There is one line per CSV yielding the relevant information: Status of processing, CSV_Name, Type, Change Count (CC, incremented automatically) and Origin. The information is generated automatically where possible.

The CIT gives an immediate overview of the CSVs of each module once it is browsed or edited. The CITs of each module can be externalized, if requested, and

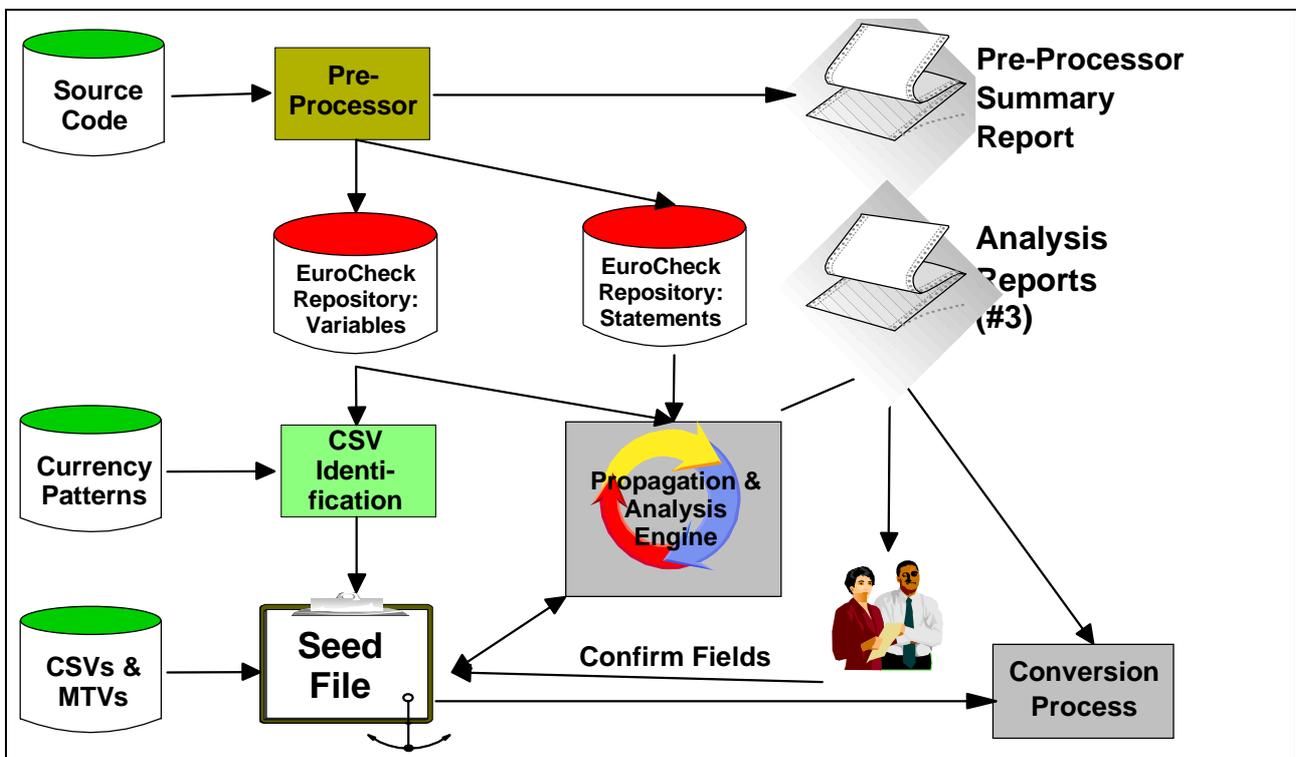


Figure 8: IBM EuroCheck

reports are available on the conversion status of all programs in a given scope.

OSCAR-Euro offers the possibility to 'smart-edit' each CSV. The user may place the cursor on a field name and press a function key to see every occurrence of that field in the module code. Like this, the relationships of the CSVs to other fields (variables and constants) are visible. There may exist a MOVE statement of the CSV to some other variable of a different type. The user may decide whether this new variable is a CSV and if so, he/she may press another function key to insert the new CSV into the CIT. This function to extend the CIT is particularly necessary where tools such as EuroCheck have not been used prior to OSCAR-Euro.

Another important feature of OSCAR-Euro is the possibility to update the code manually and record the current status in the Status column (the Change Count is updated automatically). Alternatively, for some of the euro relevant change functions, automatic code changes are feasible with the SCA-Euro tool component. In our experience the automatic update potential in euro projects is much smaller than with Y2K. However, an often used change function is the insertion of converter calls into the program code. For instance, a CSV which carries input values to some complex tariff computation may have its value converted into euro, and the subsequent calculation may be performed as in the national currency case, provided that the other variables involved have sufficient lengths, amount literals are changed as well, etc.

An overview of the OSCAR-Euro functionality is given in [Figure 10](#).

There are at least two other categories of tools which are extremely useful in the implementation phase:

- **Converter tools:** a generally applicable toolset for all kinds of amount conversions. Most important is the embedding of all EU rules on conversion and the possibility to define additional rules which may be country-specific, industry-specific or company specific, but must be applied consistently in the organization. For this functionality, we use the EuroConverter toolset [9] of our partner SOPRA. We have also built an interface from OSCAR-Euro to the EuroConverter, so that certain converter calls can automatically be inserted in the programs during migration.
- **Data migration tools:** in order to support a phased migration of data, in particular for large volumes, with minimal changes to the existing applications, IBM has provided the Euro Middleware Support [6]. This solution uses DB2 and IMS/DB product features to obtain converted amount data with minimal effort.

6. CONCLUSION

To most companies in Europe, the euro transition is more expensive than the year 2000 correction. Therefore, a comprehensive method and careful planning are required to manage the euro migration successfully over all impacted areas in the remaining timeframe.

Tools can help to attain planning reliability and can effectively support the time-consuming IT migration phases:

- analysis on organizational level, application level, and program level;

```

===== EURO CIT Start V 1.1.0 =====
Module name . . . . . : COBMBR
Status of the module . . . . . : CSVs are defined
Control information by . . . . . : CITGEN
Control information date . . . . . : 14 Feb 1998 14:12:28
Last renovated by . . . . . : RENUID
Last renovated date . . . . . : 27 Mar 1998 16:28:21
Status      CSV_Name      Type      CC      Origin
----      -
----      AMOUNT      9(x)V99   RA
----      TOTAL      9(x)V999  RA
----      COST      9(x)V9999 SCA
----      PROFIT     9(x)V99   SCA
===== EURO CIT End =====

```

Figure 9: Control Information Table (CIT) example

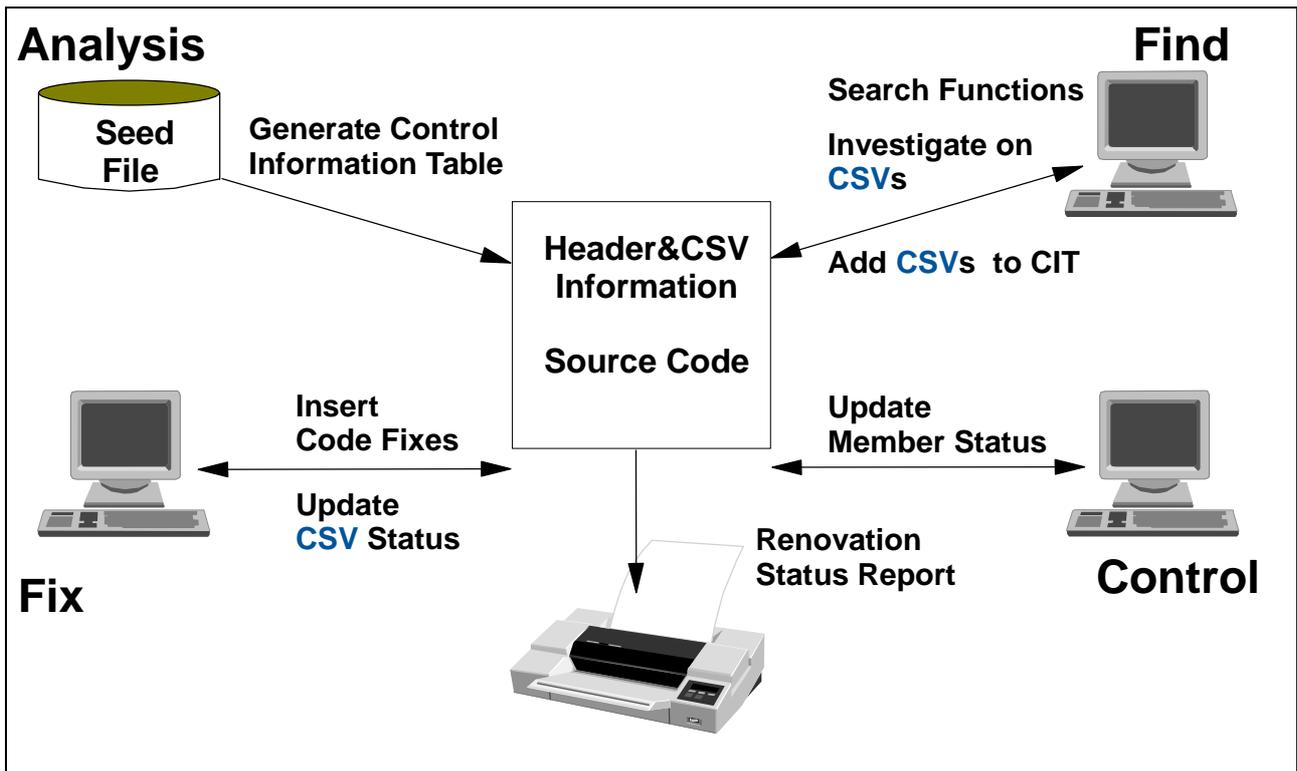


Figure 10: IBM OSCAR-Euro

- conversion of application packages, programs, and include files;
- conversion of databases, files, and jobs.

Given the variety of euro solutions and the complexity and the number of interfaces in given IT environments, euro projects can never be fully automated. This holds in particular for areas such as solution design, migration project management and partitioning strategies. Still, tools can supply the practical 'base data' for decision making.

While the tools themselves will be improved, we can improve our way of using existing tool functions today.

REFERENCES

- [1] Bank of England: Practical Issues Arising from the Euro, June 1999.
- [2] European Commission: Preparing Information Systems for the Euro, Directorate Generale XV, Internal Market and Financial Services, Brussels, September 1997.
- [3] European Council: Council Regulation (EC) No 1103/97 of 17 June 1997 on Certain Provisions Relating to the Introduction of the Euro. <http://europa.eu.int/euro/>
- [4] Gimnich, R.: Year 2000 Reengineering in Geographically Distributed Organizations. Proc. 5th. International Conference on Re-Technologies for Information Systems (RETIS '97), OCG, Vienna, December 1997.
- [5] IBM: Euro Focus - The IT impact on euro implementation projects. Form No. XSEE810, 1999. See also: <http://www.ibm.com/euro>
- [6] IBM: Euro Middleware Support. Form No. GT12-5528, 1997.
- [7] IBM: The Year 2000 and 2-Digit Dates: A Guide for Planning and Implementation. Form No. GC28-1251, 9th edition, February 1998. <http://www.software.ibm.com/year2000/>
- [8] Müller, B. and Gimnich, R.: Planning Year 2000 Transformations Using Standard Tools. Proc. 1st Euromicro Conference on Software Maintenance and Reengineering. IEEE Computer Society Press, 1997.
- [9] SOPRA: EuroConverter. <http://www.euroconverter.com/>
- [10] Weiser, M.: Program Slicing. IEEE Transactions on Software Engineering, 10(4), 1984.