



TDH022 – TECHNICAL INTEROPERABILITY GUIDELINES AND API MANAGEMENT

Operative Document Interaction Patterns

Versione: 0.2

Data: 22/02/2022

Version	Release Date	Release Type
0.1	21/12/2021	First Release – Italian
0.2	22/02/2022	Second Release – English

Contents Index

CHAPTER 1 – INTRODUCTION	4
1.1 Interaction Patterns: Preliminary information	4
CHAPTER 2 – APPLICATION SCOPE	5
2.1 Recipients of this document.....	5
CHAPTER 3 – REFERENCES AND ABBREVIATIONS.....	6
3.1 Document Reading Notes.....	6
3.2 Terms and Definitions	6
CHAPTER 4 – GENERAL PRINCIPLES	8
4.1 Blocking interactions	8
4.2 Remote Procedure Call.....	8
4.3 Non-Blocking Interactions	9
4.4 Idempotence.....	9
CHAPTER 5 – BLOCKING PATTERNS	10
5.1 [BLOCK REST] Blocking REST.....	10
5.2 [BLOCK SOAP] Blocking SOAP.....	11
CHAPTER 6 – NON-BLOCKING PATTERNS.....	12
6.1.1 [NONBLOCK_PUSH_REST] Not Blocking Push REST	12
6.1.2 [NONBLOCK_PUSH_SOAP] Not Blocking Push SOAP.....	13
6.2 Non-Blocking Patterns RPC PULL (Busy Waiting)	14
6.2.1 [NONBLOCK_PULL_REST] Not Blocking Pull REST	14
6.2.2 [NONBLOCK_PULL_SOAP] Not Blocking Pull SOAP	16
CHAPTER 7 – CRUD ACCESS TO RESOURCES.....	17
REFERENCE BIBLIOGRAPHY AND SITOGRAPHY	18

CHAPTER 1 – INTRODUCTION

This Operative Document describes the interaction patterns between a provider and a user through APIs. Providers (for this purpose we consider Public Institutions such as, for example, Regions and Provinces, as well as Public Organizations or similar and Private Entities, including Second and Third Parties that make services and functionalities available to TDH) certified within the Tourism Digital Hub must use APIs to satisfy the needs identified by the functional and non-functional requirements of specific interactions with users, which are also certified within the Tourism Digital Hub (in this sense, all subjects using the digital services made available by providers within the Ecosystem are considered).

Interaction patterns described in this Operative Document follow what shown in the Operative Document "Interaction Patterns"¹ issued by AgID and linked to the document "Guidelines on Technical Interoperability of Public Administrations"² also issued by AgID; in addition to what reported, please refer to the two documents mentioned above for certain detailed indications indicated from time to time during this document.

1.1 Interaction Patterns: Preliminary information

Interaction Patterns correlate to Message Exchange Patterns (MEP) for SOAP and REST technologies as indicated by ModI³, except for CRUD access⁴.

Given the extremely dynamic nature of the technological context of reference, the content of this document is NOT to be considered exhaustive for the purposes of defining all the Interaction Patterns present in circulation.

¹ Online Reference: https://www.agid.gov.it/sites/default/files/repository_files/01_pattern_interazione_0.pdf

² Online Reference: https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_interoperabilit_tecnica_pa.pdf

³ Italian Public Administrations Interoperability Model

⁴ Create, Read, Update & Delete – the four basic operations of persistent data management

CHAPTER 2 – APPLICATION SCOPE

This Operative Document is prepared as an operative document related to the Technical Interoperability Guideline between the TDH and its members.

2.1 Recipients of this document

This Operative Document is intended for all providers who provide users with services and functions within the Tourism Digital Hub (TDH), as well as for the users themselves, in order to enable the fruition of the desired services and functions; therefore, these provisions can be used as a basis for the implementation of new functions in case they have to be developed from scratch or as a basis for the integration of existing functions.

The following is a list of Public and Private Parties to whom the Operational Document is addressed, both as providers and users of services and functions within the Tourism Digital Hub (TDH).

Public Parties

- Central Public Administration (e.g., Ministry of Tourism),
- Local Public Administration (e.g., Regions, Provinces...),
- National and Local Authorities (e.g., ENIT),
- Non-Profit Organizations,
- Public Enterprises related to tourism (e.g., ski lifts...).

Private Parties

- Hospitality enterprises, catering enterprises, etc.,
- Tour Operators/Travel Agencies,
- Unions,
- Private Enterprises related to tourism (e.g., ski lifts...).

CHAPTER 3 – REFERENCES AND ABBREVIATIONS

3.1 Document Reading Notes

In accordance with ISO/IEC Directives, Part 3 for drafting technical documents this Operational Document will use the keywords "MUST", "MUST NOT", "SHOULD", "SHALL NOT", "MAY" and "OPTIONAL", the interpretation of which is described below:

- **MUST**, specify a mandatory requirement to comply with Guidelines;
- **MUST NOT**, indicate an absolute no-go on specifications;
- **SHOULD** or **SHOULD NOT**, mean that the implications must be understood and carefully weighed before choosing alternative approaches;
- **MAY** or **OPTIONAL**, signifies that the reader may choose to apply or not apply the specification without any kind of implication or restriction.

3.2 Terms and Definitions ⁵

For an easier reading, a glossary of terms and definitions contained in this document is given below.

[AgID]	Digital Agency for Italy
[CAD]	Legislative Decree 7 March 2005, n. 82 - "Digital Administration Code" (also known as "CAD"), updated with amendments by Legislative Decree 76 of 16 July 2020 and converted into law with Law 120 of 11 September 2020
[CRUD]	Create, Read, Update & Delete – the four basic operations of persistent data management ⁶

⁵ Some terms and definitions explained in this paragraph are also available in the Guidelines on Technical Interoperability for Public Administrations issued by AgID (see the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

⁶ Managing the Data-base Environment (Martin – 1983) Online reference: <https://archive.org/details/managingdatabase00mart/page/380/mode/2up>

[Provider]	One of the subjects referred in Article 2, paragraph 2 of the CAD that makes e-services available to other organizations, for the use of data in its possession or the integration of the processes it has carried out
[User]	Organization that uses the e-services made available by one of the subjects referred in Article 2, paragraph 2 of the CAD
[MEP]	Message Exchange Pattern
[MODI]	Interoperability Model of Italian Public Administrations
[REST]	Representational State Transfer
[RPC]	Remote Procedure Call
[SOAP]	Simple Object Access Protocol
[TDH]	Tourism Digital Hub
[TDH022]	TDH022 - Interoperability interface of the Tourism Digital Hub
[Trust]	One of the most important ways to manage security issues in the exchange of information in the network to enable interoperability between systems. It is based on mutual recognition of interacting entities and trust in each other's behavior
[UML]	Unified Modeling Language

CHAPTER 4 – GENERAL PRINCIPLES

4.1 Blocking interactions

The following communication patterns **MUST** be used in communications **between users and providers**.

A *Blocking Interaction* is a communication in which the user sends a message to the provider and waits for a response from the provider.

The Blocking Pattern can be used when the processing of the message, by the provider, does not require an excessive load, and a consistent response can be returned, completing the involvement of the provider. This Pattern is applicable when the user cannot expose a service (we will see later that it is a non-blocking Pattern), nor is the user able to actively wait.

Often this type of communication is referred to as "synchronous", or "Request/Reply" emphasizing the fact that the user receives a response to a message immediately after his request.

In this type of communication, the definition of Time Out is particularly critical.

Time Out must be properly sized during the application design phase and must be defined and shared between user and provider. An incorrectly sized Time Out could cause avoidable traffic on the network.

4.2 Remote Procedure Call

For the explanation of this concept please refer to Paragraph 4.2 of Chapter 4 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

4.3 Non-Blocking Interactions

As mentioned earlier, there are some cases in which it is preferable to have non-blocking communications. In this type of communication, it is particularly advantageous to decouple the user from the provider. The user, therefore, once the message has been sent, expects, at first, only a take-over by the provider and may optionally return a response message. Such approach is particularly indicated when a choreography of microservices must be adopted and allows a smaller computational load from the two actors in game, also being able to demand a greater cost in terms of times of elaboration. This type of communication is often referred to as "Asynchronous".

4.4 Idempotence

For a general explanation of this concept, please refer to Paragraph 4.4 of Chapter 4 of the Operative Document "Patterns of Interaction" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (*refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents*).

For the http protocol there are certain methods that enjoy the principle of idempotence. As an example, the PUT method, contrarily to the POST, has the peculiarity to be able to create (or to update for entirety) a resource, and also to forehead of a second equal call, a new resource does not come created (for the POST method, instead, a new resource comes created).

CHAPTER 5 – BLOCKING PATTERNS

For the general principles related to the development of RPC-Like blocking interfaces, please refer to Chapter 5 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (*please refer to the section "Reference Bibliography and Sitography" for the redirect links to the mentioned contents*).

A synthetic representation of the process of interaction between user and provider, with contextual response from the latter to the user's request, is reported for convenience in the exposition of the following paragraphs.

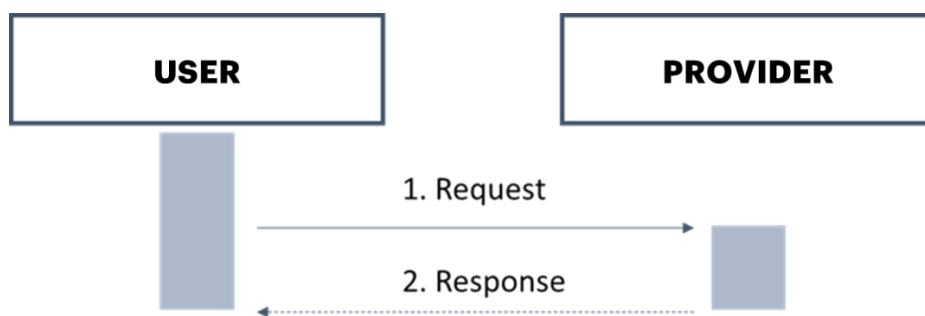


Figure 1 – Synthetic representation of RPC Blocking Interaction

5.1 [BLOCK REST] Blocking REST

If implementing via REST technology, at least the following instructions **MUST** be followed:

- The interface specification **MUST** declare all HTTP status codes provided by the interface, i.e., all those codes that it is possible to return when, from dispatchers, the request reaches the server, unless rules and policies (e.g., rate limit), with the relative response pattern, as well as eventual returned HTTP headers;
- The interface specification **MUST** state the request pattern along with any required HTTP headers;

- In step (1) of Figure 1, the user is depicted that **MUST** use the HTTP method POST verb on a URL containing the affected IDs and the method name as the HTTP verb for the execution of the procedure call;
- In step (2) of Figure 1, the dispenser is depicted that **MUST** use HTTP status 2xx unless errors occur.

For the detailed representation of the processing rules and the consultation of practical examples related to this Pattern, please refer to Paragraphs 5.1.1 and 5.1.2 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the mentioned contents).

5.2 [BLOCK SOAP] Blocking SOAP

If the pattern is implemented with SOAP technology, unlike the REST case, the invoked method is not specified in the called endpoint, since it is identified within the body. In addition, all IDs involved **MUST** be reported within the body. The following rule **MUST** be followed:

- the dispenser interface specification **MUST** declare all exposed methods with associated request and return message patterns and **SHOULD** have the Fault pattern. In addition, the interfaces **MUST** specify any required SOAP headers.

For the detailed representation of the processing rules and the consultation of practical examples related to this Pattern, please refer to Paragraphs 5.1.1 and 5.1.2 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the mentioned contents).

CHAPTER 6 – NON-BLOCKING PATTERNS

For general concepts related to RPC-Like non-blocking patterns, please refer to Chapter 6 and Paragraph 6.1 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the mentioned contents).

It is brought back, for convenience in the exposure of the successive paragraphs, one synthetic representation of the not blocking process of interaction through callback⁷:

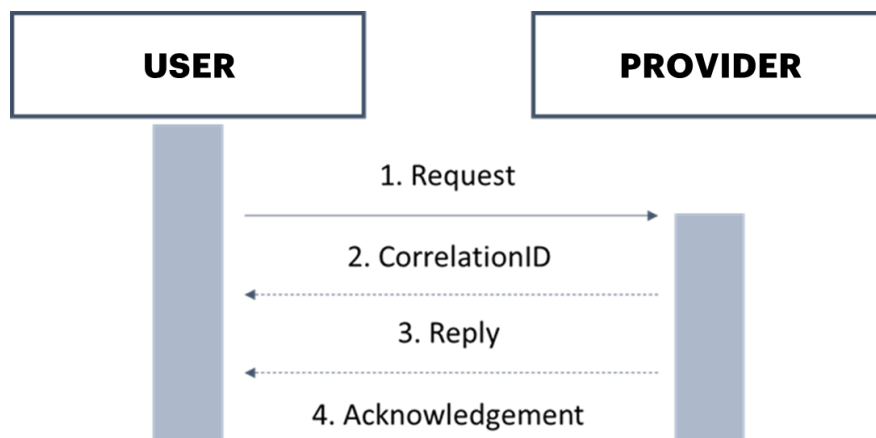


Figure 2 – Simplistic representation of non-blocking interaction by callback

6.1.1 [NONBLOCK_PUSH_REST] Not Blocking Push REST

If the pattern is implemented with REST technology, the following guidelines MUST be followed:

- The specifications of the user and provider interfaces MUST declare all HTTP status codes provided by the interface, i.e., all those codes that it is possible to return when, as a provider, the request reaches the server, unless the rules and policies (e.g., rate limit), with the relative response pattern, as well as any HTTP headers returned;
- The specifications of the user and provider interfaces MUST declare the patterns of the requests along with any required HTTP headers;

⁷ A functional or "block of code" transmitted to another functional as a parameter

- The specification of the provider interface must declare through the specific formalism the format of the callbacks; this specification must be respected by the interface exposed by the user, and therefore in the respective specification;
- At step (1) in Figure 2, the user is shown to MUST indicate the endpoint of the callback using the custom X-ReplyTo HTTP header and using HTTP method POST;
- At step (2) of Figure 2 instead, the provider is shown that MUST provide along with the acknowledgement of the request in the body, the CorrelationID using the custom HTTP header X-Correlation-ID; The HTTP status code MUST be HTTP status 202 Accepted unless errors occur;
- At step (3) in Figure 2, the focus returns to the provider who MUST use the same CorrelationID provided in step (2) again using the custom X-Correlation-ID HTTP header; The HTTP verb used must be POST;
- At step (4) of Figure 2, finally, the user is shown that he MUST acknowledge receipt of the response via an acknowledgement message; the HTTP status code MUST be HTTP status 200 OK unless errors occur.

For the detailed representation of the processing rules and the consultation of practical examples related to this Pattern, please refer to Paragraphs 6.1.1.1 and 6.1.1.2 of the Operative Document "Interaction Pattern" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

6.1.2 [NONBLOCK_PUSH_SOAP] Not Blocking Push SOAP

In case of SOAP implementation, the callback endpoint and the CorrelationID are included in the SOAP header as custom fields. Provider and user MUST also follow the following rules:

- The specification of the user and provider interfaces MUST declare all exposed methods with associated request and return message patterns and SHOULD declare the FAULT pattern. In addition, the interfaces MUST specify any required SOAP headers;
- The specification of the user interface MUST respect what is required by the provider; in particular, it is required that the provider provides a WSDL describing the callback service that the user is required to implement;

- At step (1) in Figure 2, the user MUST indicate the endpoint of the callback using the custom X-ReplyTo SOAP header;
- At step (2) in Figure 2, the provider MUST provide along with the acknowledgement of the request in the body, the CorrelationID using the custom SOAP header X-Correlation-ID;
- At step (3) in Figure 2, the provider MUST use the same CorrelationID provided in step (2) again using the custom SOAP header X-Correlation-ID;
- Finally, at step (4) in Figure 2, the user MUST acknowledge receipt of the response via an acknowledgement message.

For the detailed representation of the processing rules and the consultation of practical examples related to this Pattern, please refer to Paragraphs 6.1.1.1 and 6.1.1.2 of the Operative Document "Interaction Pattern" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

6.2 Non-Blocking Patterns RPC PULL (Busy Waiting)

For the general concepts related to this specific pattern and the differences with the Pattern RPC PUSH, please refer to Paragraph 6.2 of the Operational Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the mentioned contents).

6.2.1 [NONBLOCK_PULL_REST] Not Blocking Pull REST

If a profile is implemented using REST technology, the following rules MUST be followed:

- The provider interface specification MUST declare all returned HTTP status codes with associated response schema, minus rules and policies (e.g., rate limit), along with any returned HTTP headers;

- The interface specification MUST declare the request schemas along with any required HTTP headers;
- At step (1) of Figure 3 below, the user MUST use the HTTP POST verb (or alternatively PUT);
- At step (2) of Figure 3, the provider MUST provide along with the request acknowledgement, a path to query the processing status using HTTP header Location, the HTTP status code MUST be HTTP status 202 Accepted unless errors occur;
- At step (3) of Figure 3, the user MUST use the path in step (2) to query the status of the resource; the HTTP verb used MUST be GET;
- At step (4) of Figure 3, the provider indicates, based on the processing status, that the resource is not yet ready (the HTTP code returned is HTTP status 200 OK) or indicates that the resource is ready, using HTTP header Location, to indicate the path where to retrieve the resource (the HTTP code returned is HTTP status 303 See Other);
- At step (5) of Figure 3, the user MUST use the path in step (4) in case the resource is ready to request the resource, the HTTP verb used must be GET;
- At step (6) of Figure 3, the provider responds with a representation of the resource, the HTTP code returned is HTTP status 200 OK.

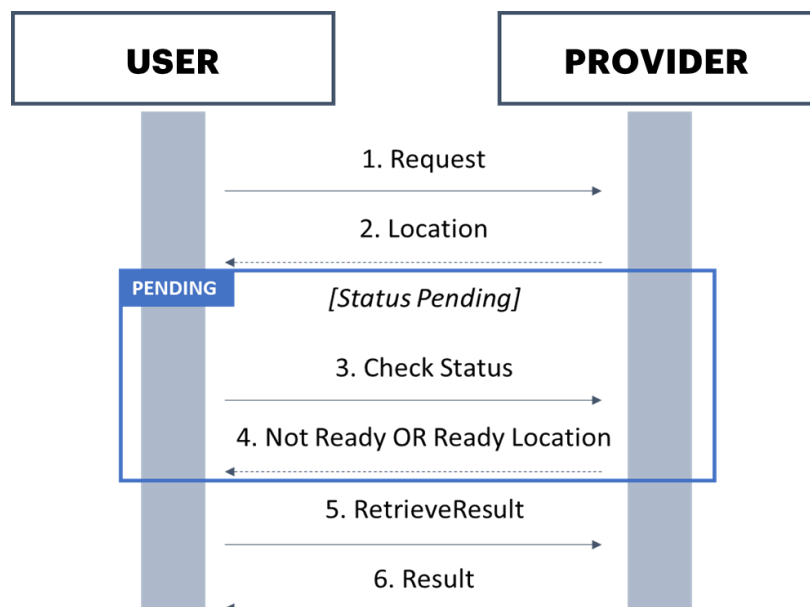


Figure 3 – Simplified representation of non-blocking interaction via busy waiting REST

For the detailed representation of the processing rules and the consultation of practical examples related to this Pattern, please refer to Paragraphs 6.1.1.1 and 6.1.1.2 of the Operative Document "Interaction Pattern" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents).

6.2.2 [NONBLOCK_PULL_SOAP] Not Blocking Pull SOAP

For the general concepts related to this specific pattern, including the details of the processing rules and the consultation of practical examples in support, please refer to Paragraphs 6.2.2, 6.2.2.1 and 6.2.2.2 of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Reference Bibliography and Sitography" for the redirect links to the cited contents). It is brought back however, to explanatory title, simplified representation of interaction (Figure 4):

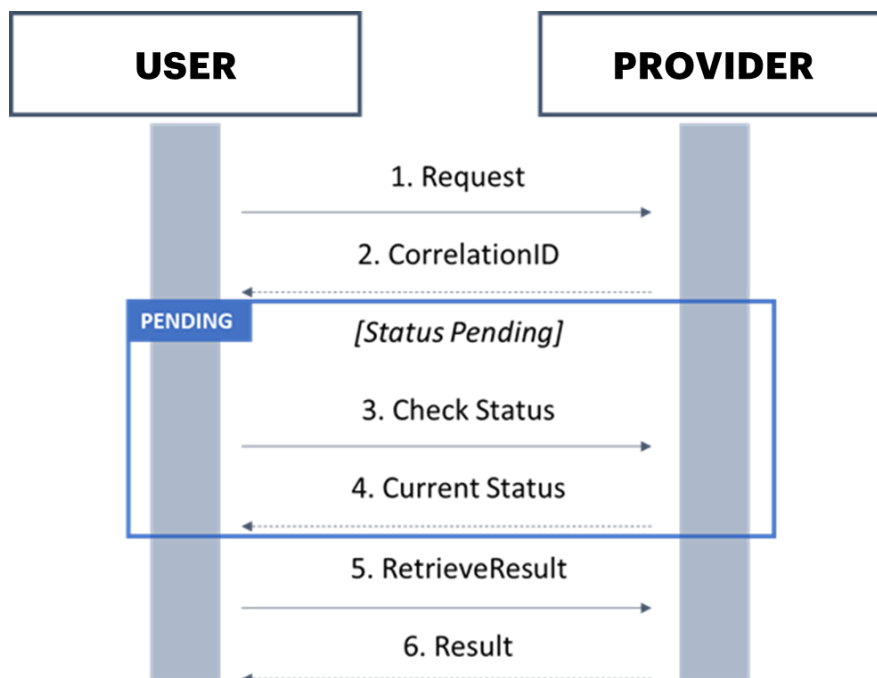


Figure 4 – Simplified Representation of Non-Blocking Interaction via Busy Waiting SOAP

CHAPTER 7 – CRUD ACCESS TO RESOURCES

For a detailed explanation of the CRUD (Create, Read, Update, Delete) operations performed on the resources of the domain of interest, please refer to Chapter 7 (and related paragraphs for processing rules and practical examples) of the Operative Document "Interaction Patterns" published by AgID, linked to the Document "Guidelines on Technical Interoperability of Public Administrations", also published by AgID (please refer to the section "Bibliography and Reference Sitography" for redirect links to the cited contents).

REFERENCE BIBLIOGRAPHY AND SITOGRAPHY

Guidelines on Technical Interoperability of Public Administrations

Author: AgID – First Release: 27/04/2021

Online reference: https://www.agid.gov.it/sites/default/files/repository_files/linee_guida_interoperabilit_tecnica_pa.pdf

Operative Document – Interaction Patterns

Author: AgID – First Release: 27/04/2021

Online reference: https://www.agid.gov.it/sites/default/files/repository_files/01_pattern_interazione_0.pdf

Managing the Data-base Environment

Author: Martin – First Release: 1983

Online reference: <https://archive.org/details/managingdatabase00mart/page/380/mode/2up>