

DOI: [10.52950/SS.2021.10.2.003](https://doi.org/10.52950/SS.2021.10.2.003)

A GENERIC PROCESS MODEL FOR SOCIOCULTURAL DATA COLLECTION AND DECISION MAKING IN FACILITY PROGRAMMING

LUBOMIR POPOV, FRANKLIN GOZA

Abstract:

This is the third paper in a trilogy of papers that started with developing the philosophical and theoretical foundations for the study of sociospatial interactions. In the second paper, we created a guiding framework for facilities programming information collection and decision making. In this paper, we use that guiding framework and develop a process model for the programming process. Our methodology involves systems theory, activity theory, and task analysis. We present a generic model of the data collection process in facilities programming. This model delineates process steps, their sequence, and their content. The intent of this model is to provide structure and content suggestions for developing project-specific facilities programming research designs for information collection, processing, and programmatic decision-making. It is a plan for action, articulated and explicated so that professional programmers can use it as a template for organizing their work. This process model is tentative and general, intended as a guide for creating customized process designs for each programming project, considering the specifics of the building type and the programming situation.

Keywords:

facility programming, theory of facility programming, methodology of programming research, process model for programming

JEL Classification: R00, L74, L29

Authors:

LUBOMIR POPOV, Bowling Green State University, United States, Email: Lspopov@bgsu.edu
FRANKLIN GOZA, University of Wisconsin-Whitewater, United States, Email: gozaf@uww.edu

Citation:

LUBOMIR POPOV, FRANKLIN GOZA (2021). A Generic Process Model for Sociocultural Data Collection and Decision Making in Facility Programming. *International Journal of Social Sciences*, Vol. X(2), pp. 31-56., [10.52950/SS.2021.10.2.003](https://doi.org/10.52950/SS.2021.10.2.003)

1 Introduction

This paper presents a generic model of the data collection process in facilities programming. The model delineates process steps, their sequence, and their content. The intent of this model is to provide structure and content suggestions for developing project-specific facilities programming plans for information collection, processing, and programmatic decision-making. This is the third part of our project focusing on facility programming methodologies and procedures. Our initial study *Philosophical Foundations and Metatheoretical Considerations for Creating Frameworks to Collect Facility Programming Information* (Popov & Goza, 2018) presented the philosophical foundations for the creation of a guiding framework. That study informed the next phase, published as *A Framework for Data Collection in Facility Programming: The Subject-Object Approach* (Popov & Goza, 2019). In that paper, we developed a framework for collecting programmatic information, but stopped short of a process model for information collection. In this paper, we do this. In this respect, the current process model will inform not only programmers and programming researchers but will also illustrate the operationalization of the framework for data collection to the level of a generic example.

In this line of development, we rely strongly on our presentations in the preceding papers. There, reader can find more detailed information about facilities programming, our methodological thinking, and the major concepts and terms. As mentioned before in Authors (2018), facility programming is about examining building users and stakeholders for developing design requirements and generating the information required in order to make design decisions (Cherry, 1999; Hershberger, 2015; Lang & Moleski, 2010; Preiser, 1993). Such programming involves a significant amount of inquiry which must be conducted with sound research methods (Popov & Goza, 2018). Although there are different paradigmatic views about the role and the process of preparing research designs, it is our belief that research requires thorough planning and that a well-designed examination should guide both data collection and analysis (Popov & Goza, 2018).

Our prior publication reviewed 40 years of relevant theoretical resources in the field of facility programming (Popov & Goza, 2018). That review revealed only a limited number of studies with frameworks designed with the purpose of collecting programming information, as well as suggestions for organizing the process of data collection. We also reviewed social science literature in several fields in search of frameworks that could be adapted to serve scholars working on programming research (Popov & Goza, 2018). Although some earlier publications provided interesting insights, not a single study was found that proposes a detailed and well-articulated process model for data collection and programmatic decision making. And although it is possible

to articulate the programming process in some of the published case studies, that work falls short of a purposfully created and theoretically substantiated process model (Popov & Goza, 2018).

Our conceptual framework for guiding the data collection process was conceived as a wide-ranging and flexible guide for the development of project-specific research instruments. Here, we interpret framework to mean a way to organize information about a social phenomenon and as a way to systematically present a class or type of phenomena for the purpose of guiding research on the specific phenomenon examined (Jabareen, 2009; Popov & Goza, 2018). Our framework (Table 1) is a theoretical arrangement for organizing relevant components and interrelationships. It will guide us in the construction of a sequence of research and decision-making tasks that will comprise a generic programming process model.

There are several reasons to offer a generic process model for facilities programming data collection and decision-making. There are dozens of building types and even more subtypes. Each one of these types is different not just architecturally, but regarding the social organization it is supposed to accommodate, the industry, and the local societal concerns. And when we program a specific building, the diversity in settings, situations, factors, and influences is so big that each project becomes a unique universe of its own. Any specific process model will have only a limited application would not be of a substantial use in other situations. The idea to propose a generic model is based on our belief that the methodological guidance in this domain should be broad enough to allow for adaptation to any specific situation, yet it should be heuristic enough to provide productive guidance. These beliefs influenced our goal: based on the philosophical principles and the content framework, to develop one possible version of the many alternatives that could emerge as a result of using these general guidelines to collect data.

2 Methodological Considerations

In this part of our project, we construct or design an abstract process model for organizing the programming effort. This constructive act can be compared to a research design using the language of the discipline of research methodology. It is a plan for action, articulated and explicated so that professional programmers can use it as a template for organizing their work. The process model presented in this paper is created on the basis on the philosophical assumptions we have previously developed (Popov & Goza, 2018) and the conceptual framework we show below (Popov & Goza, 2019). In addition, we continue using Systems Theory (Bausch, 2001; Bailey, 1994; 2001; Bausch,

2001) and Activity Theory (Bedny & Karwowski, 2007; Kaptelinin, Kuutti & Bannon, 1995) approaches to develop the formal structure of the process model. In a more concrete parlance, we juxtapose two frameworks and create the template or the guiding scaffolding of the programming process. These two frameworks are a knowledge content framework (Popov & Goza, 2019) and a formal process structure framework that is based on the notion of task (Bedny & Harris, 2008; Turner & McEwan, 2003).

We explicate and articulate the process in task clusters, tasks, and subtasks. This kind of explanation is intended to provide better communication with both researchers and practitioners, but it is particularly useful for practitioners who need a more pragmatic presentation of working ideas. We develop the task articulation process model by applying an activity perspective to job, work, and task analysis (Bedny & Harris, 2008; Engeström, 2000; Turner and McEwan, 2003).

The task articulation detail will vary depending on our perception of the complexity of the sociospatial situations, the importance of these situations for the building design, and the available time and resources during the programming process. The articulation and subdivision into several levels also depends on the views of the researchers and programmers. Some programmers might prefer to organize the process in several large parts or phases, and then articulate task clusters, tasks, and subtasks. This makes a tree-like structure with an explicit hierarchy of levels. Other programmers might prefer to make a long chain of consecutive (at least nominally) tasks without segmenting the process into sections/phases.

The framework (Author, 2019) we use here is a generic one, designed to be used in facilities programming for many building types, including work environments, civic buildings, and leisure/recreation facilities. We have tried to select the most common features pertinent to programming these building types. The framework is influenced by the design of formal organizations (Burton, et al., 2006; Cummings & Worley, 2008; Holt & Morris, 1993) because the literature in that area provides well defined and explicated solutions.

The horizontal axis of the framework is based on an activity model that consists of agents, their goals, activities, necessary conditions, and the built environment that provides these conditions. The vertical axis brings in three levels or scales for analyzing the social realm (from organization to groups to individuals). The analytical work at each level starts with the Agent and proceeds toward the Built Environment category. However, the process is recursive, and we have considered the necessity for feedback and for cycling through the research process several times until programmers are able to collect and align all required information. This approach allows for a better

explication of interactions and relationships within the sociospatial system. Furthermore, starting the analysis of social reality at the higher levels helps understand the lower levels (Banathy, 1996).

In addition to the guiding framework (Table 1.), we use our vast knowledge and experience in social science research methods and facilities programming to introduce modifications and adjustments as needed. Theoretical or conceptual models are generalizations that must be applicable to a multitude of situations. They are “one size” creations but cannot “fit all,” to use the popular adage. These models always need to be adapted and fine-tuned. In this case, knowledge about the reality of facilities programming helps us interpret each component of the framework as a guide, but also tells us how to use this guide in programming research situations. The logic of theorizing and the logic of practice might differ depending on situational obstacles, lack of resources (time and budget), and insufficient theoretical knowledge. In practice, there are shortcuts, timesavers, and segmentation and integration of tasks as needed.

THE SOCIOSPATIAL REALITY	>>>>> AGENTS: Users & stakeholders (STKH) Levels: Individuals, groups, or organizations.	>>>>> GOALS: Link agents to activities by presupposing types of actions.	>>>>> ACTIVITIES: Major linking mechanism. Relate agents to environment.	>>>>> NECESSARY CONDITIONS: Relate activities and agents to environment.	BUILT ENVIRONMENT: Provides conditions, instrumental features, resources.
LEVEL: Organization (and its social environment)	> research loop Social environment, market, competition. The organization as a whole.	>>>>> Vision, mission, goals, strategy, and policies.	>>>>> Organizational processes, operations, and activity systems.	>>>>> Necessary conditions for activity systems and flows of materials, energy, and information.	Design and planning requirements/ Spatial organization of the whole building.
	< feedback loop STKH action for feedback and negotiation.	<<<<< Conflict with STKH functions and goals	<<<<< Undesirable effects on STKH activities	<<<<< Undesirable effects and consequences for STKH.	< feedback loop Spatial effects and consequences for STKH.
LEVEL: Group/Team/ Department	> research loop Departments and teams/groups. Populations of behavior settings.	>>>>> Department, group, or unit goals: a) related to the organization, b) for sustaining the unit.	>>>>> Operations and activity systems, behavior patterns, patterns of social interaction.	>>>>> Necessary conditions for teamwork, social interaction and social needs of individuals.	Design requirements of the group/team.
	< feedback loop STKH action for feedback and negotiation.	<<<<< Conflict with SKH functions and goals	<<<<< Undesirable effects on STKH activities	<<<<< Undesirable effects on STKH, creating new problems.	< feedback loop Spatial effects and consequences for stakeholders.

LEVEL: Individual	>research loop Organism, psyche, social status; values, culture.	>>>>> Goals and objectives: life goals, professional goals, work objectives.	>>>>> Activity and behavior patterns, personal experience.	>>>>> Necessary conditions for the individuals in their activities.	Design requirements. Spatial organization of environment for the individual.
	<feedback loop STKH action for feedback and negotiation.	<<<<<< Conflict with STKH functions and goals	<<<<<< Undesirable effects on STKH activities	<<<<<< Undesirable effects on STKH.	< feedback loop Spatial effects & consequences for stakeholders.

Table 1. The general framework for collecting programming information (Popov & Goza, 2019).

In the guiding framework, we have prominently displayed the feedback loops and tasks. The assumption is that each time programmers perform a task, they will evaluate the products and collect more information or fine-tune as needed. However, in practice, this doesn't always happen, and it is not always necessary. So, in the presentation of our process model we will very often skip these feedback tasks to communicate the process model more clearly and to save presentation space.

The purpose of the framework is to guide programmers in the process of data collection and programmatic decision making. The theoretical organization in layers and components makes the analytical operations more visible and logically related. However, when working in practice, programmers are under pressure to simultaneously collect information at several levels and for several components. In this framework, the phenomena and the information about them at organizational level are better differentiated than the phenomena at departmental and individual levels. When collecting information and making decisions regarding those levels, very often programmers must consider what is happening there in a holistic way. Here, as needed, we will introduce corrections to the theoretically constructed process. These corrections come from the reality of programming practice, but also from the reality of the social world, which is not segmented in disciplinary and theoretical silos. In the real world, when we design new activities and service operations, we use and synthesize all our knowledge about the intended product.

3. Introduction to the Process Model

This process model is intended to work as a template for organizing research and decision making for specific programming projects. We intentionally do not cover the logistical, administrative, or other support activities that are needed to do programmatic research and decision making. There are many tasks of this kind. They can be seen as a parallel structure, although they are very tightly

interwoven with the content tasks. In effect, the performance of each content task requires several other logistical tasks to support it. For this project, we consider these tasks separately because they involve other types of skills and knowledge. For example, organizing meetings and workshops; defining assignments for the client participants and helping the client agents with these assignments; travel, time schedule, and budget arrangements; and many more.

As mentioned already, the variety of programming projects is huge. Just the different client management culture alone might substantially change the programming process, although we have the same building type, the same building size, and the same kind of social environment. It is not possible to offer a cookie-cutter model. So, if readers cannot find something that they do in practice, or they see something they don't do in practice, they should not be disappointed. They might consider how to insert their needed component or on the other end, how to drop a task. This is particularly true about the feedback loop and the merging of tasks or the simultaneous work on several tasks. Working simultaneously on several tasks saves a lot of time and money, both to the programming firm and the client, whose workers are involved in the programming process and spend valuable work time on it. The cyclical or spiral movement in this process is so complex and unpredictable that it cannot be "photographed" and made still in a table or process description. We also need to emphasize the linear nature of verbal descriptions that do not allow showing the complex spiraling, overlapping, and merging of tasks in practice. This is a common problem describing any social or intellectual service process because of the nature of social phenomena and entities.

Another important consideration for interpreting our model is the fact that some clients have, and others don't, documented information about some of the tasks. In some cases, the clients have developed this information themselves or have hired management consultants prior to the initiation of the facility project. In such cases, programmers still have to consider the tasks we have mentioned. Even if the information for these tasks is already documented and available, the programmers need to read these documents and discuss them with the client. This is still a task that takes time and resources and needs to be planned for.

In other cases, the clients have contracted management consultants in relation to the programming process and the programmers need to work with those consultants. In such cases, the management consultants might take ownership of the process, at least for the organizational design. They might have a different process template and might work very differently on organizational design issues. They might use a SWOT model of organizational design, which brings a different process template.

Still in other cases, clients don't realize they need management consulting help and expect that programmers will do everything. The programming firm might do that if they feel they have enough expertise or might subcontract a management consulting firm. The worst-case scenario is when neither the clients or the programmers realize that they need management consulting expertise to do what effectively is organizational design. However, our process model might enlighten such parties and inform them about what they need to do and what expertise they need to get it done.

The knowledge content framework (Popov & Goza, 2019) and a formal process structure framework (the task structure) that organize the programming process (Bedny & Harris, 2008; Turner & Mcewan, 2003) are informing the decisions about conceptualizing process tasks. The content framework has a logic derived from the realms of studying and managing social and sociospatial interactions. We have followed that logic to procure consistency and trace the relationships. However, as we mentioned before, we superimposed on this framework the logic of the programming process with its inevitable considerations for feasibility of field research, the organizational peculiarities of the paying client, and the nature of the building users. The interaction among these realms and considerations provided a more realistic research design that is not always fully complied with in each and every cell of the framework. We want to prepare the readers for interpretation of the proposed programming data collection process due to flexible, feasible, and at times, even re-inventing process of application of the framework. This means that the guiding framework should not be perceived as a rigid structure that should be followed literally but should be interpreted as a navigating instrument that allows for considering many eventualities.

In order to fit into the typical journal paper space, we have decided to focus predominantly on the content aspect of the programming tasks. We will rarely talk about research methods used in the programming process and will do that only if it is very important for getting a good perspective on that task. The research methodology can be a complimentary aspect and an issue of its own. It can be explored in a separate paper. In principle, there are many publications of research methods, while there is almost no literature highlighting the content and placement of programming tasks.

4. Project Phase: Initiation and Planning of the Process

This phase mostly involves the preparation for launching the programming project. It is about planning the process, creating management structures, and setting up the logistics. This phase doesn't evolve from the framework for collecting programming information. Rather, it follows the logic of management and logistics of the project. For these reasons, we will not go into detail.

However, we will mention some of the work done during this phase so that readers can imbed the content tasks into the whole project process.

There are several task clusters and tasks at this phase. It can start with developing communication channels between the client leaders and the programmers. It will also include a more detailed plan for the project, which is substantially more detailed than the submitted documents at the call for proposals. Still, it is early to develop a full-scale process chart with timelines and budgets.

At this phase, the main parties in the project will develop project management structures and entities; will schedule several meetings for clarifying the project scope: for exploring existing information about the client; what information is readily available and what needs to be collected; what organizations design decisions are done and is there are a need to update these decisions. The programmers will negotiate the logistics of the project. The logistics include the creation of client's staff project committees/ad hoc teams and recruitment of volunteers; procuring workspace for the programming team and the client ad hoc teams; management protocols, communication protocols, and so forth.

The existing information review is actually content-related and will be guided by a framework for organizational design information collection. These types of tasks can benefit from the information collection conceptual framework that we propose. However, for brevity, we will not explicate those steps at this point, because they vary greatly depending on client culture, existing organizational design documents, existing information, and so forth. After reviewing our presentation in this paper, the reader can explore the role of our content framework for guiding the task clusters at this phase. The actual application will be similar to the work done during the phases guided by the information collection framework, which are explained below in this paper.

5. Project Phase: Strategic and Organizational Planning.

The information collection process and the programmatic decisions described below can be categorized as "strategic." They form a hierarchy that is structured in several layers. Each upper layer presupposes to a large extent the decisions at the lower layers. Some decisions may have less influence than others or affect only specific aspects of the programming process. If a decision at a higher level is not correct, that will reverberate throughout the whole project. The decisions are interconnected in a semi-lattice structure, and an incorrect decision at the higher level might create quite of a confusion at some point down the process. An incorrect decision in the upper layers might

suggest unrealistic directions for exploration and consideration. It will also conflict with some later decisions reached based on empirical information and sound logic.

The tasks (and their content) presented here are articulated following the guiding logic of the knowledge content framework shown above, as well as our experience in facilities programming. The suggestions stemming from the framework can be interpreted in different ways. The logic of facilities programming practices provides a second line of directions. The topics and information search sequence discussed in the framework need to be interpreted creatively, with an understanding that the programming process must be adapted to real life situations. Also, considering that there are various models of organizational design, the process of collecting programmatic information and making decisions can be reshaped in various ways. In this paper, we present a generic operationalization of the knowledge content framework into a procedural guide with tasks, content, and sequence that are contingent on various situations and various interpretations of the framework. We also want to highlight that readers should not expect a verbatim operationalization of the framework. The value of the framework is to provide direction. The formulation of the tasks is informed both by the components of the framework and by programming experience.

Task 1. Initial analysis of organizational mission, goals, and strategies with the purpose to plan for the study of organizational environment. The study of the organizational environment can become an overwhelming project. For that purpose, programmers need to discuss and analyze the readiness and applicability of current strategic plans of the organization together with the client. If the plans are not current (which is pretty common), programmers need to suggest an update. The update can be done by the clients themselves, by an organizational design contractor, or by the programming firm if it has such expertise on board.

Task 2. Analyzing the functional environment of the organization. The availability, adequacy, and relevance of information regarding the strategic issues of the organization will define the scope of researching the organizational environment. This might become a very broad study that requires a lot of resources. To manage workload, programmers might filter the direction of data collection by considering only what they need for constructing organizational mission, goals, strategies, service mix, and so forth.

Task 3. Identifying stakeholders. We have separated this task from the analysis of functional environment because of its importance for ensuring that all parties that are interested, concerned, or affected by the new facility will have opportunities to express their needs, requirements,

concerns, and so forth. The list on stakeholders can be extremely long if we dig deep into the facility's functional environment. The complexity of this task is the making of the right choices. The number of stakeholders should be manageable, while every interested party is still given an opportunity to express their considerations. Furthermore, the stakeholders can be categorized in layers/categories regarding their social and functional "proximity" to the facility, the intensity of facility impact on them, and their relative position in the universe of all stakeholders in this project.

Task 4. Research and a general vision for the market niche, services, and facility capacity.

After the definition of the business domain, the goals and objectives, and the corporate philosophy, a second series of tasks is initiated. First, the basic characteristics of the business situation are analyzed, and a market segment is selected. This decision presupposes to a large extent the next series of choices. On that basis, the market catchment area of the facility is defined. The definition of the catchment area is an important prerequisite for estimating the future service load (workload) and then the capacity of the building. Considering the specifics of the catchment area, a rough profile of the whole customer population is made, and distinctive customer groups are identified. These will constitute the service population of the organization and will define the study of the potential customers regarding their service and service-related needs.

This will presuppose a study of its own, with a scope defined by the previous tasks. First, researchers need to develop sociocultural profiles of the target customer groups. Researchers are guided by the information needs for strategic decision making in this industry and this facility type. They use this information to develop service programs and envision in general how to organize the operations. This is preparation for a more extensive work on organizational operations later in the programming process. After that, researchers will engage in defining the general and specific needs of the target customer groups to get amore detailed visioning of service programs.

The activities performed in this task, or cluster of tasks, are far beyond the competencies of most programmers. Such services are usually provided by marketing and strategic planning firms. It is a good idea to subcontract high quality marketing consultants to study the market and to profile the situation. However, some facilities planning firms have enough expertise to facilitate this process and provide continuity for the process already started.

Task 5. Defining business environment and competition. The purpose is to understand the main factors and future developments regarding opportunities, constraints, and threats for the particular industry and business, and furthermore the client organization. This produced the foundations for formulating a strategy for accomplishing the corporate goals (Rothaermel, 2017). The objective is to correctly project the mix of services and programs to be offered. The environmental forecast has implications for the additional

elaboration of the mission and goals of the organization. The study of business environment also looks at the basic challenges the firm would have to meet. This involves researching the competition as a major factor in fine-tuning the mix of products and services. The strategic behavior of a corporation is influenced both by the business environment, and in turn the future business developments at a larger scale (Vorlicek & Cermakova, 2017).

Task 6. Culture of the social environment, the organization, and service population. Culture is an important factor and consideration because it defines the goal system, behavior systems, and operations of the organization and its customers or patrons. At this stage of programming, it is important to understand the culture of all levels of functioning: from the environment to the single agent or customer. The culture of the social functional environment and the serviced population influences client, customer, and patron interaction with service programs and staff (Cameron & Quinn, 2011; Šikýř, Gorokhova & Šafránková, 2019). In the process of redesigning a service-driven organization, it is important to consider how the culture of the social functional environment will interact with the organizational culture and operational patterns. Meanwhile, parallel sub-task engages with studying the culture of the service populations/customer groups. All these investigations will influence the development of organizational mission, goals, service mix, and several other steps in the process of updating organizational structures. The study of culture will continue as a parallel sub-task in the subsequent steps of the project. In this case, we articulate the study of culture to emphasize its importance.

Task 7. Defining the goal structure of the organization. This task (or task cluster) involves major work on the vision, mission, goals, and objectives of the client organizations (Burke, 2014; Burton et al., 2006; Cummings & Worley, 2008). Organizations should have such documents “on file” and should update them frequently. However, this is not what happens in “real life.” It is often necessary to update or even create new documents. Such decisions are developed in several cycles and might be changed during the programming process depending on necessity to fit organizational goals, structure, and facility resources (Cummings & Worley, 2008; Farbstein, 1986: 81). However, most of the work of this task is performed very early in the programming process. The process usually starts with visioning, then proceeds towards defining the mission, organizational values, and so forth. The functional relationship between vision and mission is very complex and authors often have different and conflicting positions about the order of interaction and production. The controversies can be resolved by engaging in a cyclical process of reinvention and adaptation of the major strategic considerations of the organizations. These sub-tasks make a foundation for developing organizational goals and objects.

The mission statement provides a general and concise overview of what the organization (facility) is supposed to do. It becomes the basis for formulating the goals. They serve as more specific guidelines for developing the goal system of the organization. Goals formally spell out what the organization is striving for and provide directions for developing the strategy. A goal is a brief statement that defines in general terms the result to be achieved in the operation of the organization. It relates to a specific aspect of operations and is therefore narrower in focus. But also, like the mission statement, it reflects an ideal toward which the facility should be striving (Burke, 2014; Burton et al., 2006; Cummings & Worley, 2008; Farbstein, 1986).

Goals are specified in terms of objectives. The objective is an operationalization a goal. While the goal describes an end result, the objective describes an activity or a system of activities that helps achieve the result specified (Burke, 2014; Burton et al., 2006; Farbstein, 1986). The goal, is more or less, an ideal image of a desired result, while an objective is its empirical referent, as far as this is possible. The objective is formulated in more precise terms, has almost univocal interpretation, and is measurable.

Task 8. Strategy. On the grounds of the information and decisions previously made, the strategy of the client firm begins to gradually emerge. Strategy is an operationalization of the client mission, a more specific definition of client business, overall philosophy, and goals (Cummings & Worley, 2008; Rothaermel, 2017). Strategy is often seen as a philosophy by itself: a guiding philosophy of the organization in commitment of its resources to attain or fulfill its goals. Strategy can also be defined as a guide to activities, services, and programs necessary to accomplish organizational goals (Cummings & Worley, 2008; Rothaermel, 2017). In this regard, the strategy describes the mode, the general approach to attaining the goal.

The practice of strategic management and the design of strategies cover a wide variety of decisions that are arranged in different formats depending on the structure of the client corporation and its complexity. At the level of strategy formulation, the options and their combinations tremendously increase. The strategic decisions are organized in several levels. Usually they are conceptualized as corporate, business area, strategic business unit, and operational strategies (Rothaermel, 2017). However, a strategic business unit may also have its strategy organized in several hierarchical domains of decisions. The "upper" levels involve decisions about the future of the whole corporation.

Top-level strategies are guiding documents for a closer look at the particular organization. Top level strategies are so general, that they have very indirect effect on the specific facility planning

decisions (Burton et al., 2006). For example, corporate strategies—like single business, diversification, integration, harvesting, retrenchment, liquidation, merger, joint venture, divestiture, and the combinations among them—direct only the most general decisions about the amount of resources that a corporation finds meaningful to invest in. In facility planning and programming, these top-level decisions are usually considered and implemented, but they are rarely questioned. This level of strategic decisions provides a basis for, let's say, considering applications for a bigger facility budget, or a preliminary anticipation that a scaling down instructions will be given from the Board and the acceptable alternatives will be only the ones that lead to a less expensive building.

The business strategies of growth, stability, and so forth have impact on decision-making when sizing the activity volume and workload, and a general attitude toward adopting new programs and services or shrinking the business profile. From that point on, a more specific approach to choosing activities and programs is needed. This strategy becomes the basis for the development of a means-ends chain that gives purpose and direction to every level in the organization (Burke, 2014; Burton et al., 2006; Rothaermel, 2017). At this level of strategic thinking, there are two decisions that strongly affect the facility plans. These are the scope and the results.

Task 9. Service programs and activity systems. The organizational strategy is gradually operationalized by making decisions about the scope of the service programs and activities. According to our guiding framework, here we move from working on the goal system of the organization to developing activity systems at the organizational level. At this point, the activities and programs that an organization wishes to pursue are narrowed down by referring to the specific market segment that was previously selected for servicing. The services and programs to be offered are gradually detailed regarding their nature and general parameters. This is a perplexing decision-making process because of the multitude of options available, the necessity for trade-offs, and the uncertain nature of the predictions and forecasts regarding the market and social environment. Defining the scope helps to plan a more focused investment and use of resources. The guidance produced at this point defines the decision area at the steps of selecting the marketing or program mix and the other more specific actions in the planning process. This reduces the area to be explored, the information to be collected, and the decisions to be made. Defining the scope is a way to decide the domains to be explored and selected for forecasting and feasibility studies in the steps that follow further on in the process.

At this level of decision-making, a major consideration is the selection of services to be offered. Even in the domain of a single business, there are a big number of service elements. It is neither possible nor necessary to provide each item of the service inventory even for a particular business

area. Because of that, an important point in the design of the organization is the selection of the services to be offered. This is an initial identification of the activities that will take place in the facility. In this way, the scope of the building is gradually refined.

The service mix provides the foundation for designing the organization. It is probably one of the most important decisions in the planning process. Once it is made, detailed studies in the domain of each service will be conducted. After that point, the decisions begin to gradually transfer from the strategic realm to the sphere of operations.

In the previous tasks, programmers have collected information and have constructed profiles of the customer groups, their needs and demands, cultural patterns of satisfying these needs, and sociodemographic characteristics that may suggest the shape of the desired services. Now on that basis, the process of identifying the services that will satisfy these needs begins.

This selection of services has two levels: "macro" and "micro." At the macro level, the general service strategy is designed. At the microlevel, a more fine-grained approach toward the service elements is adopted, considering the specific "shape" that the service would take, peculiar elements that enrich or streamline each particular service, and all attributes that distinguish among different providers. Many of these elements have direct bearing upon the organization of the space and the quantitative dimensions of the built environment.

Depending on the approach of the planners, the selection can begin with analysis of existing services, or it can start from scratch, constructing the service system as a completely new design. The first approach is more often used when there is an existing facility is present. The clients intend to use it after some kind of reconstruction. The second approach is applied in replacement projects, or when clients want to be sure they are not constrained by their past.

The programmers often aid the clients in exploring the range of services and programs and in selecting those which are most consistent with the mission and the goals of the organization, and the resources available as well. Usually experienced planners and programmers have an idea of the customer needs in a building type. They know the range of services at this facility type and use this knowledge for probing the clients on which of the customary programs and services are likely to take place. It is like providing some sort of menu and running down it.

In unfamiliar building types, the programmers have to start the explorations in service/activity program identification from scratch. They need to interact with the clients/users and develop a list of the activity programs that will take place there. This process is more explorative, time consuming,

involves more deliberation and consideration of various factors. In one case, the programmer carefully observed the functioning of other facilities from the same building type, assisted the client decision-makers to discuss the "service mix" (in this case, the amenities that the organization would provide for its patrons). The major guiding principles and goals were to satisfy better the needs of the patrons, and how to increase the public role and revenue of the organization.

Because most projects have budget limitations, programmers need to estimate the optimal service volume and from it, the capacity of the facility. This is a complex process that requires a lot of marketing expertise of the programming team. The information and decisions made at the previous tasks help to look at the business environment, market potential, service mix, and service provider activity systems in order to optimize the facility size and capacity. This process requires trade-offs between service programs and provider workload, taking into consideration profitability in services and facility budgets. The decisions at this step are very important for the initial shaping of the organizations. The content of this task can be periodically revisited in feedback loops in order to better fit service mix, provider activity systems, budget, and building.

Task 10. Developing general management and operations philosophy and model. General corporate philosophy is often discussed at the time of mission definition, but at that level it is mostly general views regarding corporate strategies in the socio-economic environment. The purpose of the initial discussion of organizational philosophy was to create a context for decision-making about the mission and the goals of the organization. After the service and program mix of the organization are adopted, the issue of organizing the operations comes into being.

The reasons for initiating this task are many. The operational philosophy has a strong impact on organizational models, staffing, and training. The basic principles of client philosophy serve also to evaluate the subsequent decisions in the process of programming the facility. The philosophy adopted by the client presupposes the general direction of development of the organization and the management of the operations. The analysis of client philosophy provides guidelines for the discussion and decision-making about operations in the next stages of the process. It suggests what subject matter and theoretical models to discuss, and which way to go in the subsequent tasks.

The choice of operational philosophy and model can affect both the organizational behavior and staffing loads, and the layout and cost of the facility. Therefore, the choices made at this task will presuppose the direction of decisions in the subsequent tasks. Once a particular philosophy and operational model are chosen, all decisions that follow should be in accordance with them. Any

discrepancies will lead to perceived inconsistencies, difficulties of reaching consensus, and problems in resolving issues in the subsequent tasks.

A very interesting situation exists in correctional facilities (prisons). There are several management philosophies and models: control model, responsibility model, and consensus model. Each of these implies a specific type of relations between inmates and custodial staff. As a result, two different surveillance philosophies emerge: remote and direct surveillance. The direct surveillance philosophy presupposes greater contact between prison inmates and staff. This precipitates changes in staffing levels and training. More importantly, direct supervision may require a floor layout that facilitates observation and surveillance. Otherwise, the number of surveillance posts has to be increased, thus increasing operation costs. Imagine how important the role of surveillance-conducive layout design is when compared with the expenses for introducing more 24-hour posts in order to secure all areas continuously observed. In addition, the surveillance strategy implies different choices in regard to durability of wall finishes, furniture and fixtures. Vandal-proof materials and furniture is much more expensive than a conventional public facility specification choices. Such environments are considered "normalized."

Task 11. Considering environmental conditions and spatial needs in general terms. At this stage, programmers make preliminary projections regarding necessary environmental conditions and spatial needs so that organizational structures can function efficiently. Programmers are interested mostly in the relationships between large components of the organization. This is important in order to imagine the general organization of the building. The idea is to create synergy and efficiency in the use of spatial resources. The goal is to provide different departments with the best adjacency options and joint support facilities in order to minimize operational expenses. Also, it is important that departments do not interfere with each other regarding flow of materials, energy, and information.

At this step, considering the emphasis on developing a process for collecting sociocultural and organizational information and making such decisions, we shift away from mainstream organizational studies and we enter the realm of sociospatial phenomena. Many of the decisions made at this task require experience in spatial organization and design, bridging the realms of the social and the spatial. However, in this article, we will be concise on these issues.

Task 12. Estimating current and future building resources. At this point of the programming process, we are at the built environment end of our framework for collecting programming information and making decisions. The programming team needs experienced architects to collect

information about existing buildings, their size, and mechanical and electrical system. It is important to estimate the usability of current building structures and the feasibility of remodeling. Programmers also study existing corporate space standards or, if there are no such documents, programmers discuss with the clients the development of new space standards.

Considering industry averages and space standards, programmers try to estimate the size of the buildings that need to be constructed or remodeled. At this step, they can provide the client with a very general idea how extensive the construction project will be, how much it will cost, and what the probable construction timeline will be. There should be enough information to estimate the feasibility of several variants of the construction project. As a result, there might be a need to realign all decisions made at the previous tasks.

Task 13. Multiple Feedback loops as needed. In our guiding framework, we have placed multiple feedback loops. These feedback loops can be initiated after each task and can lead to rethinking decisions made during that task or even several tasks back. Because of the emphasis on demonstrating the most important tasks in the programming process, as well as the page limitations of a typical paper, we will only mention that the feedback loop procedures are derivatives of the major task procedures. The feedback tasks are very important in cases when there are major discrepancies between decisions at different organizational levels, between requirements and solutions, and between financial resources and the expected building budget.

6. Project Phase: Departments and Operations planning

At this phase of the programming process, the programming team engages in developing the departments and the corresponding operation systems. These tasks will be designed on the basis of the strategy decisions presented above. Because of the nature of organizational decision making, certain decisions made at the strategic stage will be reviewed again, but this time in more detail. Another peculiarity of the process compared to the framework is that, in reality, the departmental and the individual levels of the framework are operationalized in one process. The process of researching departmental operations involves the study of individual agents and their needs. However, the programming process might involve another cycle for researching some individual agent roles and needs. The reader should not expect that the process is always linear with neatly defined inputs and outputs. Rather, it is recursive and even intertwined and entangled at times. However, even the linear format of our presentation here helps better understand the

programming process. It provides a guide for developing a custom-tailored plan for programmatic action.

Task 1. Department concept. This task starts with the introduction of the organizational concept at the departmental level. Up to this point, departments have been outlined very broadly during the organizational level process. Departmental concepts are derived from the overall organizational concept. They should not be developed by themselves and for themselves because each department is a component of the whole organization as a system. In the initial phase of organizational design, departments are matched, fitted together, aligned, and defined. When programmers move down to the departmental level and design of operations, they keep in mind this organizational structure, and continue working to fine-tune the department as well as for developing the departmental operations.

At the departmental level, programmers go into more detail about service programs and activity systems. According to the framework for collecting information, the study of departmental staff, customers, and other related groups starts with a general review that creates a background for developing the goals and objectives for this department in more detail, as well as its operations and services. After the services and operations are designed in a feasible way, programmers will have a more precise estimate what specific information they need to collect about the activity settings and agents.

Task 2. Connections and exchanges with other departments. At this stage, programmers need to develop a much closer understanding how the department is related to the other departments before they start working with client staff on the goals and objectives of the department. This is a preliminary investigation. After deciding on the mix of services and activity systems, programmers need to make a feedback loop and reconsider the nature of inputs and outputs, the points of exchange, and the flow of materials, energy, and information. These investigations and decisions might take several cycles to fine-tune each component.

Task 3. Departmental goals and objectives. At this point of the process, programmers have collected enough information to discuss the goals and objectives of the department in order to better align the department with the organizational system. This task might lead to a restructuring, arrangement, cooperation, integration, and so forth. The correct formulation of the departmental goals and objectives is important for providing a general reference framework for discussing the departmental operations and activity systems. This task is analogous to the development of the organizational mission, but at the departmental level these issues are discussed from another point

of view. Now programmers work with participants who see the departmental objectives “bottom up.” Programmers might see a somewhat different perspective of the department. Participants may propose valuable ideas in an effort to both support their own positions and to help the organization better perform. The stakeholders at the departmental level might propose or explore additional secondary objectives that support the main operations.

Task 4. Operational philosophy and models: The purpose of this task is to better understand this department’s operational philosophy and model, to evaluate it, and if necessary, to advise the client to adopt an alternative approach. Operational philosophy is the general system of assumptions, values, and beliefs for running the operations processes. The operational philosophy is both a conceptual basis and a guide for designing the operational models. It is very important to clarify the operational philosophy before working on service or activity models. One operational philosophy can lead to several operational models.

The scope of intervention varies from a simple recording of what the clients say, to an educational presentation--including information about advantages and disadvantages of alternative models, to facilitation of the operations restructuring discussion, and then up to direct proposal of a new operations method. Most of the programmers prefer to engage in only educational presentations and facilitation. However, in some situations, programmers may go to more encouraging and strongly worded suggestions and may use their ability to persuade and manage the internal politics of the department to promote certain ideas. The intervention range depends mostly, although not exclusively, on the professional commitments of the programmers, their view about their role in the process, and their confidence in their management expertise.

Task 5. Detailing the services and activity system. At the previous tasks of defining the service mix the organization will offer to its patrons, services and programs were defined in broad terms. Usually planners and clients make general decisions about types of services and the scope of the service packages to be offered. When the discussion continues at the departmental level, programmers use a more detailed and situation-specific approach. The user group may both discuss and propose some supplementary services that may improve the marketability of the main service package. This includes a detailed description of the ideas about the nature of a particular service, considerations about consequences and constraints, and the resources available.

When programmers are experienced, and the services are relatively uniform across the industry, they may just probe to be sure their understanding of the clients is correct. If programmers find that the case has specifics of its own, they may go and do some additional information gathering to

understand the differences. It might be helpful for programmers to discuss with clients the purpose, goals, and objectives of a particular service or a program, in order to get informed about the customers, typical behavior patterns, resources needed, and the interface with other services and programs in the same department and the organization in general.

For example, in a hospital, if oncology services are planned at the strategic phase, the rough organizational model will include the major components of an oncology program (e.g. prevention and early detection, diagnostic and treatment, rehabilitation, and continuing care), the main treatment therapies (e.g. superficial and orthovoltage therapy, cobalt units, hyperthermia, and brachytherapy), the available resources (radiation oncology is very expensive), and the relationship to other departments, programs, and services. At a departmental level, these aspects of operation will be discussed in more detail. There the emerging therapeutic and technology trends will be discussed extensively. The scope of each service or program will be further specified. For instance, radiation oncology can be designed in a number of ways. Certain services or treatments may be truncated, others may be impossible to eliminate without compromising the whole service package, and there may be ideas for integration with other facilities.

Very often services are formatted as programs. A program may be devoted to a specific service sub-type, but it may also cut across several typical services, thus providing a package geared toward a particular target population. These issues should be explored now and further discussed with the next tasks, particularly those related to the organization of the department and the analysis or/and redesign of the operations.

Task 6. Operations policies and procedures. The management of service/activity systems requires a number of instruments. Chief among them might be the operations policies and procedures. This is particularly important in facility types with an institutional nature of very tight operations control. The goal of this task is to identify and review the policies and procedures that may influence the design of the new facility. Programmers' responsibility very often is limited to documenting the policies that they believe would have spatial implications and provide this information to the architects. If necessary, it is appropriate to consider updating or even redrafting the existing policies and procedures. If there are no written policies, clients should be encouraged to develop a policy statement.

The policies and procedures are strongly influenced by and coordinated with the mission and philosophy of the organization, and in turn they set the framework for the operations in the facility. Policies are developed for every facet of the operation that requires some level of regulation and

control. In this regard, the operational structure of the organization needs to be analyzed, and the areas in need of regulation should be marked for policy development. Policies provide only general directions to follow so that the operations system contribute towards the attainment of organizational goals. By defining the limits of the operations design, the policies and procedures predetermine the "shape" of the activity systems in the facility. The policies and procedures stipulate activity patterns and ways the activities should be performed. In this way, they strongly influence organization of space and other spatial features that have to be decided by the architects. If the policies are not created or updated in advance, before the design of the new facility, it will be difficult to do that during the design process. And when the new facility is delivered, it is quite possible that the new activity systems might have to adapt to the opportunities afforded by the physical environment. This might considerably constrain the productivity and effectiveness of departmental operations.

Task 7. Designing the operations. In facilities programming, the design or redesign of operations might have two major parts. One is to redesign or improve the operations. The other is to look at the implications of the operations models regarding necessary conditions provided by the physical environment. The first part is in the realm of management consulting. The programming team needs to have such expertise "on board." In management consulting or organizational design terms, this might be considered as operations improvement, process improvement, or reengineering. In many situations, the management consulting terminology might supersede the social science terminology. The process of operations description and analysis is quite similar to task analysis, and it draws heavily from industrial engineering and job design (Bedny & Harris, 2008; Turner & Mcewan, 2003). Many of the specific considerations and problems can be better understood by looking at them from such a point of view. This typically happens in the institutional building types where operations processes are highly formalized and there is a lot of expertise in the field on how to organize them and how to run them. On the other side, in civic facilities like cultural or community centers, settings that are still not appropriated by management consulting, the way of thinking and conceptualizations might follow the social sciences traditions (Bedny & Harris, 2008; Turner & Mcewan, 2003).

Programmers invest a lot of time and effort to understand how users operate in spaces. This includes understanding what the users are trying to accomplish, which the major activities are, and how the process can be articulated in terms of consecutive steps. Their intent is to understand the use of space. This focus is different from the work of management consultants. When management consultants reengineer and streamline accounting operations, they are interested in things like the

intellectual acts performed, the flow of paperwork, the flow of information used, and the chain of reporting and supervising. Programming firms with less expertise in management consulting might obtain of deeper intervention. Such programmers might take an approach of facilitating a discussion of client participants and encouraging them to make informed decisions. In this field, programmers are considered to have done their job if they challenge what they think is incorrect and present alternative operations models to the clients.

Task 8. Researching the necessary conditions for departmental operations. The task of designing departmental operations/process improvement becomes a foundation for researching what we call “necessary conditions” in our knowledge content framework (Table 1.). After the operation model is established and developed in detail, programmers continue with researching what conditions these operations/activities need in order to be performed effectively and productively. For example, when office operations are studied for programming purposes, programmers focus exclusively on the behavioral structure, and its spatial dimensions in particular. Any office activities that are performed by one person at his/her desk have similar implications and could be tested in a similar way despite of the differences in their intellectual content. Similarly, programmers pay a lot of attention to actions that involve group meetings, teamwork, and movement in space. In many cases, this task provides feedback to the previous task on operations design. Programmers can and might improve operations by analyzing the spatial structure of the work and interaction process. It is not unusual that facilities programmers in this way contribute to process improvement at the workplace.

Thus, the registration of outpatients in a hospital is streamlined on the basis of reducing behavioral acts and providing opportunities for one staff member to service patients of several departments. This is the idea of the centralized registration. The productivity of a surgery room is measured considering the distance that the personnel need to walk to and from at the time of operation, thus predicting the average duration of a typical operation. This is much different from the operational focus of a surgeon who is trying to learn or to introduce a new method of thorax surgery. However, the efficiency of medical institutions depends on both these aspects, and the improvement of each one of them is important for achieving overall efficiency of the facility.

7. Project Phase: Collecting additional information at individual level

Programmers work extensively with teams of client participants and information providers. This takes a lot of time and a heavy toll on client operations. There are limits to how long and how many

times client employees can participate in facilities programming activities. For these reasons, programmers organize their work with the intent of minimizing the number of meetings and to streamline the process in order to minimize the hours of engagement of client employees. Thus, when working at the departmental level, programmers also collect information at the individual level of the building user according to the information collection framework that we have suggested. For example, the study of how employees or customers behave and act in a particular service or work situation is an integral part of the service itself. And once these client staffers or customers are invited for the departmental discussions, the programmers use their time to research individual needs.

The expansion of the guidelines about researching individual users can be organized in a more detailed presentation of the previous tasks or in a new very detailed presentation. This is work for a monograph-size publication. Considering the page limitations of a journal-size article, here we would only refer to these programming activities material and will talk about them in more detail in another article or a monograph. Up to now, we have provided enough guidance for developing a programming process and customizing it as needed.

8. Concluding Remarks

This trilogy of papers started with developing the philosophical and theoretical foundations for the study of sociospatial interactions (Popov & Goza, 2018). In the second paper (Popov & Goza, 2019), we created a guiding framework for facilities programming information collection and decision making. Now, in this, the third, paper, we used the guiding framework and developed a process model for the programming process. This process model is tentative and general, working as a guide for creating customized process designs for each programming project, considering the specifics of the programming situation. Furthermore, this model is only one of the possible operationalizations of the guiding data collection framework. We envisage other options as well.

The general nature of the framework requires additional effort for its operationalization, but at the same time, it allows for developing several process models or design for programming research. Considering the limited space of a journal article, we focused on the strategic and operational levels of decision making and information collection in facilities programming. In this case, we agreed that the information at the level of individual agents will be collected parallel to research at operational level in order to create a more holistic research process and save time and financial resources. Our methodology provides many levels of freedom when operationalizing the philosophical and

theoretical foundations. We are open, and we work on alternative knowledge content frameworks, as well as the ensuing process models of data collection and decision making in programming.

9. References

- BAILEY, K. (1994). *Sociology and the New Systems Theory*. Albany, NY: State University of New York Press. https://doi.org/10.1007/0-387-36274-6_19
- BAILEY, K. (2001). Systems theory. In J. Turner, (Ed.), *Handbook of Sociological Theory*, pp.379-401. New York, NY: Springer. https://doi.org/10.1007/0-387-36274-6_19
- BANATHY, B. (1996). *Designing Social Systems in a Changing World*. New York, NY: Springer. DOI: <https://doi.org/10.1007/978-1-4757-9981-1>
- BAUSCH, K. (2001) *The Emerging Consensus in Social Systems Theory*. New York, NY: Kluwer Academic/Plenum Publishers.
- BEDNY, G. and KARWOWSKI, W. (2007). *A Systemic-Structural Theory of Activity: Applications to Human Performance and Work Design*. Boca Raton, FL: CRC Press/Taylor and Francis. <https://doi.org/10.1201/9781420009743>
- BEDNY, G, and HARRIS, S. (2008). "Working sphere/engagement" and the concept of task in activity theory. *Interacting with Computers*, Vol. 20, No. 2, pp. 251-255. <https://doi.org/10.1016/j.intcom.2007.07.005>
- BURKE, W. (2014). *Organization Change: Theory and Practice*, 4th ed. Thousand Oaks, CA: SAGE.
- BURTON, R., ERIKSEN, B., HÅKONSSON, D. and SNOW, C. (2006). (Eds.). *Organizational design: A step-by-step approach*. New York, NY: Springer.
- CAMERON, K. S. and QUINN, R. E. (2011). *Diagnosing and Changing Organizational Culture: Based on the Competing Values Framework*, 3rd ed. San Francisco, CA: Jossey-Bass.
- CHERRY, E. (1999). *Programming for Design: From Theory to Practice*. New York, NY: John Wiley.
- CUMMINGS, T. and WORLEY, C. (2008). *Organizational Development and Change*, 9th ed. Mason, OH: South-Western College Publ.
- DAVIS, G. and SZIGETI, F. (1979). Functional and technical programming: When the owner/sponsor is a large or complex organization. In *Proceedings of the Fourth International Architectural Psychology Conference*. Louvaine-la-Neuve, July 10-14, 1979.
- ENGSTRÖM, Y. (2000). Activity theory as a framework for analyzing and redesigning work. *Ergonomics*, Vol. 43, No. 7, pp. 960-74. doi: 10.1080/001401300409143. PMID: 10929830.
- FARBSTEIN, J. (1986). *Correctional facility planning and design*. New York, NY: Van Nostrand Reinhold.
- HERSHBERGER, R. (2015). *Architectural Programming and Predesign Manager*. New York, NY: Routledge. <https://doi.org/10.4324/9781315645728>
- HOLT, G.R. and MORRIS, A.W. (1993). Activity theory and the analysis of organizations. *Human*

Organization, Vol. 52, No. 1, pp. 97-109.

- JABAREEN, Y. (2009). Building a conceptual framework: Philosophy, definitions, and procedure. *International Journal of Qualitative Methods*, Vol. 8, No. 4, pp. 49-62. <https://doi.org/10.1177/160940690900800406>
- KAPTELININ, V., KUUTTI, K. and BANNON, L. (1995). Activity theory: Basic concepts and applications. In B. Blumenthal, B., Gornostaev, J. and Unger C. (Eds.), *Human-computer interaction*, pp. 189-201. New York, NY: Springer. https://doi.org/10.1007/3-540-60614-9_14
- LANG, J. and MOLESKI, W. (2010). *Functionalism Revisited: Architectural Theory and Practice and the Behavioral Sciences*. New York, NY: Routledge. <https://doi.org/10.4324/9781315254838>
- POPOV, L. and GOZA, F. (2019). A framework for data collection in facility programming: The subject-object approach. *International Journal of Social Sciences*, Vol. VIII(2), pp. 108-133. DOI: 10.20472/SS2019.8.2.007
- POPOV, L. and GOZA, F. (2018). Philosophical foundations and metatheoretical considerations for creating frameworks to collect facility programming information. *International Journal of Social Sciences*, Vol. VII(2), pp. 92-108. DOI: 10.20472/SS2018.7.2.005
- PREISER, W.F.E. (Ed.). (1993). *Professional Practice in Facility Programming*. New York, NY: Van Nostrand Reinhold. <https://doi.org/10.4324/9781315717012>
- ROTHAERMEL, F. (2017). *Strategic management*, 3rd ed. New York, NY: McGraw-Hill.
- ŠIKÝŘ, M., GOROKHOVA, A. and ŠAFRÁNKOVÁ, J. (2019). Opinions on managing postmillennials by management university students. *International Journal of Social Sciences*, Vol. 8, No. 1, pp. 43-54., DOI: 10.20472/SS.2019.8.1.004
- TURNER, P. and MCEWAN, T. (2003). Activity theory: Another perspective on task analysis. *The Handbook of Task Analysis for Human-Computer Interaction*, Ch. 21, pp. 423-440. London: Kluwer.
- VORLICEK, J. and CERMAKOVA, K. (2017). Strategic behavior as the cause of business cycles. *International Journal of Economic Sciences*, Vol. VI(1), pp. 33-40. DOI: 10.20472/ES.2017.6.1.003