Data-Driven Industrial Human-Machine Interface Temporal Adaptation for Process Optimization

Daniel Reguera-Bakhache, Iñaki Garitano, Roberto Uribeetxeberria, Carlos Cernuda and Urko Zurutuza
Faculty of Engineering, Electronics and Computing
Mondragon Unibertsitatea
Arrasate-Mondragón, Spain
{dreguera,igaritano,ru,ccernuda,uzurutuza}@mondragon.edu

***Abstract*—The application of Artificial Intelligence (AI) into Industrial Human-Machine Interfaces (HMIs) moved old systems with physical buttons and analogue actuators into adaptive interaction models and context-based self adjusted interfaces.**

To date, little attention has been paid to industrial Human-Machine Interfaces (HMI) which play a vital role in the communication between worker and complex productive systems. Current industrial HMIs do not take into account operator behaviour, but rather focus on the production process. To enhance User Experience (UX) and improve performance it is necessary to adapt the interface to the needs of the operator.

This paper proposes a Machine Learning (ML) based operator interaction Data-Driven methodology to extract a set of interface adaptation rules. The methodology optimizes the interaction by reducing the number of actions and hence the amount of time and possible errors in repetitive monitoring and control tasks. An experiment with real operators was conducted to validate the proposed approach. The system was able to extract their interaction patterns and propose temporal interface adaptations, leading to a personalized, adaptive and more effective interaction.

***Index Terms*—Adaptive user interfaces, adaptation rules, intelligent industrial HMI, temporal interaction patterns.**

I. Introduction

Currently, in the so-called era of Industry 4.0, the rapid advance of new technologies and their application to manufacturing, has given rise to more and more complex automated industrial processes. This increase in complexity, has generated a need for more innovative information visualization systems that allow the machine operator to perform supervision and control tasks more effectively [1]. In such a context Human-Machine Interfaces (HMI) have become key drivers of the manufacturing process [2].

In recent decades, HMIs have been designed by industrial machine manufacturers focusing exclusively on the process. These are simple user interfaces that display information of different variables and how they impact the industrial process without considering the characteristics of the operator [3]. The main goal of industrial HMIs is to control the process and based on its status, help operators in the decision making process.

More recently, methodologies such as User-Centered Design (UCD) have been introduced, allowing the involvement of machine operators in the redesign of industrial HMIs [4]. However, this approach could be greatly improved with the addition of information about how industrial control system operators interact with HMIs.

The analysis of the knowledge extracted from HMI interactions, can provide more accurate information about their behaviour, capabilities and process knowledge. This data can then be applied to the interfaces redesign process leading to more efficient interaction by reducing the number of operator interaction errors. Therefore, introducing Adaptive User Interfaces (AUI) into industrial environments has emerged as a very promising technique to improve the interaction efficiency and to reduce the number of errors in the industrial process control.

Clickstream is generally understood as the sequence of interactions that the user performs with the different elements of the interface [5]. In industrial HMIs, this interaction is carried out through click events on the different interactive elements. These events are gathered in a file marked with a timestamp, and hence, we can determine at what time and with which element the operator has interacted. *Clickstream analysis* is one of the most frequently used techniques to understand user behaviour whilst is interacting with the interface [6] [7].

Modern industrial scenarios are frequently characterized by two factors: (i) high industrial process operator specialization and (ii) complex production systems, which have a strong influence on the activity and performance of the operator. In such a context it is critical that HMIs make smart adaptations to the capabilities of the operator [8], actions performed [9], and context [10]; while respecting the usability and integrity of the interface and without adversely affecting operator performance.

Any set of repetitive actions based on the process status performed by the operator over time, can be considered as an interaction pattern that can be automated. Such sequences can be gathered and Machine Learning (ML) algorithms can be applied to extract operator interaction patterns and provide industrial HMI adaptations. Due to the technological specifications and limitations of current industrial HMIs, these adaptations cannot be deployed automatically in the interface and hence, in this work, they are presented as a set of rules which will be subsequently supervised and implemented by a human. With the implementation of these adaptations in the industrial HMI, the interaction will be optimized reducing the number of clicks and error probability.

However, one of the most challenging aspect of AUIs is to decide when to adapt. If the layout is constantly changing, the usability of the industrial HMI can be penalized [11], adversely affecting operator effectiveness and productivity, and increasing the number of errors. It is therefore critical to determine in which time interval an interaction pattern occurs with the greatest frequency. With this information to hand, the AUI can generate an adaptation in the correct time interval without negatively impacting the usability of the HMI.

This paper presents a Data-Driven methodology, which, by applying ML to the information obtained from the industrial HMIs, extracts knowledge from the interaction data and proposes temporal adaptations in the HMI. These adaptations are formulated as a set of Event Condition Action (ECA) rules. To evaluate the feasibility of the proposed methodology an experiment was conducted in a real setup. The experiment tracked the interaction of 34 operators against a real industrial process HMI for a period of 151 days, 5 months.Following the proposed methodology, 54 interface adaptation rules were extracted, leading to interaction time optimization and error probability reduction.

II. Related Work

Adaptive User Interfaces (AUI) can broadly be defined as user interfaces that can be adapted based on user and the context, modifying not only the content to be displayed, but also the actions which can be performed without penalizing usability [12]. In recent years, there has been growing interest in AUIs and their application to different fields. For example to improve accessibility [13] or to obtain personalized interfaces [14].

One of the most widely used methodologies for adaptation has focused on analyzing the context of use. In [15] an adaptation framework was presented with emphasis on platform, user model and environment. Changing contexts and user emotions as inputs for user interface adaptation were studied in [16]. Although, these methods improve usability and enhance UX, they lack the incorporation of knowledge extracted from user interaction data in the adaptation process.

In contrast, other studies have incorporated user interaction data in the adaptation process. In [17] a framework for redesigning interface widgets based on user behavior was proposed. In [18] the authors presented an adaptation of the elements of a menu. This adaptation is performed by identifying the most and the least selected elements. The use of user interaction data in both studies is well-established, however, none of the presented adaptation scenarios were similar to the characteristics and limitations of industrial scenarios.

The application of AUI in industrial environments has been examined in several studies. In [19], an adaptive framework was proposed to assist operators in complex assembly processes. Through the use of different gesture and objects recognition devices, the system guided the operator in complex tasks. A system to develop AUI in Android devices was implemented and tested in [20]. This adaptation was based mainly on operators, roles and state of the system in manufacturing scenarios. Both these approaches are device-oriented but they do not consider industrial HMI, one of the most important element in manufacturing processes. Finally, in [21] an adaptive Cyber-Physical System (CPS) was presented, designed to help the operator by adaptive scheduling in the decision making process and thereby improving factory performance.

One final approach in the literature has been the evaluation of AUIs. In [22] the authors proposed a study of the influence of interface adaptations on end-user satisfaction. The combination of user feedback and context monitoring to evaluate the usability of interface real-time adaptations was studied in [23]. Other studies, such as [24] have shown that AUIs can facilitate users in complex tasks.

The aim of this paper is the definition of a Data-Driven methodology for interface temporal adaptations. Taking into account the characteristics and limitations of industrial HMIs, the methodology analyzes information from operator interaction and proposes a set of temporal adaptation rules to improve the interaction process and reduce interaction errors without penalizing usability.

III. Methodology

In this section we present the proposed Data-Driven methodology to generate temporal interface adaptations expressed as adaptation rules.

This methodology takes as inputs the industrial HMI definition and the interaction dataset, and by leveraging data-driven techniques, generates as output a list of interface adaptations rules in specific time intervals. Those adaptations consist of interface redesign actions and moving interface elements.

We start with the definition of the inputs, which includes a formal description of the industrial HMI and the operator interaction dataset. Next, we explain the different steps of the proposed methodology to (i) build valid interaction sequences, (ii) determine which are the most frequent sequences and hence candidates to adaptation, (iii) identify the time intervals when the sequences occur most frequently and (iv) generate the adaptation rule. Finally, the industrial HMI temporal adaptation rules are presented as set of Event Condition Action (ECA) rules.

A. Formal Description of the industrial HMI

Industrial HMIs are composed of information panels and clickable elements formed in most cases by several screens which run on industrial machines enabling real-time process monitoring and control. They also allow operator interaction, for example, to navigate between different screens or modify process variables. Thus, it is necessary to identify the elements that comprise all the interfaces, specifying whether they are informational items (e.g: text elements, labels) or interactive items (e.g: buttons and links).

In the current research a customized lightweight User Interface Description Language (UIDL) was designed because industrial HMIs are highly manufacturer dependent in terms of layout options and interface elements. A customized UIDL

TABLE I: Description of interface elements.

Field	Description
Id	Element identifier
Type	Element type (Informative or interactive)
Content	Displayed content (Text, image, alarm)
Event	Different events that can be triggered (Click)

allows us to formally describe each element of each interface, regardless of manufacturer limitations and characteristics, having all the necessary information related to the industrial HMI.

The next section sets out the screens that comprise the industrial application and for each interface the displayed elements are extracted. The most relevant information of each element, such as type, content and event is described in Table I.

A formal description of all interfaces $H = [hmi_1, hmi_2, \dots, hmi_n]$ and different items $I = [item_1, item_2, \dots, item_n]$ that comprise them are obtained as an output of this preliminary step. It is worthwhile mentioning that, each $item_i$ can be informative or interactive and belongs to a single interface hmi_i.

B. Interaction Dataset Specification

One of the existing shortcomings in industrial production environments is that industrial HMIs are designed without prior knowledge of how each operator interacts with the interface for process control and monitoring. The lack of this information in the designing process can make the interaction not optimal in terms of number of clicks. By default, industrial HMIs do not register operator's interactions. Although, some manufacturers, do offer additional tools[1] to track the raw interactions of users and process variables, storing them sequentially in a log file.

In the present paper the interaction raw dataset is considered as a sequential record of unique items, where each row is composed of **event** $e_i := \left[t_i, a_i^j, p_i\right]$ where:

- $t \in \mathbb{N}$ is the time stamp.
- $a \in A = \{a_1^j, a_2^j, \dots, a_{n-1}^j, a_n^j\}$ where A is a finite set of known interactions, indexed by j, in the different elements of the interfaces.
- $p_i = (v_1, \dots, v_m) : i \in I$ is the process variables values.

The output of this step is a log file where the interaction of every operator (including the temporary mark and process variables values) is registered, as illustrated in Table II.

C. Generation of Valid Interaction Sequences

In this step raw interaction dataset is parsed to generate valid worker interaction sequences. Perer et al. [25] defined an event sequence $E =< e_1, e_2, \dots, e_m >$ $(e_i \in D)$ as an ordered list of events, where D is the event set and i defines the order. This means that e_i has happened before that e_{i+1}.

[1]Siemens WinCC/Audit, https://support.industry.siemens.com/cs/document/109749101/wincc-audit-v7-5-sp1 (Accessed: Apr 2020)

TABLE II: Raw interaction dataset specification.

Timestap	Action	Process variables
t_0	a_0^2	$[v_0^1, v_0^2, \dots, v_0^m]$
t_1	a_1^1	$[v_1^1, v_1^2, \dots, v_1^m]$
t_3	a_3^6	$[v_3^1, v_3^2, \dots, v_3^m]$
t_7	a_7^5	$[v_7^1, v_7^2, \dots, v_7^m]$
t_8	a_8^1	$[v_8^1, v_8^2, \dots, v_8^m]$

In addition to this, to be considered as a sequence, E must contain at least two events.

Using the above definition as well as raw interaction dataset (described in the previous section) as input, it is then possible to extract valid interaction sequences. We consider a valid interaction sequence $s_i = [e_{init}, e_1^i, \dots, e_{k_i}^i, e_{final}]$ to be a set of events e_i where:

- a_{init}^j and a_{final}^j are known, determining the beginning and the end of a sequence.
- l is sequence's length and must be ≥ 2.

Algorithm 1 extracts all valid sequences in a sequential operation. First, $cond_u$ and $cond_v$ are defined with the specific actions which determine sequence beginning and end. Once both conditions are set, the algorithm examines the array finding the position p_u of the first action a_i^j that meets the condition $cond_u$. Then, it tries to locate the position p_v of the first action a_i^j that meets condition $cond_v$, updating the position of p_u if it finds a new one. Finally, the subsequence $s := E[e_{p_u}, e_{p_v}]$ is appended to S.

Algorithm 1 Valid sequences extractor.

```
Data: < E = (e_1, e_2, ..., e_n), cond_u, cond_v >
Result: S = (s_1, s_2, ..., s_n)
n ← len(E)
i ← 1
while (i ≤ n) do
    while (i ≤ n)and(e_i ≠ cond_u) do
        i ← i + 1
    end
    p_u ← i
    while (i ≤ n)and(e_i ≠ cond_v) do
        if (e_i = cond_u) then
            p_u = i
        end
        i ← i + 1
    end
    p_v ← i
    if (i ≤ n) then
        Append E[e_{p_u}, e_{p_v}] to S
    end
    i ← i + 1
end
```

This step generates a dataset $S = (s_1, s_2, \dots, s_n)$ which contains all the valid sequences performed by the operator.

D. Filtering and Selection of Candidate Sequences

The main objective of this step is to select sequences that occur with high frequency, and hence, candidates for adaptation. We classify a sequence s as frequent if its repetition ratio r in a finite time interval is above a predefined threshold α. The repetition ratio r and α have an inverse relationship.

Once α is established defining which is the minimum repetition ratio to be considered as a repetitive sequence, we review all the valid sequences of dataset S calculating r for each sequence s. If r is higher than α, the sequence s_i is appended to candidate sequences list C. Algorithm 2 sets out the process.

Algorithm 2 Filtering candidate sequences.

```
Data: S = [s_1, s_2, ..., s_n]
Result: C
Initialize α
for (i = 0, i < len(S), i++) do
    Calculate r of s_i
    if r > α then
        Append s_i to C
    end
end
```

The output of this step is a dataset C, containing the sequences whose repetition ratio r is above the threshold performed by the operator in an defined finite time interval. These interaction sequences become candidates for the adaptation.

E. Temporal Adaptation Rules Generation

Using the dataset C as input, in this step we generate the temporal adaptation rules in two phases: (i) Time interval detection and (ii) Adaptation actions based on sequence mining.

The time interval T where the action is most frequent is identified and the sequence is parsed to determine which $item_i$ might be adapted. The adaptation rule defines which $item_i$ should be moved to which interface hmi_i. In this way, the number of user clicks required to perform the action is reduced and therefore the interaction is optimized.

The rules are formally denoted by an ECA rules where:

- E: Event a_1^j is triggered
- C: $t_i \in T$
- A: Move $item_i$ to hmi_1

1) Detection of Time Intervals: To resolve the problem of identifying the time intervals in which the rule should be activated, it is necessary to first determine the intervals over the time in which user interaction is more frequent. This can be addressed by *1-D* Density-based clustering.

This type of unsupervised learning method identifies different clusters based on the density over the space. Dense areas are considered as a cluster and non-dense areas as a separation. These clusters indicate the time intervals when operator interactions are more frequent, and therefore the adaptation rule should be activated.

In this step, we determine the relevant clusters from dataset C. The first and last element from each cluster are selected, extracting the information of the begining and the end of the time interval. By detecting these time intervals, we ensure that the adaptation rule is only activated when the candidate sequence is frequent.

2) Adaptation Actions Based on Sequence Mining: Once the time intervals are defined, the adaptations are generated. In this step, the interaction sequence is parsed analyzing each $item_i$ the user has interacted with.

When the end of the sequence is reached, we check which information elements exist in this interface hmi_n, and if they are not in the interface hmi_1 an adaptation is proposed. Algorithm 3 describes the process.

Algorithm 3 Adaptation actions generator.

```
Data: < C = (s_1, s_2, ..., s_n) >
Result: R = (r_1, r_2, ..., r_n)
foreach s_i in C do
    extract different steps of s_i
    if item_i of hmi_n is informative and not in hmi_1 then
        move item_i to hmi_1
    end
end
```

At the end of this step a set of adaptation rules $R = [rule_1, rule_2, \ldots, rule_n]$ are generated. The implementation of this set of rules R, will modify the interfaces, leading to a reduction of number of clicks in repetitive tasks performed by the operator.

IV. Validation

We conducted an experiment to validate the proposed methodology, with two main objectives: (i) To infer interface adaptations based on operator-HMI interactions and (ii) To validate the inferred adaptation rules by the application in the industrial HMI.

With this purpose in mind, we divided the experiment into two phases: (i) Phase 1 lasts until enough interaction data to generate interface adaptation rules have been gathered to train the model, and (ii) Phase 2 comprises the validation of changes made in the interface after applying the adaptation rules.

The aim was to test and validate the proposed methodology in a real industrial scenario. The experiment consists on an industrial machine in which a number of operators interact with a productive system through the HMI to obtain a personalized product. In this context, an industrial mixing machine from the food sector was used in the experiment. Fig. 1 illustrates the main interface of the industrial mixing machine, in which the operator can adjust different parameters by clicking on the interactive elements. Every time a mixture is ordered the operator carries out a set of single interactions with the machine HMI. These interactions are related to adjusting different parameters such as additive quantity or mixture type.

The interaction process of preparing a mixture can be described as a finite state machine (FSM) where the operator sets up different parameters until the values are considered OK. Fig. 2 displays the finite state machine.

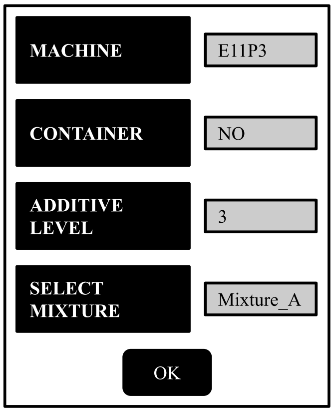

Fig. 1: Industrial mixing machine main interface.

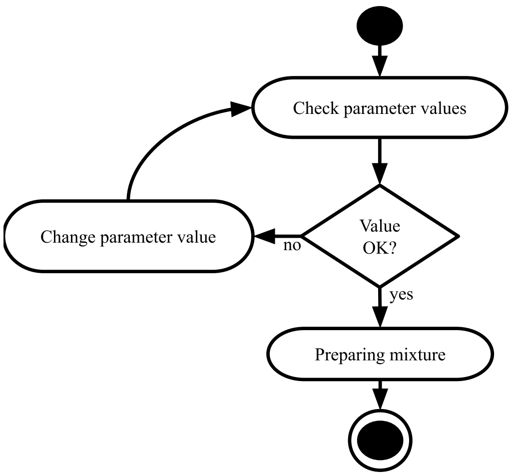

Fig. 2: Interaction process of the finite state machine.

A. Participants

Thirty-four volunteers (21 male and 13 female) aged 23 to 45 participated in this study. All the interactions with the different elements of industrial mixing machine interface were tracked during the working shift.

B. Formal description of the interface and interaction dataset

The industrial mixing machine application consists of 6 different interfaces. A JSON file describes these six interfaces and all the elements. Each element has a single numeric identifier and belongs to a unique interface. Fig. 3 illustrates an example interface that is formally described as follows:

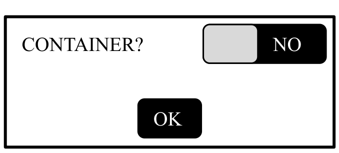

Fig. 3: Container selection interface.

TABLE III: Example trace of raw interaction dataset.

Timestamp	Element
1557994085106	BTN1OK
1557994215793	BTN0u8
1557994221803	BTN1Container
1557994230690	BTN1Additive
1557994231974	BTN3ReduceAdditive

```
interface4:{
  elements:{
    45:{
        type:text,
        text:"Container?",
        editable:no,
        style:{background-color:black,
               text-color:white}
        }
    46:{
        type:toggle,
        event:click,
        action:togglevalue}
    47:{
        type:button,
        text:"OK",
        event:click,
        action:navigate,
        style:{background-color:black,
               text-color:white}
        }
        }
 }
```

The next step was to track all single interactions with the different elements of the interfaces. Every time a operator interacted with the HMI all the information related to that interaction (Time stamp, element and process variables) was collected in a file. Table III represents an example trace of gathered interaction dataset in a time slot.

C. Procedure

1) Valid sequence generation: Taking into account the formal description and the interaction dataset, the valid interaction sequences were generated. In this step 10,008 single interactions were transformed into 894 valid sequences. Each sequence corresponded to a mixture customization.

A valid sequence starts when the operator logs into the system and finishes when clicks OK button. Table IV shows an example of a valid interaction sequence. The sequence starts when the user logs into the system *'BTN0u9'*, reduces additive level *'BTN3ReduceAdditive'* (by default starts from 3), selects mixture-A *'BTN5Mixture-A'* and finishes when *'BTN1OK'* is clicked.

2) Candidate sequences selection: This step was divided into two different actions, (i) first the most interactive operators were selected and then (ii) the most repetitive sequences were identified.

Fig. 4 shows all the sequence repetitions. As we can observe, the sequence repetitions follow an exponential trend meaning that the most repetitive interaction sequences were repeated many times and the less common interactions were repeated infrequently. For this reason, we defined the threshold

TABLE IV: Example of a valid interaction sequence.

Timestamp	Element
1558333048305	'BTN0u9'
1558333063963	'BTN1Additive'
1558333065339	'BTN3ReduceAdditive'
1558333067170	'BTN3OK'
1558333069467	'BTN1Mixture'
1558333073638	'BTN5Mixture-A'
1558333075042	'BTN5OK'
1558333076809	'BTN10OK'

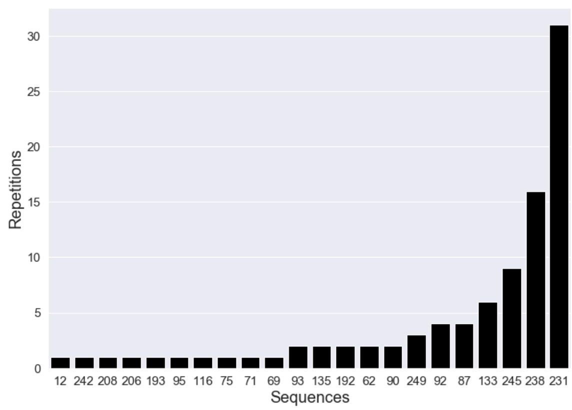

Fig. 4: Operator most repetitive sequences over time.

α for filtering the candidate sequences as 1st quartile because statistical values as average or median were not representative in this situation. At the end of this step, the adaptation candidate sequences were obtained for each operator, which were those above the threshold.

3) Density-based clustering: To determine the dynamic fitting intervals, three different clustering algorithms were tested: HDBSCAN [26], OPTICS [27] and MeanShift [28]. In the case of HDBSCAN and OPTICS, an initial parameter tuning is required. This configuration is problem dependent and can distort the results if is not properly done. This is why MeanShift algorithm is applied, specifying not to cluster all points. Hence, outliers could be discarded and more precise intervals were obtained.

We represented with a point each repetition of the candidate sequence over time, then we performed the density clustering identifying the most density areas considered as a cluster.

Fig. 5 illustrates the distribution on a 1-D space before the clustering and Fig. 6 displays the different clusters and outliers detected by MeanShift algorithm. In this case, for the displayed sequence, 3 different clusters were detected, corresponding to the hours when operator prepares more mixtures.

These automatically detected clusters are consistent with the preliminary exploratory analysis undertaken to observe the time intervals in which operators prepare more mixtures:

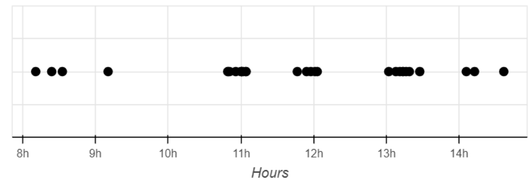

Fig. 5: Sequence repetition over time before clustering.

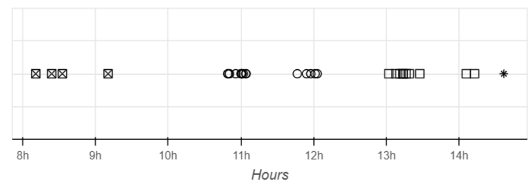

Fig. 6: Identifyed clusters after applying MeanShift algorithm.

between 7:00 am and 8:00 am when the work shift starts, between 10:00 am and 11:00 am when the operator usually has a break, and between 1:00 pm and 2:00 pm after operator has lunch break. Fig. 7 illustrates times of the day when the operators interact more with the industrial mixing machine interface.

For each detected cluster, we selected the leftmost and rightmost points, obtaining the begin and the end of the adaptation interval.

4) Adaptation rules generation: In this step the candidate sequences per operator were parsed and the adaptations rules were generated. Those adaptations consisted on moving the final informative elements to the initial HMI.

An example of adaptation rule is:

- Event: *u3 is logged.*
- Condition: *current time is in detected clusters [["08:10:37"-"09:25:36"] , ["10:49:03"-"12:04:57"], ["13:01:50"-"14:36:44"]]*
- Action: *Move element BTN3Additive to interface2* and *Move element BTN5Mixture-C to interface2.*

The rule can be explained as operator u3 performs the same interaction, in this case customizing the mixture, in the identi-

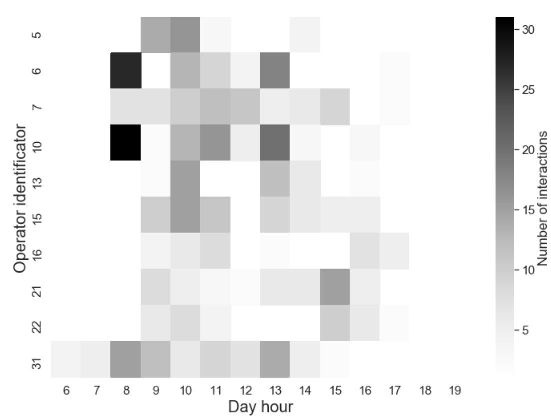

Fig. 7: Times of day when the greatest interaction takes place.

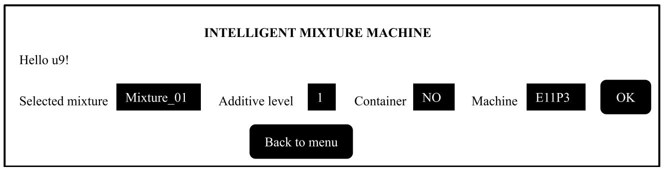

Fig. 8: Industrial mixing machine main interface adapted.

TABLE V: Results obtained in Phase 1 (α = 25th percentile).

Description	Value
Single interactions	10,008 clicks
Valid sequences	894
Noise rate	11.26%
Candidate sequences	54 sequences
Nº of rules generated	54 rules
Average nº of clicks	10.36 clicks
Average sequence duration	29.62 seconds
Accuracy	91.7%

fied time intervals (clusters). Therefore, it is proposed to adapt the initial interface displaying informative elements of the final interface. This adaptation reduces the number of clicks and improves the interaction, since the operator could select the most probable additive level and mixture type in the first interface, according to the historical data.

In addition, the proposed adaptation is integrated in the application by parsing the generated adaptation rules. Once the adaptation rule is activated, the operator will see the proposed adapted interface in Fig. 8.

V. Results and discussion

The interaction data gathered during the experiment allowed us to validate the proposed methodology. Table V and Table VI present an overview of the obtained results. For the experiment's Phase 1, 10,008 were used to infer the interface adaptation rules. In the experiment's Phase 2, 1,024 single interactions were used to validate the model. Accuracies achieved in both phases have been 91.7% and 84.5%, respectively for training and validation.

Establishing the threshold α as the 25th percentile for candidate sequences filtering, we obtained 54 different interaction sequences and hence 54 temporal adaptation rules. On the other hand, by extending the threshold to a less restrictive value, we observed that the number of candidate sequences increased. When α was set to the 30th percentile, the number of candidate sequences increased to 195 sequences and with the 35th percentile to 212. It should be noted that high volume of interface adaptations adversely affects usability [11]. This is why α was set to a restrictive value.

To assess the improvement in interaction by the application of the generated temporal adaptation rules three different evaluation metrics were defined:

- **Interaction sequence number of clicks**: Number of clicks required by the operator to perform the action inside the detected time interval.
- **Interaction sequence duration**: Time in seconds taken by the operator to perform the sequence inside the detected time interval.
- **Operator validation of the adaptation rule**: Operator assessment of the proposed temporal adaptation.

TABLE VI: Results obtained in Phase 2 Validation.

Description	Value
Single interactions	1,024 clicks
Valid sequences	91
Average nº of clicks	4.58 clicks
Average sequence duration	7.24 seconds
Accuracy	84.5%

A. Interaction sequence number of clicks

Table VI clearly shows a significant decrease in the number of clicks. Using the standard, unmodified interface the average of number of clicks in candidate sequences was 10.36, whereas after the adaptation rule was activated, the average was reduced to 4.58 clicks.

This decrease occurs because the activation of adaptation rule adapts the interface to the operator. Once the operator accesses the main interface of the industrial mixing machine and the adaptation rule is generated and applied, the initial interface adapts itself and displays the informative elements as shown in Fig. 8. Thus, the operator has only to press "OK" if the adaptation rule is correct, or the "Back to the menu" button and perform the sequence to select the values in the contrary case.

In addition to decreasing the time required for the operator to complete a given task, the lower number of clicks also has implications for error reduction. Less clicks means less opportunity for errors when carrying out repetitive monitoring and control tasks.

B. Interaction sequence duration

Table VI shows the reduction in interaction sequence duration. Applying the proposed methodology to the interaction dataset, the sequence duration was reduced from an average of 29.42 to 7.24 seconds. Being between 3-4 seconds the interaction sequence duration when the proposed adaptation is successful.

Once the adaptation rule is activated and the interface is adapted, the operator does not have to interact with the industrial mixing machine interface for setting different parameters, and hence the interaction sequence duration is significantly reduced.

C. Operator validation of the adaptation rule

This metric is defined as the percentage of times the operator presses the "OK" button when an interface adaptation rule is activated and presented in the interface.

As can be seen in Table VI, 84.5% of proposed interface adaptations were accurate, and validated by the operator selecting OK in the adapted interface. This signifies that the interface adaptations rules generated by the system were widely accepted. In the remaining 15.5% of cases the adaptation rules did not present the required parameters and the operators clicked the "Back to menu" button to make further adjustments.

VI. Conclusions and future work

The use of Adaptive User Interfaces (AUI) in industrial scenarios suggests a promising paradigm where interfaces are able to adapt intelligently to the needs of the operator, and hence improve overall industrial process performance. In this paper, a Data-Driven methodology for generating temporal adaptation rules in industrial scenarios was defined and then validated experimentally. The main contribution is a methodology to extract interaction patterns as adaptation rules from a raw operator-HMI interaction dataset. These adaptations optimize the interaction by reducing number of clicks thereby improving operator efficiency when performing repetitive control and monitoring tasks.

Obtained results by the end of the experiment validated the adaptations presented the main interface, although several improvements could be carried out in future works, to further enhance the methodology. To improve time interval detection, a more exhaustive study of the different density clustering algorithms parameter tunning is required. With this modification, we would be able to select the best clustering algorithm for each situation and obtain adaptation intervals of much greater accuracy. As regards threshold definition for candidate sequences filtering, further research should be carried out to establish an optimal value for each scenario.

References

[1] O. Niggemann, G. Biswas, J. S. Kinnebrew, H. Khorasgani, S. Volgmann, and A. Bunte, "Data-driven monitoring of cyber-physical systems leveraging on big data and the internet-of-things for diagnosis and control." in *DX@ Safeprocess*, 2015, pp. 185–192.

[2] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6, pp. 1–10, 2017.

[3] V. Villani, L. Sabattini, F. Loch, B. Vogel-Heuser, and C. Fantuzzi, "A general methodology for adapting industrial hmis to human operators," *IEEE Transactions on Automation Science and Engineering*, 2019.

[4] E. Aranburu-Zabalo, G. Lasa-Erle, G. Iruretagoiena-Irazusta, D. Reguera-Bakhache, and J.-K. Gerrikagoitia-Arrien, "New user centered methodologies for software development in the industry 4.0 era," 2017.

[5] L. Shen, L. Cheng, J. Ford, F. Makedon, V. Megalooikonomou, and T. Steinberg, "Mining the most interesting web access associations," in *WebNet World Conference on the WWW and Internet*. Association for the Advancement of Computing in Education (AACE), 2000.

[6] Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson, "Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 321–330, 2016.

[7] G. Wang, X. Zhang, S. Tang, H. Zheng, and B. Y. Zhao, "Unsupervised clickstream clustering for user behavior analysis," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 225–236.

[8] V. Villani, L. Sabattini, J. N. Czerniaki, A. Mertens, B. Vogel-Heuser, and C. Fantuzzi, "Towards modern inclusive factories: A methodology for the development of smart adaptive human-machine interfaces," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2017, pp. 1–7.

[9] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *2014 12th IEEE international conference on industrial informatics (INDIN)*. Ieee, 2014, pp. 289–294.

[10] L. Urbas, S. Hennig, H. Hager, F. Doherr, and A. Braune, "Towards context adaptive hmis in process industries," in *2011 9th IEEE International Conference on Industrial Informatics*. IEEE, 2011, pp. 244–249.

[11] L. Findlater and K. Z. Gajos, "Design space and evaluation challenges of adaptive graphical user interfaces," *AI Magazine*, vol. 30, no. 4, pp. 68–68, 2009.

[12] L. Rothrock, R. Koubek, F. Fuchs, M. Haas, and G. Salvendy, "Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms," *Theoretical issues in ergonomics science*, vol. 3, no. 1, pp. 47–84, 2002.

[13] M. González-García, L. Moreno, and P. Martínez, "Adaptation rules for accessible media player interface," in *Proceedings of the XV International Conference on Human Computer Interaction*, 2014, pp. 1–8.

[14] A. Aztiria, E. Castillejo, A. Almeida, and D. L. De Ipiña, "Adapting user interfaces based on user preferences and habits," in *2014 International Conference on Intelligent Environments*. IEEE, 2014, pp. 9–15.

[15] V. Genaro Motti, "A computational framework for multi-dimensional context-aware adaptation," in *Proceedings of the 3rd ACM SIGCHI symposium on Engineering interactive computing systems*. ACM, 2011, pp. 315–318.

[16] C. Märtin, C. Herdin, and J. Engel, "Model-based user-interface adaptation by exploiting situations, emotions and software patterns," in *International Conference on Computer-Human Interaction Research and Applications*, 2017.

[17] L. Leiva, "Interaction-based user interface redesign," in *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. ACM, 2012, pp. 311–312.

[18] S. Bouzit, G. Calvary, D. Chêne, and J. Vanderdonckt, "Interface adaptivity by widget promotion/demotion," in *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS '19. New York, NY, USA: ACM, 2019, pp. 18:1–18:6. [Online]. Available: http://doi.acm.org/10.1145/3319499.3328237

[19] F. Wallhoff, M. AblaBmeier, A. Bannat, S. Buchta, A. Rauschert, G. Rigoll, and M. Wiesbeck, "Adaptive human-machine interfaces in cognitive production environments," in *2007 IEEE International Conference on Multimedia and Expo*. IEEE, 2007, pp. 2246–2249.

[20] O. Neira, A. N. Lee, J. L. M. Lastra, and R. S. Camp, "A builder for adaptable human machine interfaces for mobile devices," in *2013 11th IEEE International Conference on Industrial Informatics (INDIN)*. IEEE, 2013, pp. 750–755.

[21] D. Mourtzis and E. Vlachou, "A cloud-based cyber-physical system for adaptive shop-floor scheduling and condition-based maintenance," *Journal of manufacturing systems*, vol. 47, pp. 179–198, 2018.

[22] L. Findlater and J. McGrenere, "Impact of screen size on performance, awareness, and user satisfaction with adaptive graphical user interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1247–1256.

[23] E. Yigitbas, I. Jovanovikj, K. Josifovska, S. Sauer, and G. Engels, "On-the-fly usability evaluation of mobile adaptive uis through instant user feedback," in *IFIP Conference on Human-Computer Interaction*. Springer, 2019, pp. 563–567.

[24] L. Zhang, Q.-X. Qu, W.-Y. Chao, and V. G. Duffy, "Investigating the combination of adaptive uis and adaptable uis for improving usability and user performance of complex uis," *International Journal of Human–Computer Interaction*, pp. 1–13, 2019.

[25] A. Perer and F. Wang, "Frequence: Interactive mining and visualization of temporal frequent event sequences," in *Proceedings of the 19th international conference on Intelligent User Interfaces*, 2014, pp. 153–162.

[26] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering." *J. Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[27] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," in *ACM Sigmod record*, vol. 28, no. 2. ACM, 1999, pp. 49–60.

[28] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 603–619, 2002.